"ANALYSING MULTICLOUD CHALLENGES BY MANAGING END-TO-END OPERATIONS FOR COMPREHENSIVE MULTICLOUD MANAGEMENT THROUGH DIFFERENT TECHNIQUES"

A Thesis
Submitted towards the Requirements for the Award of Degree of

Doctor of Philosophy
In
COMPUTER SCIENCE & ENGINEERING

Under the Department of Computer Science & Engineering

By

Malege Sandeep

(Enrolment No.: 161588517363)

Under the Supervision of

Dr . Yamaganti Rohita

Department of Computer Science & Engineering

2025

P.K. UNIVERSITY

NH- 27, Vill. Thanra (P.O. Dinara)

Shivpuri, M.P. India - 473665

www.pkuniversity.edu.in

CERTIFICATE OF THE SUPERVISOR

This is to certify that the work entitled **"Analysing Multicloud Challenges by Managing End-To-End Operations for Comprehensive Multicloud Management Through Different Techniques"** is a piece of research work done by **Shri/Smt./Ku. Malege Sandeep** Under my Guidance and Supervision for the degree of Doctor of Philosophy of **faculty of Computer Science & Engineering , P.K. University (M.P) India.**

I certify that the candidate has put an attendance of more than 240 day with me. To the best of my knowledge and belief the thesis:

I – Embodies the work of the candidate himself.
II – Has duly been completed.
III –Fulfil the requirement of the ordinance relating to the Ph.D. degree of the University.

Signature of the Supervisor

Date: …….

P.K.University
Shivpuri (M.P.)

Enrolment Number 161588517363

Select Course Ph.D. in Computer Science & Engg. (Course Wo

Select Semester 1

Result

Enrolment Number : 161588517363

Candidate Name : MALEGE SANDEEP

Course : Ph.D. in Computer Science & Engg. (Course Work)

Father's Name : MALEGE ANAND

Year/Sem : 1

Mother's Name : M. CHANDRAKALA

Session : 2020-21

Subject Name	Internal T	Internal T	Internal P	Internal P	External T	External T	External P	External P	
Research Methodology	N/A	N/A	N/A	N/A	50	41	N/A	N/A	41 / 50
Subject Specialization	N/A	N/A	N/A	N/A	100	76	N/A	N/A	76 / 100
Review of Literature	N/A	N/A	100	71	N/A	N/A	N/A	N/A	71 / 100

Marks Obtained 188 Result Pass Max Marks 250

- Student must pass in Theory and Practical separately.
- For pass the candidate is required to obtain 40% marks in each paper and 50% marks in aggregate.
- For pass the Ph.D candidate is required to obtain 65% marks in aggregate.

P.K.UNIVERSITY
SHIVPURI (M.P.)

UniversityEstablished Under section 2F of UGC ACT 1956 Vide MP Government Act No 17 of 2015

CENTRAL LIBRARY

Ref. No. PKU/C.LIB /2025/PLAG. CERT./227 **Date: 10.11.2025**

CERTIFICATE OF PLAGIARISM REPORT

1. Name of the Research Scholar	:	Malege Sandeep
2. Course of Study	:	Doctor of Philosophy (Ph.D.)
3. Title of the Thesis	:	Analysing Multicloud Challenges By Managing End-To end Operations For Comprehensive Multicloud Management Through Different Techniques
4. Name of the Supervisor	:	Dr. Yamaganti Rohita
5. Department	:	Computer Science & Engineering
6. Subject	:	Computer Science & Engineering
7. Acceptable Maximum Limit	:	10% (As per UGC Norms)
8. Percentage of Similarity of Contents Identified	:	9%
9. Software Used	:	Drillbit
10. Date of Verification	:	29.10.2025

(Librarian, Central Library)
P.K.University, Shivpuri (M.P.)

LIBRARIAN
P.K. University
Shivpuri (M.P.)

ADD: VIL: THANRA, TEHSIL: KARERA, NH-27, DIST: SHIVPURI (M.P.) -473665
MOB: 7241115902, Email: library.pku@gmail.com

DrillBit

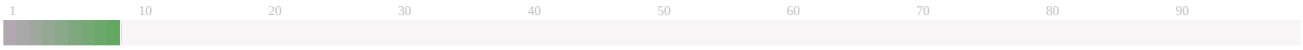
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

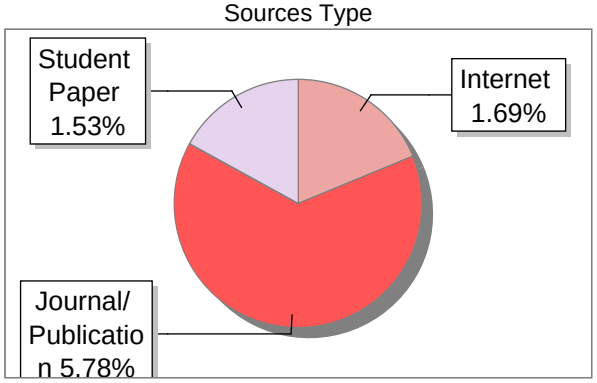
Author Name	Malege Sandeep
Title	ANALYSING MULTICLOUD CHALLENGES BY MANAGING END-TO-END OPERATIONS FOR COMPREHENSIVE MULTICLOUD MANAGEMENT THROUGH DIFFERENT TECHNIQUES
Paper/Submission ID	4588710
Submitted by	library.pku@gmail.com
Submission Date	2025-10-29 10:32:32
Total Pages, Total Words	207, 58732
Document type	Thesis Chapter

Result Information

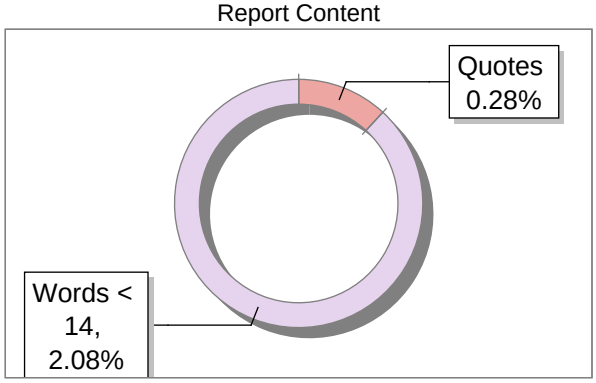
Similarity **9 %**



Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DECLARATION BY THE CANDIDATE

I declare that the thesis entitled "**Analysing Multicloud Challenges by Managing End-To-End Operations for Comprehensive Multicloud Management Through Different Techniques"** is my own work conducted under the supervision of **Dr . Yamaganti Rohita** at **Department of Computer Science & Engineering , P.K. University (MP), India.**

Approved by Research Degree Committee. I have put more than 240 days of attendance with supervisor at the center.

I further declare that to the best of my knowledge the thesis does not contain my part of any work has been submitted for the award of any degree either in this University or in any other University, without proper citation.

MALEGE. SANDEEP

Signature of the candidate

Date: ……

Place: …….

FORWARDING LETTER OF THE HEAD OF INSTITUTION

The Ph.D. thesis entitled "**Analysing Multicloud Challenges by Managing End-To-End Operations for Comprehensive Multicloud Management Through Different Techniques**" Submitted by **Shri/Smt./Ku. Malege Sandeep** is forwarded to the university in six copies. The candidate has paid the necessary fees and there are No Dues outstanding against him/her.

Name ……..…….…. Seal ……

Date: …… …….. Place: …….....

(Signature of Head of Institution where the Candidate was Registered for Ph.D degree)

Signature of the Supervisor

Date: ……..

Address: ……..

Place: …….

……..

COPYRIGHT TRANSFER CERTIFICATE

Title of the Thesis: **"Analysing Multicloud Challenges by Managing End-To-End Operations for Comprehensive Multicloud Management Through Different Techniques"**

Candidates Name : Ms/Mr. Malege Sandeep

COPYRIGHT TRANSFER

The undersigned hereby assigns to the P.K.University, Shivpuri (MP) all the copyrights that exist in and for the above thesis submitted for the award of the Ph.D. degree.

MALEGE. SANDEEP

Date: Ms/Mr. **Malege Sandeep**

Signature

ACKNOWLEDGEMENT

The Acknowledgement of a research endeavor is an opportunity to express appreciation for the joint effort of those who have been sources of motivation and support and who have contributed invaluable knowledge toward achieving success in the pursuit of knowledge .With utmost humility and pride, I reflect on the inspiring individuals who have propelled my work forward, and I am filled with immense pleasure, gratitude and thanks for their unwavering dedication.

First for most I would like to thanks almighty god for preservence, courage and strength bestowed on me to undertake this project.

Furthermore, I extend my sincere thanks to the Hon'ble **Shri Jagdeesh Prasad Sharma** Chancellor, P.K University Shivpuri (MP) for providing the necessary infrastructure and facilities that were essential during my research work .My profound sense of gratitude, faith and awe goes to Hon'ble **Prof. (Dr.) Yogesh Chandra Dubey** Vice Chancellor, P.K. University Shivpuri (MP) for his generous and valuable support throughout my research.

I take this pleasant opportunity to express my deep sense of gratitude and reverences to respectable Hon'ble **Dr. Jitendra Kumar Mishra** Director (Admin), P.K.University Shivpuri (MP) for his constant encouragement and providing me with the required facilities that enabled me to complete my project work.

For most, I extend my deepest appreciation to my esteemed Supervisor **Dr . Yamaganti Rohita** Department of **Computer Science & Engineering ,** P.K.University without whom this work would not have come to fruition. His exceptional guidance, unwavering support, kind cooperation and provision of conductive work environment have been invaluable beyond measure. I remain indebted to him for instilling in me a relentless pursuit of excellence, a strong sense of honesty, and respect for ethical principles that governs our profession.

I would also like to extend my gratitude and deep appreciation to respected **Prof.Bhaskar Nalla** Dean Research & I/c Ph.D.Cell, **Dr. Aiman Fatma** Dean of Academics, **Prof. Jitender Kumar Malik** Dean of Faculties, **Dr.**

Deepesh Namdev Registrar P.K. University for inspiring me from the beginning of this work in spite of their busy schedules.

I owe a great deal of application and gratitude to entire faculty members of Department of Computer Science & Engineering and all non teaching staff of Computer Science & Engineering without their support it would have been difficult to shape my research work .

I also express my gratitude to **Ms Nisha Yadav**, Librarian and **Mr. Anand Kumar** Asst.Librarian of P, K, University and also IT cell staff **Mr. Alok Khare** and his colleagues.

I have no words to express my gratitude to my family members and my better half for their love, support and constant encouragement during the entire course of my research work.

Date ……. **Malege Sandeep**

Place ……. **Enrolment No: 161588517363**

LIST OFABBREVIATIONS

S.No	Abb.	Full Form
1	AWS	Amazon Web Services
2	CC	Cloud Computing
3	NIST	National Institute of Standards and Technology
4	DTR	Docker Trusted Registry
5	BCO	Bee Colony Optimization
6	CPU	Central Processing Unit
7	CSPs	Cloud Service Providers
8	EC2	Elastic Compute Cloud
9	SLA	Service Level Agreement
10	PoR	Pseudo-Randomization
11	ACL	Access Control Lists
12	DAF	Data Access Frequency
13	GDPR	General Data Protection Regulation
14	HIPAA	Health Insurance Portability and Accountability Act
15	ML	Machine Learning
16	ABC	Ant Colony Optimization
17	VPN	Virtual Private Network
18	ROI	Return On Investment

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

ABSTRACT

Multi-cloud setups are now necessary for enterprises looking to improve performance, agility, and cost-effectiveness by using many cloud service providers in today's quickly changing digital ecosystem. However, the use of multi-cloud architectures brings with it a number of new difficulties and problems, such as interoperability, data management, security, and cost optimisation. This paper assesses several methods for accomplishing thorough multi-cloud management and analyses the significant difficulties encountered in managing end-to-end operations in a multi-cloud environment.

The goal of this study is to provide a framework that will assist organisations in managing multicloud more successfully. It will include all the important topics, including performance, cost optimisation, security, management, and management, and it will assist the organisations in reaching their goals. There are several operational problems associated with multicloud administration, including security, compliance, and governance, as well as monitoring, management, optimisation, and performance. There are several challenges associated with operationally managing multicloud, including the need for centralised monitoring tools, application portability, workload (task) management, resource management and optimisation, service cooperation, build and deployment, etc. Having a centralised administration platform is very challenging since various service providers have different operating models, services, and APIs for controlling their cloud. In recent years, there has been a lot of study conducted in this area. A lot of businesses are concentrating on implementing best practices and standards for multicloud management. In recent years, many tools like as RightScale, Scalr, Clickr, and others have been developed to tackle these challenges in multicloud operations. However, these technologies do not provide a comprehensive solution for multicloud administration and do not cover all elements of issues like resource optimisation, consolidation, migration, orchestration, monitoring, and so on.

This provides incentive for doing research to create a comprehensive framework that oversees end-to-end multicloud operations and resolves the majority of the issues raised in the previous section on multicloud environment management. The suggested framework consists of five operational phases, each of which tackles a particular

multicloud management difficulty and provides a solution to satisfy the business goals of the organisation. An orchestrator technology called Kubernetes is used to facilitate multicloud cooperation and activity orchestration. A centralised monitoring system is also offered to keep an eye on all of the resources in the participating clouds. After analysing the observed metrics, a selection of important resources from each cloud platform is examined in further detail.

The study results provide valuable insights into cutting-edge methodologies and optimal approaches that facilitate the administration of multiple clouds for organisations. These include enhanced operational control, streamlined processes, and regulatory compliance, all while optimising cloud performance and minimising expenses. The thorough analysis of these methods provides helpful advice on how businesses may effectively handle the difficulties presented by multi-cloud ecosystems.

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Multi-cloud strategies are becoming more popular as a means for enterprises to take advantage of many cloud service providers in the ever-changing world of cloud computing. In a multi-cloud setup, private clouds like OpenStack coexist with public ones like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. Businesses may cut expenses, boost performance, assure redundancy, and increase flexibility using this method. Nevertheless, a thorough management structure is required to handle the myriad operational issues brought about by managing a multi-cloud system.

Resource management is a major obstacle in the way of effective multi-cloud administration. Because different cloud providers have varied offerings, price structures, and performance metrics, it's important to manage and distribute resources properly. Complex algorithms are required for resource optimization and workload scheduling.

When it comes to managing several clouds, security and compliance are paramount. Securing infrastructure while complying with numerous legal frameworks is a problem for enterprises whose data and workloads are dispersed across several cloud environments. Data breaches, illegal access, and compliance violations are more likely to occur in multi-cloud settings due to the decentralized structure of these systems, which makes it harder to implement security measures uniformly across all platforms. It is already difficult to execute a single security policy when dealing with many cloud providers, each of which may have its own set of regulations on data encryption, access control, and protection.

In the end, businesses' capacity to utilize the appropriate combination of technologies, approaches, and strategies determines the effectiveness of multi-cloud management. Businesses may conquer the difficulties of managing end-to-end operations in multi-cloud settings with the help of automation, orchestration, and analytics powered by artificial intelligence, and unified monitoring systems. These methods enhance

security, compliance, performance, and cost-efficiency while simplifying multi-cloud administration. In order to keep up with the ever-increasing usage of multi-cloud, companies need to develop all-encompassing management strategies that let them make the most of their cloud expenditures while reducing risks and inefficiencies.

1.2 CLOUD COMPUTING

Cloud computing (CC) is typically seen as a computing paradigm, including several computers linked to private or public networks to deliver dynamic, scalable infrastructure for applications, data, and file storage. It is also known as Internet Computing, as all hardware and software resources are customized to meet the requirements of end-users. It is a system that offers a collection of shared computer resources, encompassing diverse applications, deployment locations, storage, and networking capabilities. Cloud computing has three primary applications: desktop computing, which involves hardware, operating systems, and software accessed over the Internet instead of being maintained locally. [1]

Cloud computing is expansive and temporary; nonetheless, it essentially mirrors virtualized third-party hosting. The term "cloud" in "cloud computing" denotes the Internet; hence, users of cloud computing services share computing resources through Internet-connected devices. It encompasses a diverse array of fundamental concepts, including distributed computing, grid computing, virtualization, mainframe computing, and utility computing. Cloud computing is an increasingly prevalent concept across several contemporary domains, including the information technology business, academic institutions, and beyond.

Diverse scholars have articulated varying definitions of cloud computing from distinct application perspectives, although a consensual definition remains elusive. Among the several explanations, one commonly recognized definition provided by NIST is: Cloud computing is a paradigm that enables users to readily access shared pools of customizable computer resources as needed. These resources may be rapidly deployed and released with minimal administrative work or interaction with service providers. Both businesses and consumers derive significant advantages from cloud computing, as it obviates the necessity for acquiring and installing expensive software and hardware. Infrastructure and services are available for rental on an as-needed basis. In

the event of data loss on your hard drive due to a malfunction, you may back it up to the cloud and retrieve it from any Internet-enabled device at any time. Cloud computing is utilized efficiently across government entities, corporations, public and private enterprises, and research institutions. Cloud computing offers several advantages, including scalability, flexibility, efficiency, reliability, and speed. The cost of maintenance is reduced. Dropbox exemplifies a premier cloud environment, enabling users to effortlessly exchange files and images with anyone, irrespective of their Dropbox account status. Figure 1.1 illustrates the fundamental architecture of cloud computing.

FIGURE 1. 1 Basic structure of cloud computing

(Source: Secondary source taken from https://www.researchgate.net/figure/Basic-Structure-of-Cloud-Computing_fig1_329880577)

1.2.1 DEPLOYMENT MODEL OF CLOUD COMPUTING

Cloud deployment models illustrate the specific sort of cloud environment, characterized by ownership, scale, and accessibility. Four unique deployment methods are available, from which the customer can select one that aligns with their site requirements. [2] Cloud deployment models clarify the reason and nature of the cloud. Each deployment models specifications are explained below.

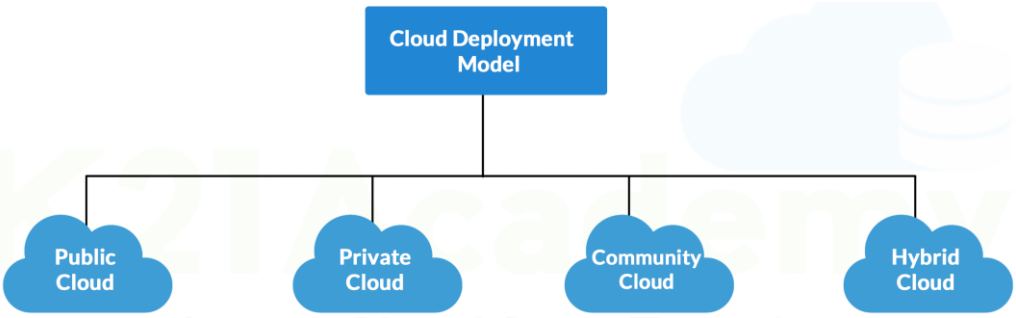


FIGURE 1. 2 Cloud Deployment Models

(Source: Secondary source taken from https://k21academy.com/cloud-blogs/cloud-computing-deployment-models/)

Private Cloud

These services are utilized by individual systems, requiring the user to manage their server and infrastructure to mitigate costs. Although it is not an economy, it provides significant value from a security standpoint. This sorting model is the optimal selection for managing sensitive data. The primary advantages of the private cloud are dependability, security, and virtualization. VMware exemplifies a private cloud by offering a virtualized infrastructure of several virtual servers, allowing clients to incur costs solely for the equipment, licensing, and maintenance.

Public Cloud

The cloud services are provided for public usage by the community. It is characterized by the accumulation of groups, scholars, institutions, and similar entities. The capital utilization overhead is reduced in the public cloud, hence achieving economies of scale. Google serves as the quintessential example of the public cloud, providing services either at no cost or through a pay-per-use model. The principal advantage of the public cloud is its scalability, flexibility, ease of use, and geographical independence; yet, it offers a diminished level of security.

Hybrid Cloud

This strategy hosts sensitive data in a private cloud while presenting shared data and use on a public cloud. It offers protection and comfort to the client. The primary advantages of the hybrid cloud are flexibility, scalability, security, and economies of

scale. Microsoft Azure exemplifies a mixed cloud. The virtual server is managed by this entity, while the private cloud oversees data security. It establishes a policy for compensating the utilization of these services.

Community cloud

It is mostly utilized for a certain segment of a company's clientele. Security standards and policies are disseminated throughout several enterprises. It is a private cloud, although a substantial cohort of clients shares the specifications, therefore diminishing expenses. Government agencies may exchange policy, compliance, and supplementary security elements inside a social cloud environment.

1.2.2 CLOUD COMPUTING SERVICES

Contemporary industries require a diverse array of cloud services to fulfill their particular requirements. The National Institute of Standards and Technology (NIST) provides three principal service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

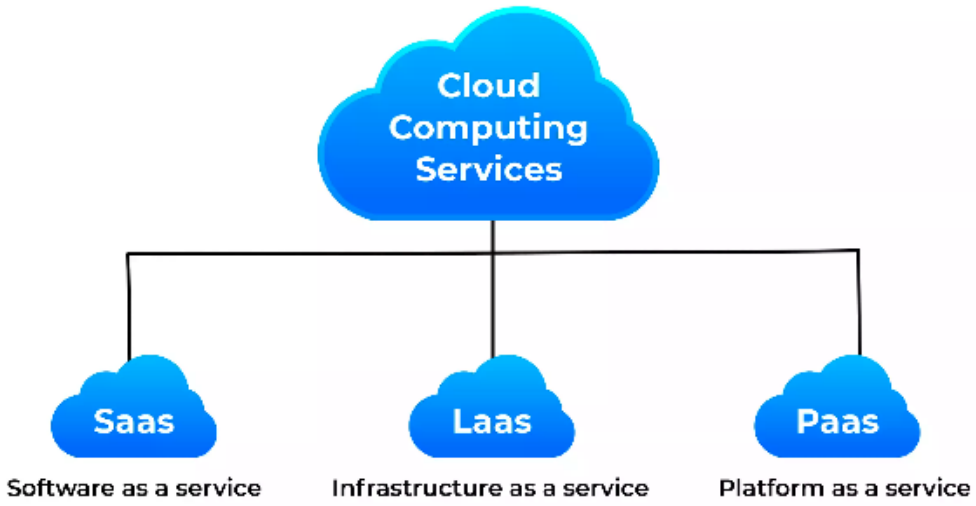


FIGURE 1. 3 Cloud computing services

(Source: Secondary source taken from https://timetraining.ae/learninghub-detail/what-are-cloud-computing-services)

SaaS

The SaaS layer occupies the pinnacle of the cloud architecture. It enables us to

execute the software online without the necessity of installation or maintenance. The complimentary Google service Gmail is an outstanding Software as a Service (SaaS) application. The SaaS software finances its services through free or paid subscriptions. They remit a fixed monthly fee per user and possess the ability to modify services at their discretion. It reduces maintenance costs and is scalable, flexible, and rapidly updatable. Services such as Google Apps, Dropbox, and Sales Power, which manage customer connections, are also widely recognized.

PaaS

The cloud service architecture includes this intermediary layer. A database, operating system, and runtime environment constitute the computing platform provided to the client. Customers need not concern themselves with expenditures related to downloading or installing this essential infrastructure, as these services are perpetually available to them. This will be highly beneficial for software engineers, testers, and deployers. For instance, in the case of Google Apps Engine, developers may occasionally require browser access to the service provider's standard software and tools. It manages scalability, flexibility, backup, and fault tolerance.

IaaS

This is the bottom rung for cloud service providers (CSPs). Networking, storage, and operating systems may all be provided via cloud computing to customers who are ready to pay. Small firms often struggle to afford the infrastructure needed for data centers. The IaaS provider leases these service components to the company. Usage expenses are mainly determined by three metrics: CPU utilization, data storage, and network bandwidth utilization. Virtual machines allow the customer to access the Infrastructure as a service environment. Key features of the IaaS provider include the elimination of capital expenses, the provision of variable volume, and the enhancement of reliability and stability. There are many more examples, including as EC2, Azure, etc.

1.2.3 CLOUD COMPUTING CHARACTERISTICS

The cloud makes a wide number of features accessible. A few of them are listed below:

On-demand services

The materials are available to consumers 24/7. They have a direct channel to the service provider and may add or cancel services online with ease. Members have the option to pay for all resources simultaneously or pay for only the resources they actually use. The price of the software might vary from one provider to another. Resources working together: In a cloud computing environment, data, programs, and hardware parts are stored on a distant server. While consumers have no idea which resources are available, many tenancy models are employed to assign and reassign them to customers based on their needs. The phrase "multi- lease technology" is used to explain scenarios in which several users unknowingly share the same ID resources.

Flexibility

Because their resources may be released or reallocated as needed, cloud providers can easily scale up or down their capacity to meet customer demand. In times of short-term resource crunch, companies might use the resources provided by CSPs to keep operations going. One advantage is the ability to scale up or down software components and other resources.

Quality of service

The service that consumers receive via cloud computing must be top-notch. Additionally, the availability of resources like as memory and bandwidth should be guaranteed in the service level agreement at all times.

Wide network access

Cloud computing is evolving in tandem with the web, making it possible for users to access cloud services from a wide range of devices including laptops, tablets, and smartphones. The utilization of private clouds inside specific organizations allows for broad network access, which in turn improves the utilization of public and hybrid clouds.

Pricing

In theory, there is no expense associated with cloud computing. We base our billing

on the amount of resources that our customers really use. Customers will be able to utilize them to monitor and track the correct usage of services after data on their use is recorded and monitored. It aids clients in understanding the advantages of cloud computing.

1.2.4 CLOUD COMPUTING ADVANTAGES

Cloud computing has several advantages. The following are some of the advantages.

- **Cost-effectiveness:** The savings in time and finances achieved via the utilization of more economical hardware and software is a significant selling proposition. Moreover, purchasers ought to compensate just for what they genuinely utilize. The quantity, quality, and duration of cloud service utilization all influence the total expense.

- **Scalability:** Purchasers might adjust the number of assets required according to client requests. It enables the utilization of cloud processing paradigms with adaptability or scalability.

- **Availability:** This means that services can be provided at any moment, even in the event of system maintenance or failure. Reliability, duplication, and error tolerance are all enhanced to accomplish this.

- **Adaptability:** Adaptability is achieved when users are able to move about freely, share resources, use their own devices, and work with various CSPs.

- **Mobility:** With cloud computing, we can access our resources from any location as long as we have an Internet connection.

1.2.5 APPLICATIONS OF CLOUD COMPUTING

Cloud computing is a cost-efficient, on-demand innovation that guarantees accessibility, flexibility, and minimal latency. Social networking, internet hosting, efficient distribution, and ongoing instrumented data preparation are integral components of basic and exceptional Cloud-supported software services. The fundamental prerequisites for development, organization, and transmission vary for each type of application. Clouds exhibit variable demands, supply configurations,

infrastructure dimensions, and resources; clients possess diverse, active, and quality of service requirements; applications have changing performance, workload, and dynamic scaling needs; consequently, evaluating the efficacy of provisioning solutions within a legitimate cloud computing environment for multiple applications in transient conditions is highly complex. [3]

Due to the inflexible nature of the system, it is frequently essential to utilize certified platforms, such as Amazon Elastic Compute Cloud (EC2) and Microsoft Azure, to evaluate application performance under certain conditions (accessibility, pending tasks). The expansion of dependable findings is an exceptionally noteworthy endeavor.

The task of reorganizing benchmark components throughout an extensive cloud computing infrastructure across many testing is both laborious and disheartening. Restrictive methods are executed utilizing pre-existing cloud-based scenarios and are outside the jurisdiction of client administration planners. Consequently, evaluating Cloud conditions in repeatable, reliable, and adaptable situations has demonstrated difficulty for benchmark testing.

Integrating cloud computing principles and standards into IT innovation initiatives may enhance cost-efficiency and agility for enterprises. To transition to a more energy-efficient enterprise, the substantial flood of data represents a dynamic enhancement for structural frameworks regarding exceptional loads that can facilitate critical calculation and administration. Given the overwhelming influx of information and the burden on its technological infrastructure, it is anticipated to do certain tasks.

Capacity planning for information and delivery can be conducted internally, through cloud computing, or via a hybrid approach of both methods. The advanced hardware and programming languages, development environments, and adaptable databases of the progressive cloud exemplify the dynamic and transformative influences in the realm of business software providers, open-source software choices, intellectual property legislation, antitrust governance, and international management entities.

Concerns over the safety and dependability of mission-critical and enterprise-level cloud services persist and must be solved for the cloud to remain a viable compensation alternative. The enthusiasm for corporate information innovation

frameworks, established and executed over the previous 50 years as part of information preparation, cannot be compared to these cloud breakthroughs. Cloud developments cannot be seen inherently better than web-based application retrofitting and framework programming.

SYS-ED professionals and CETi innovation partners are assessing the combination of programs that constitute the virtual infrastructure, secure and encrypted frameworks, and enhancement tools required to develop both private and public cloud environments.

1.2.6 CHALLENGES IN CLOUD COMPUTING

It is widely acknowledged that cloud providers should be the first to detect threats and implement measures to reduce the frequency of assaults. This would allow users to rely on the cloud with greater confidence. Everyday operations provide security risks, such as collecting logs, maintaining identities, organizing and accessing data, reporting issues, and transferring data across many sources. First, even while encryption is a great method for protecting sensitive data on storage servers, it doesn't guarantee that no one else won't be able to decipher it. A number of concerns about cloud systems have been brought up in previous studies. The prevalence of server-side assaults and the susceptibility of private cloud data are the root causes of these problems. Figure 1.4 clearly shows the primary obstacles. [4]

FIGURE 1. 4 Challenges in Cloud Computing

(Source: Secondary source taken from https://www.researchgate.net/figure/The-threat-in-Cloud-Computing_fig1_342068474)

Data Integrity

Among the many challenges of cloud computing is ensuring the data's validity. According to the principles of data integrity assurance, the data that is saved and the data that is acquired from the server are nearly indistinguishable. Some examples of possible dangers to data integrity are as follows: In 2013, more than two million accounts were compromised on social networking sites including LinkedIn, Facebook, and Twitter. In 2014, hackers attacked Sony's PlayStation Network by renting a server using Amazon's Elastic Compute Cloud (EC2) service. In 2015, hackers gained access to over two million Amazon S3 accounts, including passwords. Although their validity has been questioned, there are complete cloud-based solutions to these concerns that have been presented. If one person can access data from several devices, it leaves the data more susceptible to corruption, so the argument goes. It is imperative that service providers implement digital signatures or comparable technology to ensure the integrity of the data kept on their systems. This continues to occur even though Amazon suggests encrypting data before storing it on an S3 server. Data might also be authenticated and kept intact through the use of alternative means such as public audits, encryption, and proof of readability protocols. When companies store information on the cloud, it is crucial that the data remains intact to prevent any losses.

Privacy

Cloud resources should take measures to ensure that sensitive information cannot be deduced from a user's access behavior. This principle is fundamental to the safety of systems hosted on the cloud. Data breaches may have devastating effects on organizations and individuals whose sensitive information is at risk, including financial records, medical records, and social security numbers. Hackers can possibly access and use any user data if they have the password to a service, including an email account. Searchable encryption technologies and cryptography allow for safe data encryption while ensuring that authorized individuals may still access the encrypted data. The system that was developed in 2009 and solely used homomorphic encryption had a significant operational and budgetary impact. Amazon S3 does not guarantee the security of user information. Cloud services like Gmail and Google Docs are so popular that any security breaches involving user data have become

common knowledge. Even though deduplication technology in cloud storage has reduced storage demands and prices, data remains susceptible due to the utilization of stored file hashes. The proof of ownership approach was proposed as a workaround for this issue by validating cloud users. By implementing role-based access and privileges, it is also possible to confirm that a user has the permission to access or modify the data, or that the data belongs to the correct owner. By combining the two methods, we may reach our objective. Data is kept across several servers using multi-cloud technology, and separate clouds may connect with one other, significantly lowering security concerns.

Availability

A service's availability is the probability that it will be available and operational at any given time. Maintaining client confidence and preventing revenue losses due to service level agreement (SLA) violations are both made possible by high availability. In order to ensure that their servers are constantly operational, CSPs employ strategies such as redundancy and hardening. In addition to immediately reducing profitability, service disruptions have a negative impact on the customer experience. Amazon and other CSPs include a provision in their service level agreements (SLAs) that state explicitly that service interruptions are possible. In the event that a customer's storage data exceeds the limit or if payment for services is delayed, a temporary suspension of services may be experienced. Service providers frequently employ a cluster of servers for data backup and load balancing purposes. This is the procedure followed to resolve issues such as broken web servers or system mistakes. Every major cloud service provider, including Google Apps, Amazon S3, Amazon EC2, and Amazon Web Services (AWS), has had outages.

1.3 CLOUD STORAGE

A kind of distributed computing, "cloud storage" involves a number of machines storing data in what are called "logical pools," as seen in figure 1.5. In order to store customer, corporate, or application data, businesses pay third-party providers for storage space. [5] Thanks to its many advantages, cloud storage has recently emerged as a serious competitor to traditional on-premises storage. This surge in popularity has transpired during the past few years.

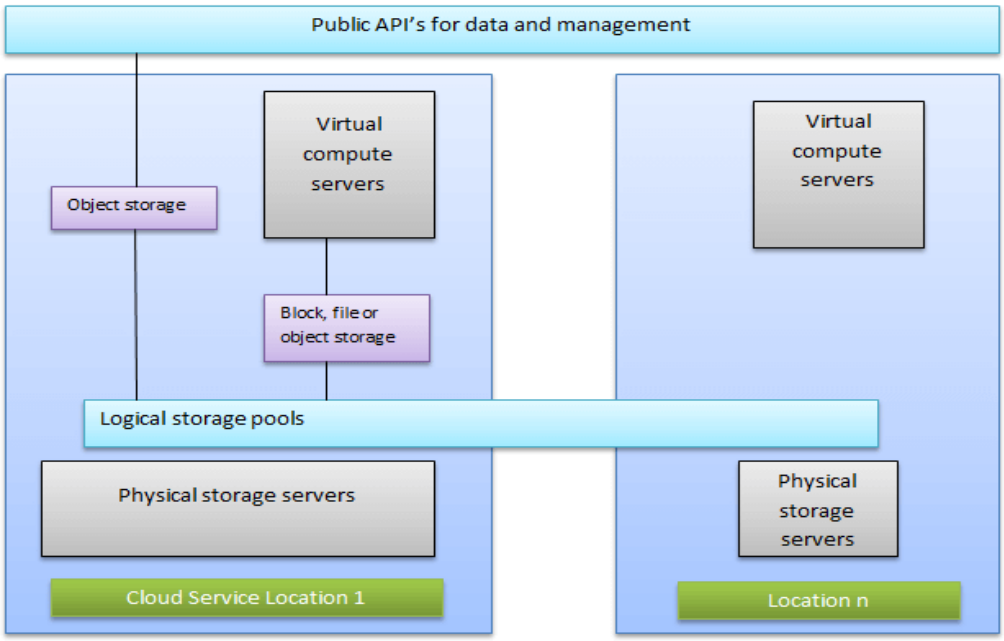


FIGURE 1. 5 Cloud Storage Architecture

(Source: Secondary source taken from https://electricalfundablog.com/cloud-storage-architecture-types/)

There isn't a universally accepted set of characteristics for cloud storage architecture, and many cloud storage systems use a variety of architectural styles. Still, most cloud storage systems really use thousands of separate storage devices. A distributed file system, network, and other storage middleware combine these devices to provide cloud storage services to its customers. Most cloud storage designs include a storage resource pool, SLAs, distributed file systems, service interfaces, and maybe other parts as well. The primary goal of cloud storage systems is to provide scalable, on-demand storage that can accommodate numerous tenants. A common component of cloud storage architecture is an API front end, which allows users to access data stored in the cloud. It is becoming more common to see file service front ends, web service front ends, and even more conventional interfaces like tape library front ends (iSCSI and internet SCSI) in cloud computing environments, although this API is more commonly known as the SCSI protocol in conventional storage systems. Located beneath the application's front end is the storage logic layer of the middleware. Data reduction and duplication are two of the many operations that this layer is responsible for carrying out. When it comes to cloud storage, the actual data storage is handled by the backend. Either a traditional back end for physical disks or

an internet protocol with particular properties might make up the back end. The responsibility for storing data lies with the cloud storage infrastructure.

1.3.1 TYPES OF CLOUD STORAGE

Similar to cloud computing, cloud storage operates on a virtualized infrastructure and offers scalable capacities, accessible interfaces, and measurable resource utilization. We use Amazon Simple Storage Service (S3), an external provider of cloud-based storage solutions. The original meaning of the phrase was a hosted object storage service, but it has now been expanded to include other kinds of data storage that are presently offered as a service, such as block storage.

Storage solutions that can be hosted and deployed with cloud storage qualities include object storage software like Open Stack Swift and object storage services like Amazon S3, Oracle Cloud Storage, and Microsoft Azure Storage. There are four main categories of cloud storage that are defined by the storage and retrieval features offered by each:

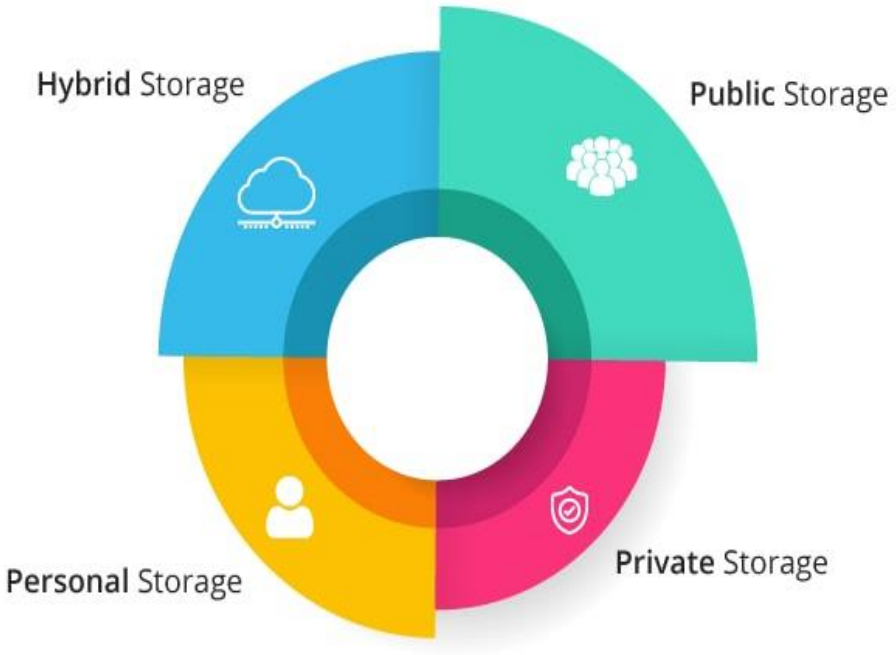


FIGURE 1. 6 Types of Cloud Storage

(Source: Secondary source taken from https://www.blomp.com/4-types-of-cloud-storage-services/)

- **Personal Cloud Storage:** This kind of cloud storage is a subset of public cloud storage that lets users access their data from anywhere in the globe. It also facilitates the sharing and synchronization of data between different devices. For private cloud storage, consider Apple's iCloud.

- **Public Cloud Storage:** on this model, the company and storage provider are separate entities, and the enterprise's data center is not used to store data on the cloud. The whole of the company's public cloud storage is the responsibility of the cloud storage provider.

- **Private Cloud Storage:** The firm's data center houses components from both the cloud storage provider and the company. Even if private clouds assist resolve potential performance and security concerns, you may still get the advantages of cloud storage by switching to this model.

- **Hybrid Cloud Storage:** This hybrid model combines public and private cloud storage; the organization's private cloud houses critical data, while the public cloud stores less critical data.

1.3.2 ADVANTAGES AND CHALLENGES OF CLOUD STORAGE

There are a plethora of options for storage, including a plethora of different categories. Online communication and digital photos have their own special storage mechanisms.

Other methods of storage are more generalizable. Any kind of data may be stored in one of the existing systems.

Depending on your needs, you may choose both large and small cloud storage solutions. There are several benefits and drawbacks to data management that are related to the storage qualities of the data. [6]

Cloud data management has the following benefits:

- **Off-Site Backup:** Businesses and organizations may now affordably establish off-site copies of their data using cloud storage, sometimes called remote backups. This is a significant improvement over old backup techniques.

- **File Accessibility:** Access to data kept in the cloud is made possible over the internet for everyone.

- **Security of Data:** Data stored in the cloud is protected from hackers thanks to authentication methods that are put in place before data access is granted.

- **Effective Use of Bandwidth:** Data transport is made unnecessary when stored in the cloud. One example is the potential replacement of sending huge files by email with just forwarding a link.

- **Ease of Management:** When you use a cloud application, managing the underlying software, hardware, and network becomes much easier. Cloud storage allows customers to access their data from any computer with an internet connection, while the supplier handles all the technical aspects.

- **Cost-effectiveness:** Businesses may save money on operational costs with cloud storage services since they eliminate the need for costly hardware and staff to maintain such systems. Not only are cloud services scalable and available at all times, but they are also the most cost-effective option.

Here are some of the difficulties associated with cloud storage:

- **Vendor Lock-in:** One of the main obstacles to widespread use of cloud computing is vendor lock-in, which is caused by an absence of standards. When consumers are stuck with one cloud storage provider and risk price increases, decreased availability, or the firm's bankruptcy, this is called vendor lock-in. Users are reluctant to switch providers due to the prohibitive cost of bandwidth, even if the new provider offers better service or cheaper prices. Therefore, customers are put in a difficult situation if a provider changes the prices. It also takes a long time to move large amounts of data across different cloud service providers.

- **Data unavailability:** To what extent cloud computing will keep growing is largely dependent on how readily available services are. Data availability is one of, if not the most important, indicators for customers. While cloud service providers (CSPs) do their best to ensure that their services are available at all

times, there is always the chance that anything beyond of their control might cause them to go down. Some instances of such unplanned events include server downtime, natural disasters, power outages, and other similar situations. The August 8, 2016, outage of Google Cloud Storage and File Backup Server caused users to lose a substantial amount of money. Regrettably, Alibaba Cloud's domain name resolution became ineffective due to unexpected and huge traffic assaults that occurred in the afternoon of December 7th, 2017. Huge monetary losses may occur if customers couldn't use their services.

- **Data Privacy Leakage:** Customers should exercise caution when dealing with untrustworthy cloud storage providers (CSPs), since their data is being kept by a third party. When users put all their data storage eggs in one cloud provider's basket, they leave themselves open to data breaches. The vast majority of data breaches happen when an untrustworthy CSP steals sensitive user information without the user's knowledge or permission. Data breaches can occur for a variety of reasons, including malevolent insiders at CSP who steal or corrupt the data and external attacks that compromise data privacy.

1.4 MULTI-CLOUD COMPUTING

The multi-cloud architecture, which pools resources from several clouds (hosted by different organizations) to perform a single operation, offers a potential answer to the security issues surrounding cloud computing.

The two-tiered architecture known as multi-cloud is also dubbed cloud-of-clouds or inter-cloud. One of the primary tiers is the initial one, called intra-cloud.

The second layer, inter-cloud, is a sophisticated level that uses several cloud interactions. Due to the novelty of multi-cloud, there is less extensive study on it compared to single clouds.

Nevertheless, previous studies have demonstrated that by utilizing several clouds, when many clouds are used instead of just one, some risks may be mitigated or even removed. [7]

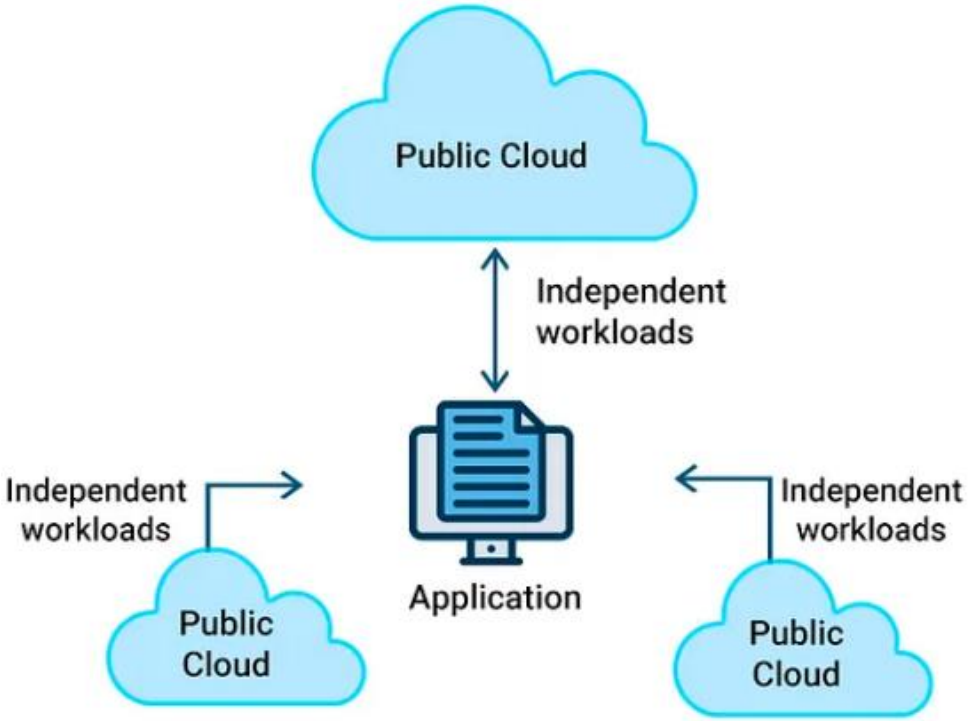


FIGURE 1. 7 Multi-Cloud computing architecture

(Source: Secondary source taken from https://medium.com/@haripriyasrikanth0/hybrid-cloud-strategies-with-azure-arc-my-practical-insight-ba4754e387b2)

1.4.1 MIGRATION FROM SINGLE CLOUD TO MULTI-CLOUD

Researchers have been focusing more on multi-cloud systems as a result of the present uncertainty surrounding single-cloud solutions. Eighty percent of company managers are most concerned about the likelihood of system failures, downtime, and data loss, according to studies on the downsides of cloud computing. This is because, maybe due to the unsatisfactory security offered by individual cloud providers, more and more clients are choosing multi-cloud solutions. In order to solve many challenges associated with cloud computing and offer reliable services, the ICStore architecture was developed. Factors to think about are consistency, reliability, honesty, and secrecy. The RACS method reduces data transmission costs by transparently dispersing data among several cloud storage providers. [8]

Doing data replication and storing it in several clouds can further reduce the likelihood of server failure and data loss. The approach uses RAID technology to

maintain the file system. The HAIL policy controls the accessibility, integrity, and security of files across various storage systems or servers by utilizing two basic cryptographic methods. Among these methods are public key cryptography (PDP) and pseudo-randomization (PoR). In the case that fault detection uncovers problems, one of the building components, PoR, is utilized to assess and reallocate storage resources.

1.4.2 SECURE AND DATA ACCESS ISSUES IN MULTI-CLOUD COMPUTING WITH INTER-SERVER COMMUNICATION SECURITY

Cloud computing is seen as a prospective paradigm for computing that provides online services such as software on demand, platforms, and infrastructure. Organizations and people favor it for its inexpensive, scalable, and remotely accessible capabilities; yet, data security, user privacy, and access control remain pressing issues in the cloud environment.

Upon outsourcing the data, the data owner relinquishes control over it. Servers are typically equipped with Access Control Lists (ACL) to authorize user access based on their identification numbers.

The server may inadvertently or intentionally grant data access to unauthorized users. In the past, fine-grained access control systems were implemented to resolve this issue.

Attribute-Based Encryption, generally known as the ABE method, was first introduced by Sahai and Waters in 2005. ABE now serves as a comprehensive solution to the access control issue.

It fundamentally links attributes designated to the user's private keys. ABE is an augmented iteration of the fundamental IBE system. ABE may be categorized as KP-ABE (Key Policy Attribute Based Encryption) and CP-ABE (Cipher text Policy Attribute Based Encryption).

Traditional encryption methods, including identity-based encryption and public key encryption, are designed for the direct ciphering of data pertaining to a single user. However, these solutions are inadequate for one-to-many encryption applications.

The CP-ABE technique, facilitates a flexible one-to-many encryption model that has

established itself as a standard for fine-grained access control over encrypted data. This approach allows a user to establish an access policy for their encrypted data, with keys provided by an authority enabling privileged users to access the data depending on their qualities.

Consequently, only if the receiver's attribute set complies with the access policy established by the data owner will they be allowed to get the encrypted data. However, a significant issue persists, namely attribute revocation that requires attention.

User ID attributes are not fixed and may often change; hence, new attributes can be added, existing ones altered, or old attributes withdrawn. Consequently, a flexible attribute management method is essential for real-time CP-ABE deployment.

CP-ABE methods are already typical for cloud systems; nevertheless, multi-cloud environments require an enhanced level of security due to their infrastructural features. Multi-cloud storage systems mitigate several dangers associated with single cloud storage systems by virtue of their ability to store data across multiple servers and facilitate inter-cloud data communication.

1.5 MULTI-CLOUD ENVIRONMENT

Recently, multi-cloud storage has effectively mitigated the hazards associated with regional providers while offering additional benefits such as satisfactory responsiveness, enhanced load balancing, and swift data recovery. In the realm of multi-cloud storage, users prioritize several factors, with cost-effectiveness and high availability being paramount. There exists a discrepancy in the pricing of same services across different suppliers, each of whom offers a distinct service while upholding equivalent competency. Performance correlates with pricing. If consumers seek to enhance data availability, they must incur a larger cost. [9]

When you employ resources and services from more than one cloud provider, you're engaging in multi-cloud computing. Many times, all it takes to do this is to use software-as-a-service (SaaS) from a cloud provider like Salesforce or Workday. The term "multi-cloud" is most often used in the business world to describe the use of PaaS and IaaS from many cloud providers, such as Microsoft Azure, IBM Cloud,

Google Cloud Platform, and Amazon Web Services (AWS), as illustrated in figure 1.8.

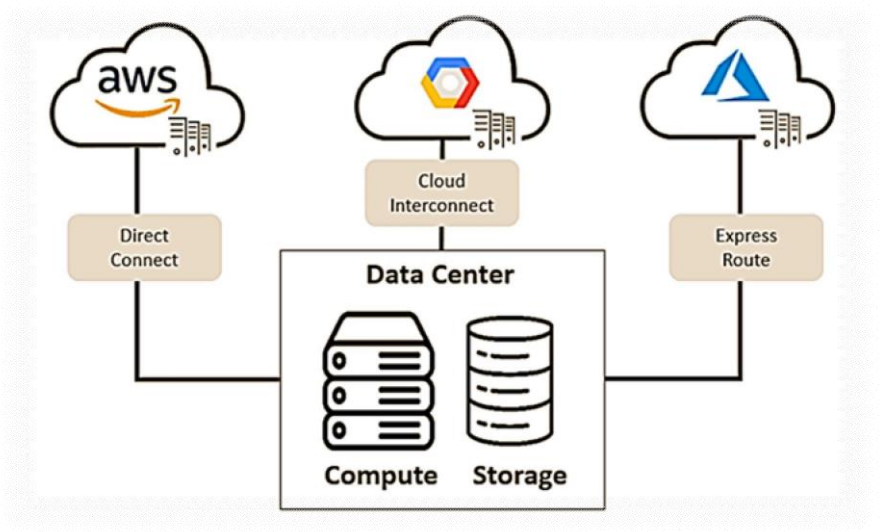


FIGURE 1. 8 Multi-cloud Environment

(Source: Secondary source taken from https://www.factioninc.com/blog/what-is-multi-cloud/)

Cloud computing platforms that are compatible with many cloud providers' infrastructures are known as multi-cloud solutions. It is usual practice to construct multi-cloud solutions using open-source and cloud-native technologies, such as Kubernetes, which are offered by all public cloud providers. More than that, though, they often come with the ability to manage workloads across several clouds from a unified interface—a "single pane of glass" type of thing. Numerous leading cloud providers, Together with cloud solution providers such as VMware, we now provide all-encompassing multi-cloud solutions for a wide range of IT demands, such as development, data warehousing, cloud storage, artificial intelligence, machine learning, disaster recovery, business continuity, and more.

1.5.1 HYBRID CLOUD VS. MULTI-CLOUD STORAGE

There are benefits and drawbacks to using both hybrid clouds and multi-cloud setups in a business setting. The two names are often confused and used interchangeably. With many businesses already using multi-cloud computing, it's reasonable to assume that this divergence will grow. As is often acknowledged, the multi-cloud strategy

makes extensive use of cloud services, the majority of which are supplied by many independent cloud solution providers. By taking this route, companies have a better chance of finding cloud-based solutions that work for their various divisions. Parts of a hybrid cloud, in contrast to a multi-cloud setup, frequently work together. Hybrid cloud environments are characterized by the mixing and communicating of processes and data, as opposed to multi-cloud systems, which operate autonomously as separate silos.

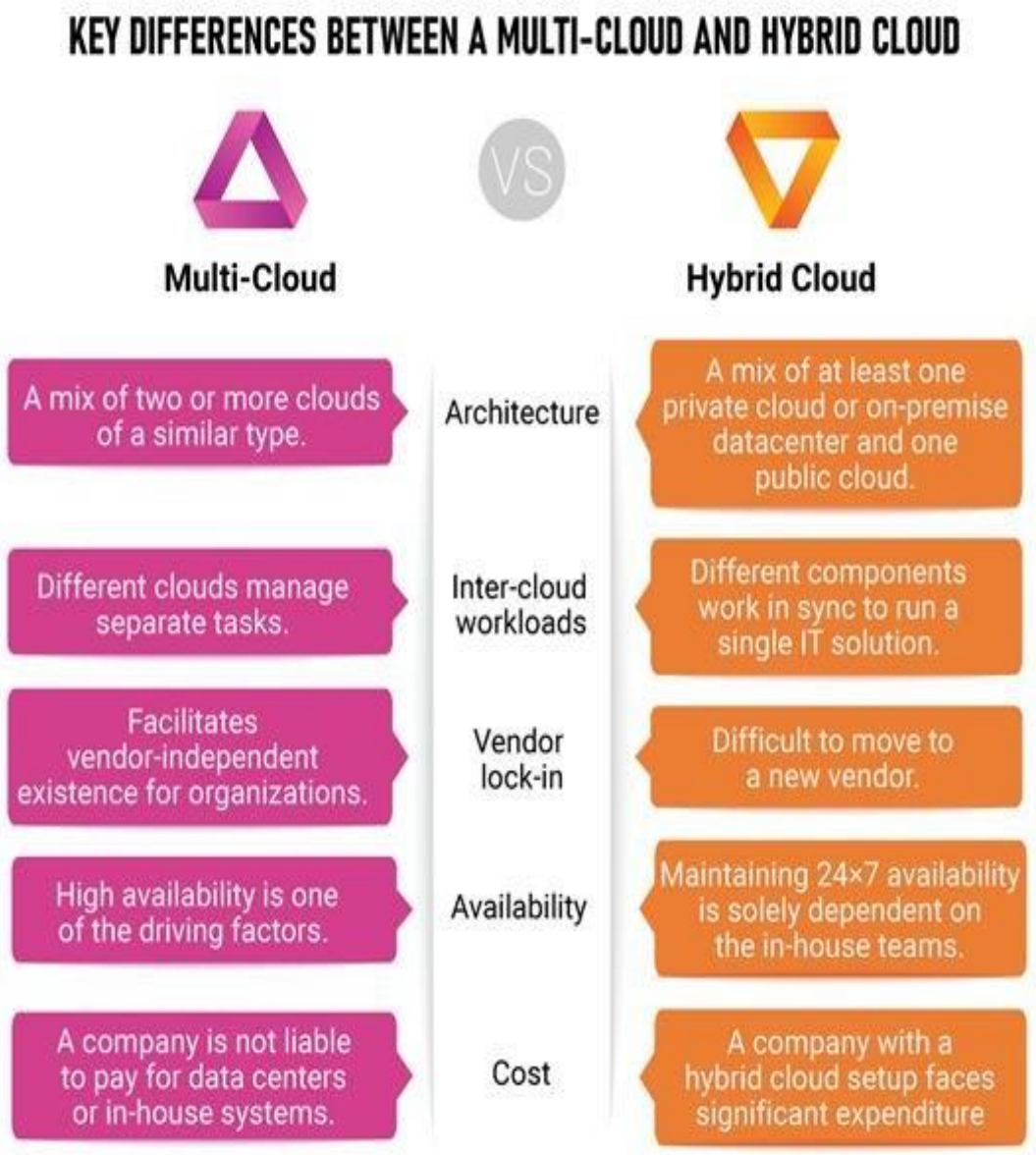


FIGURE 1. 9 Hybrid Cloud vs. Multi-cloud

(Source: Secondary source taken from https://www.spiceworks.com/tech/cloud/articles/multi-cloud-vs-hybrid-cloud/)

It is usual for applications on hybrid cloud systems to use load balancing in combination with web services and public cloud apps. A private cloud environment allows for the simultaneous hosting of databases and data storage. Any number of operations is possible using the resources housed in the cloud-based system, just as they are in private and public clouds.

A multi-cloud architecture allows for the management of an app's networking and computation needs by one cloud provider, with data storage and retrieval handled by another provider's database. In multi-cloud configurations, Azure's resources may be accessible to certain applications only.

However, there are certain apps that might rely solely on AWS's resources. Applying this principle in tandem with a combination of public and private clouds is another example.

Some applications may need access to resources only available in public clouds, while others may need access to resources only available in private clouds.

Despite their many differences, cloud-based services essentially allow companies to better meet the demands of their consumers in terms of both efficiency and effectiveness.

1.5.2 DATA MANAGEMENT IN MULTI-CLOUD STORAGE

With the promise of better data accessibility and reduced maintenance costs, the number of people using the cloud to store their data is growing. We shall simplify and abstract the use cases for cloud storage, as shown in Figure 1.10, for the sake of simplicity and clarity.

Clients have numerous needs while keeping objects in the cloud, including data accessibility, cheap cost, and high availability, among others. An additional facet of user needs is data access frequency (DAF), which quantifies the number of data requests made in a 24-hour period..

This is important since users' data often consists of shared files. This case study illustrates the risks of depending on only one cloud provider. To mitigate these dangers, multi-cloud storage distributes user data among several CSPs.

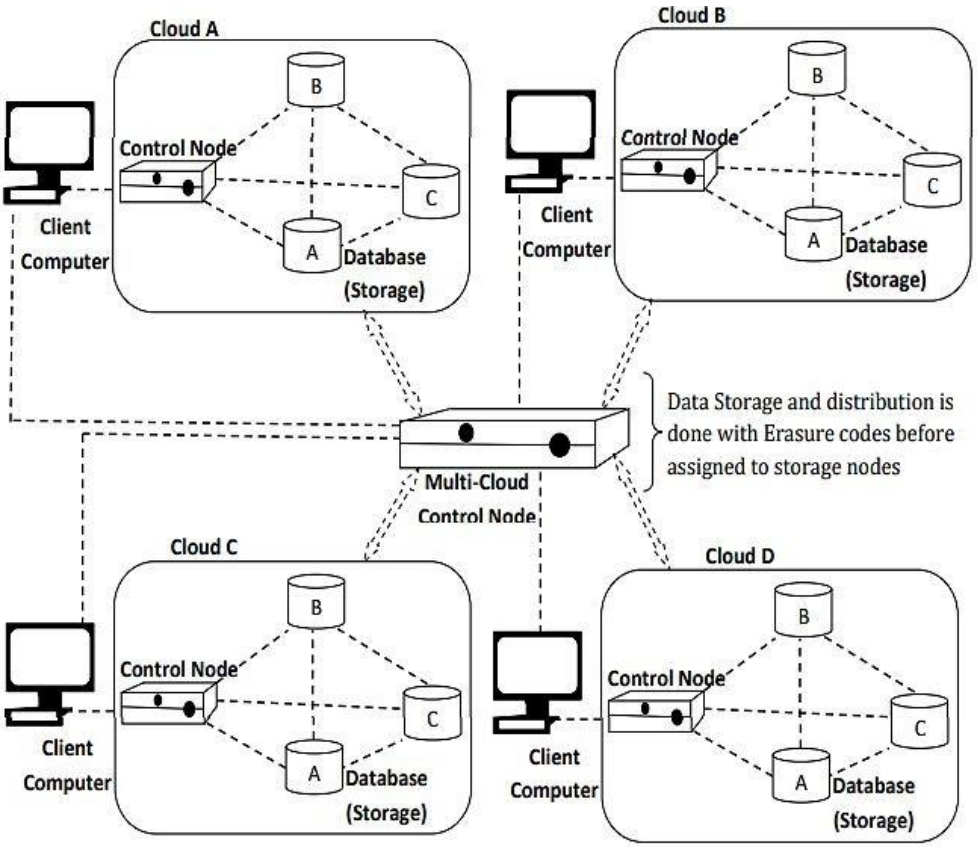


FIGURE 1. 10 Multi-Cloud Storage Architecture

(Source: Secondary source taken from https://www.semanticscholar.org/paper/Reliable-Multi-cloud-Storage-Architecture-Based-on-Mugisha-Zhang/38ddcc887c23ee2533228d0691dd9c70dfb80350)

- **Lowering data retrieving cost and avoiding vendor lock-in:** There are two components to the total cost of data retrieval: the GET request cost and the out-bandwidth cost. Customers may acquire their data for the lowest feasible out-band cost and request cost via erasure coding. CSPs.

- **Achieving high data availability:** Data stored in a single cloud service must have an availability rate of 99.99% or higher. This bar is much higher for the availability of SLA in many CSPs. Nonetheless, hearing that data centers belonging to many well-known CSPs have gone down is not shocking. Typical data dispersion mechanisms in multi-cloud storage include erasure coding and replication, as mentioned in the introduction. Replication may improve availability compared to erasure coding, but it requires a lot of space in the

database.

- **Protecting data privacy:** Erasure coding allows each CSP to retain just a portion of the user's initial data object. Regardless of the vector, an attacker compromising the cloud provider's network from inside or outside cannot use a data chunk to recreate the initial user object. Data stored in many clouds using erasure coding provides some level of security for user information.

Given the trade-off between data availability, retrieval cost, and storage cost, multi-cloud storage presents a significant challenge despite its advantages. The high expense of using high availability CSPs makes this an important consideration. Answering the topic of how to choose cloud storage providers and erasure coding settings, it optimizes data availability at the lowest feasible cost.

1.5.3 CHALLENGES IN MULTI-CLOUD ENVIRONMENTS

Complexity in Resource Management

It is quite complicated to manage resources across different cloud platforms. Allocating resources, scalability, and monitoring are all areas in which cloud providers (such AWS, Azure, and Google Cloud) differ from one another. The coordination of workloads becomes more challenging when IT workers are required to negotiate multiple interfaces, APIs, and management consoles. Furthermore, efficient automation and orchestration technologies are necessary for monitoring resources like as storage, computing, and networking services across several platforms in order to prevent duplication and maximize use. This complexity is sometimes made worse by the absence of consistent dashboards or interfaces, which in turn causes operational expenses to rise and efficiency to suffer.

Security and Compliance Issues

Due to the fact that different platforms have different security procedures, access limits, and monitoring capabilities, security is a major problem in multi-cloud setups. Particularly in domains such as encryption, network security, and identity management, it is difficult to provide consistent security procedures across clouds. Complex operations like incident detection and response become more difficult in

decentralized multi-cloud settings. Another difficulty is staying in line with industry rules such as GDPR and HIPAA, which might vary from one country or service to another in terms of what is legally required. Organizations should keep their security policies consistent across all cloud platforms and make sure that data handling and storage conform to jurisdictional standards.

Data Integration and Interoperability

Interoperability problems are common when integrating data across different clouds. Data may not be easily transferable across systems due to cloud companies' usage of proprietary technologies or formats. Regardless of the hosting environment, applications and services in multi-cloud architectures must interact properly. Database, storage, and communication protocol incompatibilities might cause activities to run more slowly and need bespoke work to fix. Because of this lack of standardization, data processing and transportation becomes more complicated, which can lead to data silos, performance issues, or platform-specific data inconsistencies.

Network Latency and Connectivity

When apps in a multi-cloud setup need to transfer data across data centers that are located in different physical locations, network latency becomes an enormous issue. Application performance and user experience might be affected by delays in data transfer caused by the physical isolation of cloud infrastructure. When moving massive amounts of data across clouds, connectivity problems might develop, which in turn causes reaction times to go down and operating expenses to rise? Furthermore, businesses need to put money into strong networking solutions so they can keep communication latency low, handle traffic spikes, and prevent outages that might harm mission-critical services.

Vendor Lock-In Risks

Getting stuck with one cloud provider is a real possibility when you use a multi-cloud approach. It may be difficult to switch cloud providers due to the uniqueness of each company's offerings, technology, and platforms. Because of the reliance on proprietary technologies or APIs, switching providers becomes a much more difficult and expensive proposition. Furthermore, application rewrites or reconfigurations may

be necessary and time-consuming when moving workloads or data across providers. Organizations must implement thorough backup plans to avoid becoming overly dependent on any one cloud provider, as the key objective of multi-cloud solutions is to maintain flexibility.

Monitoring and Performance Management

Due to the absence of a single monitoring solution across cloud platforms, it can be tough to monitor the performance of applications and services in a multi-cloud context. It is challenging to get a comprehensive picture of the infrastructure since different cloud providers provide different monitoring tools and performance data. Proactive performance management is impeded and real-time issue detection and diagnosis are made more difficult as a result of this fragmentation, which causes gaps in visibility. Organizations run the danger of downtime, service degradation, and operational inefficiencies that affect user experience if monitoring is not effective and streamlined.

Cost Optimization and Management

The complexity of managing costs in a multi-cloud setting surpasses that of a single cloud. Computing, storage, networking, and other services offered by cloud providers all have their own unique pricing models, which makes it hard to keep track of and compare prices. Avoiding hidden costs and keeping track of unused or underused resources across several clouds is a common concern for organizations. Unanticipated cost spikes can also result from workloads and scalability that are difficult to foresee. If you want to optimize your costs in a multi-cloud environment, you need to know how each provider charges and how you use their services. You should also use cost management tools that show you how much you're spending and where you can save money.

1.6 MULTI-CLOUD MANAGEMENT THROUGH DIFFERENT TECHNIQUES

To overcome the challenges of managing several cloud platforms, effective multi-cloud environment management necessitates the use of a wide range of approaches. For all-encompassing multi-cloud administration, you can use the following methods

and tools:

1.6.1 AUTOMATION AND ORCHESTRATION

When it comes to managing several clouds, automation is key. Organizations may streamline the process of creating, growing, and removing cloud resources from various platforms using automation technologies such as Terraform, Ansible, or CloudFormation. Regardless of their hosting location, orchestration platforms like Kubernetes make it easy to coordinate applications and services. Businesses may streamline their operations and save time and effort by automating repetitive processes and dynamically managing workloads. This way, they can decrease the need for manual intervention and ensure that all cloud environments have consistent settings.

1.6.2 AI/ML-BASED CLOUD OPTIMIZATION

Advanced capabilities in workload prediction, resource utilization optimization, and overall cloud performance enhancement are offered by artificial intelligence (AI) and machine learning (ML) techniques. By studying consumption patterns and suggesting strategies to decrease unnecessary or underused resources, these methods may be applied to optimize costs. Autonomous performance monitoring, efficient resource allocation, and demand forecasting are all capabilities of AI/ML-based solutions like CloudHealth and Turbonomic. Through real-time optimization of cloud consumption across several providers, this aids enterprises in achieving improved operational efficiency.

1.6.3 CONTAINERIZATION AND KUBERNETES

Containers have transformed cloud administration by offering a uniform and portable environment for programs. Kubernetes, as a premier orchestration solution, has enhanced the efficiency of managing containers across multi-cloud settings. Kubernetes enables developers to automate the deployment, scaling, and maintenance of containerized applications across various cloud environments, delivering high availability and resilience. The separation of applications from the underlying infrastructure allows enterprises to circumvent vendor lock-in, facilitating the transfer of workloads between clouds without necessitating reconfiguration.

1.6.4 INFRASTRUCTURE AS CODE (IAC)

Infrastructure as Code (IaC) is an essential methodology in multi-cloud administration, facilitating infrastructure delivery and management via code. Instruments like Terraform, Pulumi, and AWS CloudFormation enable enterprises to delineate and administer cloud architecture in a declarative manner. Infrastructure as Code (IaC) enables enterprises to provide version control for their infrastructure configurations, automate resource deployment, and consistently duplicate environments across various cloud platforms. This method enhances operational efficiency, minimizes human mistakes, and increases agility in the management of changing cloud settings.

1.6.5 MONITORING AND OBSERVABILITY TOOLS

Effective monitoring and observability are crucial for sustaining visibility in multi-cloud operations. Instruments such as Prometheus, Grafana, and Datadog offer instantaneous insights into the performance of applications and infrastructure across various cloud environments. These technologies facilitate the detection and resolution of issues by monitoring essential metrics, logs, and traces prior to user impact. Improved observability enables enterprises to sustain maximum performance, minimize downtime, and swiftly ascertain the core causes of possible issues, irrespective of the cloud platform utilized.

1.6.6 CLOUD-NATIVE SECURITY AND COMPLIANCE

Maintaining security and compliance across many cloud platforms poses a considerable problem. Cloud-native security solutions such as Prisma Cloud, Dome9, and Azure Security Center provide extensive functionalities to safeguard data, apps, and infrastructure inside a multi-cloud environment. These solutions facilitate centralized security administration, allowing for uniform rules for access control, encryption, threat detection, and vulnerability management. Automated compliance assessments guarantee that enterprises adhere to industry requirements, including GDPR and HIPAA, across all cloud environments. Moreover, zero-trust security frameworks and comprehensive identity management systems are critical strategies for protecting multi-cloud settings.

1.6.7 COST MANAGEMENT AND OPTIMIZATION TOOLS

In multi-cloud systems, managing costs may rapidly become burdensome owing to the varying pricing patterns of each provider. Instruments such as Cloudability, Flexera, and AWS Cost Explorer assist firms in overseeing and enhancing their cloud expenditures by offering real-time insights into resource utilization. These programs evaluate consumption statistics and propose improvements, like rightsizing instances, removing wasted resources, or securing more favorable pricing. By utilizing these technologies, firms may maximize the value of their cloud expenditures while circumventing superfluous costs.

1.6.8 CROSS-CLOUD NETWORKING SOLUTIONS

Interconnectivity among cloud platforms constitutes a sophisticated element of multi-cloud administration. Technologies such as Cisco Multi-cloud Fabric and Aviatrix provide safe, high-performance networking across many cloud providers. These technologies enable efficient data transmission, minimize network latency, and improve security across clouds. Network segmentation, traffic management, and virtual private network (VPN) configurations may be automated and centrally administered, guaranteeing dependable connection for multi-cloud applications.

1.7 STATEMENT OF AIM

The aim of this research is to analyze the key challenges posed by multi-cloud environments and to explore how managing end-to-end operations can provide a comprehensive framework for effective multi-cloud management. This study seeks to identify the complexities involved in operating across multiple cloud platforms, including issues such as resource management, security, data integration, performance monitoring, and cost optimization. By leveraging a variety of techniques, such as automation, AI/ML-based optimization, containerization, Infrastructure as Code (IaC), and cloud-native security, the research aims to develop strategies that can streamline operations and address the inherent difficulties of managing a multi-cloud infrastructure. The ultimate goal is to propose solutions that enable organizations to enhance the efficiency, flexibility, and resilience of their multi-cloud ecosystems while mitigating operational risks and improving overall performance.

Therefore, we entitle our study is "*ANALYSING MULTI-CLOUD CHALLENGES BY MANAGING END-TO-END OPERATIONS FOR COMPREHENSIVE MULTI-CLOUD MANAGEMENT THROUGH DIFFERENT TECHNIQUES*"

1.8 NEED AND SCOPE OF THE STUDY

Need of the study

As organizations increasingly adopt multi-cloud strategies to leverage the unique capabilities of various cloud providers, managing these complex environments effectively becomes crucial. The need for this study arises from the growing challenges associated with end-to-end operations in multi-cloud settings, where traditional management practices may fall short. Organizations face difficulties in coordinating resources, ensuring consistent security and compliance, integrating data across platforms, and optimizing costs in a fragmented cloud landscape. Moreover, the rapid evolution of cloud technologies and the increasing sophistication of cyber threats further exacerbate these challenges. This study addresses the urgent need to understand and navigate these complexities by analyzing current multi-cloud management practices and exploring innovative techniques that can enhance operational efficiency. By developing a comprehensive framework for managing end-to-end operations, the research aims to provide actionable insights and strategies that can help organizations effectively tackle multi-cloud challenges, ensuring robust, scalable, and cost-effective cloud solutions.

Scope of the study

Due to the growing complexity encountered by enterprises using multi-cloud strategies, research into the difficulties of multi-cloud end-to-end operations management is essential. Integrating cloud services seamlessly, keeping security and compliance strong, maximizing performance, and successfully controlling costs are important challenges that organizations face as they try to take advantage of the cost savings, resilience, and flexibility of different cloud services. To help enterprises get the most out of their multi-cloud investments, this report identifies and addresses these obstacles, providing vital insights and solutions. On top of that, it helps push cloud computing forward by advancing techniques and technology that make cloud

environments more safe, efficient, and scalable. To improve operational efficiency, competitiveness, and innovation capability in the digital economy, this research is crucial for enterprises to traverse the complexity of multi-cloud management.

1.9 OBJECTIVES OF THE STUDY

Following are the main objectives of this study.

1. To describe the importance of providing a ranking process in selecting the right cloud for existing workload migration
2. To gives overview on multi-cloud strategy for addressing the common business challenges, lack of process standardization and cloud migration
3. To discuss different procedures for resource optimization within cloud using Markov Decision process and Re-enforcement learning
4. To design a scheduling algorithm for optimal workload management.
5. To design and prototype a unified management platform that integrates various tools and techniques for comprehensive multi-cloud management.

1.10 DEFINITIONS OF THE KEYWORDS

Cloud computing

Computing in the cloud refers to the practice of delivering various types of computer resources, such as servers, storage, databases, networking, software, and analytics, over the internet ("the cloud") in order to facilitate scalability, agility, and access to new ideas. Cloud services offered by third-party suppliers such as (AWS), Microsoft Azure, and (GCP) allow enterprises to lease rather than own and manage their technology and infrastructure.

Cloud Storage

What we call "cloud storage" is really just storing data on faraway computers and making it accessible online. Cloud computing companies offer consumers and businesses the option to store information in the cloud rather than on local storage

devices like servers or hard drives.

Multi-cloud

In a multi-cloud setup, many cloud computing services from various suppliers are integrated into a single physical location. Organizations may fulfill multiple business objectives, boost resilience, and prevent vendor lock-in by using the capabilities of several cloud platforms, such as (AWS), Microsoft Azure, and (GCP).

Multi-cloud Management

Managing and maximizing the usage of several cloud computing services from different providers within a single IT environment is known as multi-cloud management.

1.11 LIMITATIONS OF THE STUDY

The study on analyzing multi-cloud challenges by managing end-to-end operations for comprehensive multi-cloud management through different techniques encounters several limitations. Firstly, the scope of data collection may be constrained by the availability and uniformity of information across diverse cloud platforms, making it challenging to generalize findings. Additionally, the rapid pace of technological advancements in cloud computing can quickly render the study's results and recommendations outdated. Vendor-specific issues further complicate the analysis, as variations in features, interfaces, and APIs among different cloud providers can affect integration and interoperability. The complexity of multi-cloud environments adds another layer of difficulty, as the diverse configurations and services may limit the applicability of the study's findings. Security and compliance variability across industries and regions may also impact the relevance of the recommendations. Resource limitations, including budget constraints and access to advanced tools, may restrict the depth of the study.

Furthermore, differences in IT staff expertise and experience can influence the practical implementation of management techniques. Lastly, the dynamic nature of multi-cloud challenges means that emerging trends and future developments may not be fully addressed by the current research.

1.12 PROPOSED FRAMEWORK

Research design

In order to achieve this, a multi-cloud environment is constructed for research, using four distinct cloud service providers: AWS (Amazon Web Services), Microsoft Azure, Google Cloud Platform, and OpenStack (a private cloud). There are five distinct phases to the multi-cloud management framework, with each phase addressing a different aspect of the difficulties inherent in managing multiple clouds.

Research Approach

By utilizing a combination of machine learning and optimization techniques, this research aims to address several critical operational problems. These include optimizing resources, managing resources, scheduling workloads, and selecting the appropriate resources (virtual machines and the cloud) for deployment. Many scholars who have studied such issues in the past overlook how to choose the best cloud for installation in a multi-cloud setting. Additionally, while optimizing cloud resources, they do not account for information gathered from the current status of the resources for decision-making. Making accurate decisions in a short amount of time is crucial in a dynamic setting where the system's status changes rapidly. Therefore, having a high-speed processing architecture for faster calculations and decision-making is essential.

Research Method

We compare the proposed framework to comparable solutions on the market, such as RightScale and Clickr, and find that it is more robust and beneficial for the challenges that the industry currently faces. It tackles the issue methodically, measures it accurately, and provides solutions in a systematic manner.

Experiments performed

We conclude by outlining the performance of the proposed multi-cloud framework under various settings and demonstrating its efficacy in real-time scenarios through lab tests. Several lab simulations mimic real-world situations with multiple options.

The computational results obtained using the framework show how effectively this approach addresses the problems in the multi-cloud setting.

1.13 PLAN OF WORK

Chapter 1: Introduction

Chapter 1 introduces the study by outlining the importance and relevance of analyzing multi-cloud challenges through effective management techniques. It provides an overview of multi-cloud environments, the complexities involved in managing them, and the need for comprehensive strategies to address these challenges. The chapter sets the stage for understanding the objectives of the research, the scope of the study, and the significance of managing end-to-end operations in a multi-cloud landscape.

Chapter 2: Review of Literature

Chapter 2 offers a detailed review of existing literature on multi-cloud management and related topics. It examines previous research, theories, and methodologies related to resource management, security, cost optimization, and performance monitoring in multi-cloud environments. This chapter identifies gaps in the current body of knowledge and provides a foundation for the study by contextualizing the research within the existing scholarly work.

Chapter 3: Dynamic Resource Management Strategies for Multi-Cloud Environments

Chapter 3 explores various strategies for dynamic resource management in multi-cloud environments. It discusses techniques for optimizing resource allocation, scaling, and balancing workloads across multiple cloud platforms. This chapter highlights the importance of agility and flexibility in managing resources effectively and presents various approaches to address the challenges associated with dynamic resource management.

Chapter 4: Monitoring, Parameter Ranking, And Optimization for Predictive Knowledge Management

This chapter focuses on advanced techniques for monitoring and optimizing

multicloud environments. It discusses the importance of continuous monitoring for performance, security, and compliance, and introduces methods for ranking key operational parameters. The chapter also delves into optimization techniques that can be used to improve predictive knowledge management, allowing organizations to make data-driven decisions for better resource utilization.

Chapter 5: Experimental Results and Discussion

Chapter 5 provides a comprehensive analysis of the experimental results obtained from applying the proposed methods and techniques. It discusses the outcomes of various experiments, evaluates their effectiveness, and compares them with existing solutions. This chapter presents a thorough discussion on the findings, their implications for multi-cloud management, and any observed limitations or anomalies.

Chapter 6: Conclusion, Recommendations, and Future Scope

Chapter 6 summarizes the key findings of the study and their implications for managing multi-cloud environments. It offers recommendations based on the research results and suggests practical steps for organizations to enhance their multi-cloud management strategies. The chapter also outlines potential areas for future research, identifying opportunities to further explore and address ongoing challenges in multi-cloud management.

CHAPTER 2
REVIEW OF LITERATURE

Nagaraj, Bharath Kumar (2024) [1] Organizations looking for efficient, adaptable, and scalable solutions have a potential opportunity with the combination of artificial intelligence (AI) and multi-cloud architectures. Nevertheless, in order for this integration to be implemented successfully, a number of problems must be solved. In this article, we will look at the main obstacles to AI integration in multi-cloud settings and provide some solutions. To begin, a major obstacle to AI integration across different cloud platforms is compatibility. Improving data and service interchange is made more difficult by the fact that each cloud provider has its own set of APIs, data formats, and infrastructure settings. Adopting uniform data formats, APIs, and interoperability frameworks are essential standardization initiatives to solve this. Improving portability and easing interaction between AI components deployed across different cloud environments may be achieved by employing containerization technologies such as Docker and Kubernetes. Second, integrating AI across several clouds is fraught with difficulties related to data management and governance. To guarantee data security, integrity, and regulatory compliance in all cloud settings, strong governance structures are required in response to data sovereignty concerns, data privacy legislation, and compliance mandates. These risks may be reduced and confidence in multi-cloud AI installations can be fostered by the use of thorough data management measures, such as data encryption, access limits, and auditing procedures. Because artificial intelligence workloads deployed across several clouds run the risk of experiencing latency, network bottlenecks, and resource congestion, optimizing performance is an important consideration. To optimize performance and reduce operational costs across diverse cloud infrastructures, innovative orchestration methods like auto-scaling and workload scheduling algorithms allow dynamic resource allocation and load balancing. Making sure multi-cloud AI systems are resilient and fault-tolerant is another difficulty. Preventative actions are required to keep systems available and reliable in the face of unavoidable cloud outages, network interruptions, and hardware breakdowns. To make systems more resilient and less susceptible to failures, it is recommended to use data replication algorithms, disaster recovery procedures, and redundancy mechanisms across different regions of the cloud.

González-Rodríguez, Miguel et al., (2024) [2] A solid grasp of processor scheduling algorithms is essential for instructors of the operating systems course that is required for computer engineering degrees. Unfortunately, it has been discovered that our existing understanding of classical algorithms is inadequate for this particular scenario. In order to find new trends and algorithms that may improve student training and topic teaching, it is suggested to do a state-of-the-art review in the area. Consequently, research on the current state of the art is extensive, and study guides are created to help students understand the algorithms. Furthermore, a software simulator is created to evaluate several algorithms in a controlled setting, enabling the most promising ones to be validated for use in classroom instruction.

Fadhil, Heba. (2024) [3] It is critical to optimize system performance in diverse and dynamic contexts while efficiently managing computing operations. Consequently, this article takes a close look at methods for scheduling tasks and allocating resources. Using different task-to-node ratios, the paper assesses five algorithms: Genetic Algorithms (GA), Ant Colony Optimization (ACO), Firefly Algorithm (FA), and Simulated Annealing (SA). In order to measure how well an algorithm performs, the study focuses on two metrics: finish time and deadline. It then conducts an extensive analysis of how these algorithms behave under various workloads. The tests show that algorithmic behaviors vary with workload in distinct ways. Even though the ACO algorithm struggled with Task 5, the GA and PSO algorithms performed well in the 15-task and 5-node situation, finishing all tasks ahead of schedule. An expanded system that supports an adaptable algorithmic approach depending on workload factors is proposed in the research. With 10 tasks and 5 nodes, the GA and PSO algorithms achieved numerical success by finishing all of the tasks before the deadlines, however the ACO algorithm encountered problems with some of the tasks. According to the research, the aforementioned system provides a unified strategy for dealing with the chaotic issues of work scheduling and resource distribution. An aggressive and intelligent scheduling method is available, which runs asynchronously when there are more tasks submitted for completion and dynamically aborts when system usage and load cascade too much. If you're having trouble allocating resources or managing project timelines, the suggested design appears like a complete answer. It focuses on a strategy for selecting algorithms that are flexible and based on semantic properties. Empirical demonstration of each algorithm's performance under different

circumstances is provided by effects of obtaining measurable statistical findings from the experiments. Nowadays, it's very critical to strive for better performance and resource efficiency in computer settings. The complex dynamics between job scheduling, target allocation, and address demand on computer systems also emerge as an important component of system efficiency and effectiveness. Efficient task coordination and resource allocation are necessary to accomplish performance goals, reduce operational costs, and maximize energy use. Peak performance and resource optimization via strategic scheduling are the topics covered in this article, as is the use of met heuristic techniques in purpose schedules. Numerous activities demanding different processing commitments, interdependence condition durations, and preference levels are running simultaneously in today's computer settings, thanks to the favorable expansion of sophisticated applications. Even with sensible and reasonable use, the availability of computer resources is relatively dynamic, which hinders system functioning. Poor performance, increased operational costs, and an unpleasant user experience are possible outcomes of inefficient scheduling of various activities and resources. As a result of shifting workloads and resource constraints brought forth by distributed systems, these difficulties are exacerbated when cloud and edge computing are used.

Merseedi, Karwan & Zeebaree, Subhi. (2024) [4] Cloud computing has grown in popularity among both academics and businesses because to the idea of several clouds. Businesses are increasingly turning to cloud computing to fulfill their computational needs, which has led to a rise in the usage of multi-cloud approaches. Focusing on federated and hybrid cloud systems, this paper presents a comprehensive examination of cloud architectures designed for dispersed multi-cloud computing. Examining the pros and cons of using and managing several cloud resources from different providers is the focus of this research. The study begins by outlining the fundamentals of multi-cloud methods, highlighting the significance of adaptability, scalability, and resilience in modern computing environments. The research continues by delving into the details of hybrid cloud designs, particularly those that aim to merge public and private cloud resources in an effortless way. Major considerations such as data sovereignty, security, and workload orchestration are addressed within the framework of hybrid cloud deployments. Furthermore, the study explores federated cloud architectures, which provide light on how businesses may manage and

coordinate workloads across several cloud providers. Federal cloud computing is complicated; understanding it requires looking at how resources are identified, policies are enforced, and interoperability processes are handled. This study explores recent innovations in technology, best practices, and standards that facilitate the maturation of multi-cloud ecosystems. The report examines current research and industrial processes, highlighting areas that need improvement and suggesting potential paths for future advancement. In order to develop and implement distributed multi-cloud solutions with expertise, decision-makers, researchers, and practitioners will need a complete understanding of the shifting cloud architecture scenario. Finally, by drawing on a wide variety of resources, this essay gives a comprehensive review of federated and hybrid cloud architectures. The research aims to contribute to the ongoing discussion on optimizing and designing cloud environments in light of the dynamic nature of technology by conducting a thorough examination of the challenges and opportunities related to multi-cloud computing.

Kaur, Tanvir. (2024) [5] with the proliferation of cloud computing, multi-cloud networking has emerged as a critical component of contemporary network architecture. This article discusses the concept of multi-cloud networking and the advantages and disadvantages of implementing it. Among the many advantages of multi-cloud networking are reduced operating expenses, more vendor flexibility, improved availability and resilience, access to top services, and compliance with data sovereignty regulations. Architectural challenges, security and compliance concerns, administration, staff training and expertise, etc. are just a few of the complications of multi-cloud networking. To illustrate these points, the article cites several research, surveys, and real-world instances. Addressing these issues is critical for getting the most out of multi-cloud networking.

Voruganti, Kiran. (2024) [6] Optimizing the deployment and administration of cloud resources across multiple environments is the goal of this article, which investigates the vital role of multi-cloud orchestration in navigating the cloud computing world of today. Research Approach: This study examines the strategies and tools that enable effective multi-cloud orchestration by conducting a systematic review of academic articles, industry reports, and case studies. The Flexera 2021 State of the Cloud Report is one of the sources that were used to compile this study, insights from

Gartner, and academic contributions from jamshidi et al. and garg et al. Conclusions: Optimizing operational efficiency, resilience, and cost-effectiveness in cloud deployments is made possible by multi-cloud orchestration, according to the study. To address the complex needs of contemporary applications, it highlights the strategic advantages of coordinating a diverse range of cloud services, such as public, private, and hybrid clouds. For multi-cloud systems to run smoothly, securely, and compliantly, the research highlights the need of sophisticated orchestration tools. Specialized influence in the fields of thought, policy, and practice: By adhering to the guidelines provided in this article, businesses may successfully use multi-cloud orchestration to maximize their cloud investments and create a harmonious outcome.

Kotha, Rajesh. (2023) [7] Organizational ability to manage high-level plans is enhanced by multi-cloud solutions in information technology. Consequently, multi-cloud refers to the practice of using many cloud providers rather of just one. This allows enterprises to take use of the capabilities supplied by each provider.

Salehnia, Taybeh et al., (2023) [8] in recent times, IoT performance has been improved by using cloud and fog computing. Cloud computing's immense potential speeds up the analysis and storage of large data gathered from Internet of Things devices. To alleviate security concerns and unnecessary delays, new fog-based methods are emerging that may enhance service quality for IoT applications. To add to that, a more effective way of scheduling tasks lowers energy usage, which in turn lowers CO2 emissions from cloud and fog nodes. Here, the necessity for a reliable method of scheduling tasks that takes into account the best way to manage resources in the Internet of Things is becoming ever more apparent. When it comes to satisfying customer demands, the Internet of Things' (IoT) job scheduling based on fog-cloud computing is vital. System performance may be enhanced by optimal job scheduling. So, the Multi-Objective Moth-Flame Optimization (MOMFO) algorithm is used to schedule Internet of Things (IoT) job requests on resources in this research. By improving fog-cloud computing-based Internet of Things services, it shortens the time it takes to complete task requests, speeds up system throughput, and decreases energy usage. A lower proportion of CO_2 emissions is a direct result of lower energy usage. After that, the datasets are used to assess the suggested scheduling approach that addresses the job scheduling issue. To evaluate its efficacy, the suggested technique is

compared to Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Firefly Algorithm (FA), Salp Swarm Algorithms (SSA), and Harris Hawks Optimizer (HHO). Experimental results show that the suggested method has accelerated the system's performance rate while decreasing the time it takes for Internet of Things (IoT) jobs to finish and throughput time. This has decreased latency caused by task processing, energy consumption, and carbon dioxide emissions.

Senjab, Khaldoun et al., (2023) [9] Improving the performance of data center infrastructure is becoming more critical as cloud services proliferate. Data centers can keep up the efficiency and speed needed to provide high-quality cloud services with the support of HPC, cutting-edge networking technologies, and resource optimization tactics. One optimization technique that offers advantages including greater integration and interoperability, quicker deployment and scaling, better resource consumption, increased portability, and security is running programs in containers. These advantages may help businesses enhance their application deployment and administration, which in turn allows them to better react to the ever-changing demands of their company. Containerized application deployment, scaling, and administration may be automated using Kubernetes, a container orchestration system. One of its standout features is its scheduling mechanism, which allows users to plan the deployment and execution of containers across a cluster of computers. Using the available nodes in the cluster, this method finds the optimal location for the containers. Within the framework of Kubernetes, this article presents an exhaustive analysis of many scheduling techniques. We describe and classify them into four groups: general scheduling, scheduling based on multi-objective optimization, scheduling with an AI emphasis, and auto scaling enabled scheduling. We also point out where there are gaps and problems that need to be addressed by future studies.

N., Nivedhaa & Pub, Iaeme. (2023) [10] this paper delves into how contemporary IT infrastructures may improve operational efficiency and agility by integrating DevOps approaches with cloud management. Infrastructure as Code (IaC) tools, continuous integration and delivery (CI/CD) pipelines, configuration management solutions, monitoring systems, and other important DevOps technologies used for managing cloud infrastructure are covered in detail. We can see the benefits, drawbacks, and appropriateness of popular technologies like Terraform, AWS CloudFormation,

Jenkins, and Kubernetes by comparing and contrasting them. In addition to outlining recommended practices for implementing DevOps in the cloud, the article discusses the complexity, security, and integration concerns that businesses have when trying to implement the methodology. It goes on to look at how DevOps and cloud management are going to change in the future due to new trends and technologies including edge computing, multi-cloud strategies, server less computing, and AI-driven automation. Organizations may optimize their DevOps methods and use cloud technology to promote business success and preserve a competitive advantage. This paper attempts to provide insights into these improvements and their ramifications.

Alyas, Tahir et al., (2023) [11] Dynamic resource provisioning becomes more available with cloud computing. It is critical to monitor a working service and make adjustments when certain criteria are met. This study investigates a decentralized multi-cloud setup for QoS assurance and resource allocation, resource estimation, and resource modification in response to workload and resource parallelism. The diverse range of service and resource suppliers makes resource allocation a difficult and intricate task. A coordinated plan of action is required to provide a consistent level of service when several service and resource providers are involved. The goal of a well-planned and reasonable distribution of resources is to provide high-quality service. Finding the most important metrics to use in creating a system to distribute resources is also part of it. In this paper, we provide a framework for developing a decentralized multi-cloud resource allocation method according to the given criteria. Information availability, efficiency, and teamwork are the lynchpins of the suggested paradigm. In order to optimize resource allocation and long-term service quality, these three segments are further separated into subsets. The proposed framework has been validated using the CloudSim simulator. In order to achieve sustained QoS, several studies have been performed to determine the optimal setups for improving cooperation and allocating resources. The results validate the parameters and the proposed architecture for a decentralized multi-cloud system.

Somasundaram, Prakash & Pub, Iaeme. (2023) [12] in today's ever-changing cloud computing landscape, security concerns arise while managing and accessing resources across CSPs. With its capacity to safely traverse the complexities of multi-cloud systems, federated access control is the subject of this paper's exploration. It brings

attention to the security problems, including the need of governance and compliance, difficulties in managing identities and access, and worries about the privacy and security of data. Also included are practical implementation details and recommended practices for federated access control. The results of the study highlight the critical role that federated access control plays in protecting users' privacy and security while making full use of the capabilities offered by different CSPs.

Balasubramanian, Sivasubramanian et al., (2023) [13] A state-of-the-art cloud infrastructure that can effectively handle AI-intensive workloads is essential, as the demand for AI applications continues to skyrocket. With an emphasis on the difficulties associated with AI-intensive workloads, this article explores the complexities of cloud-based workload allocation and scheduling. The goal of our study is to find better ways to schedule and allocate AI workloads in the cloud by analyzing current tactics, finding their limits, and proposing new ones. We highlight scalability, changing workload characteristics, and resource heterogeneity as the core concerns facing AI-intensive task management in our explanation of the obstacles. Allocating resources efficiently is a challenge for AI workloads due to their variable computing needs, and adaptive techniques are needed to handle changing computational requirements over time. Furthermore, guaranteeing scalability is crucial for maintaining performance in cloud settings as AI models and datasets grow in complexity and quantity. In order to better understand the benefits and drawbacks of modern and more conventional approaches to task distribution, we have conducted a comprehensive literature analysis. While surveying the current state of affairs, we also get into scheduling methods used for AI-intensive work management. Our innovative approach to these problems is based on scheduling that is informed by machine learning, efficient task migration mechanisms, and dynamic resource provisioning. The goal of the framework is to use machine learning algorithms to forecast the features of AI workloads and to deploy resources in an adaptable manner according to the changing nature of these workloads.

Alonso, Juncal et al., (2023) [14] The ubiquitous adoption of the Internet of Things paradigm and the maturation of cloud computing as a utility service have both contributed to a meteoric rise in demand for data storage and processing power. Due to many flaws, the conventional usage of cloud services, which centered on

consuming from a single source, is no longer appropriate. One of the most crucial is the danger of vendor lock-in. The way applications are planned, built, deployed, and managed across such diverse ecosystems is being impacted by the shift from using a single cloud provider to combining many kinds of cloud services, which we are helping to bring about. In keeping with the idea of multi-cloud applications, the end product is a successful heterogeneity of approaches, methodologies, tools, and frameworks. This research has many objectives. The main goal is to describe the concept of multi-cloud from an application development perspective by defining multi-cloud native applications based on the existing literature. Afterwards, we lay the framework for architecturally describing these kinds of applications. Finally, some unanswered research questions have been uncovered by us based on the data. A systematic literature review (SLR) was used to achieve this goal in which a substantial body of primary research published between 2011 and 2021 was examined and categorized. Five key study themes have emerged from the in-depth examination, all aimed at bettering the DevOps lifecycle of "multi-cloud native applications" throughout development and operation. At the end of the study, the authors point the way toward research priorities and future projects that the software community should undertake.

Somasundaram, Prakash. (2022) [15] When businesses use multi-cloud solutions, staying in compliance with regulations across different platforms and countries becomes much more difficult. This article takes a look at the present state of compliance management software and discusses new developments in the field. The paper begins by presenting the challenges that businesses face when they use various cloud service providers. These providers have different security protocols, data policies, and regulatory frameworks that apply to different regions. As a result, businesses have a lot of work to do to ensure that their systems are compliant with each other and with industry and regional standards. The paper continues by discussing technological solutions, with a focus on automation and machine learning. These solutions streamline compliance processes, such as intelligent violation detection, automated compliance reporting, and continuous cloud infrastructure monitoring. Additionally, the paper explores how block chain technology can improve transparency and immutability of compliance data, strengthening the overall compliance posture. Adopting a holistic approach that aligns multi-cloud operations

with regulatory guidelines and optimizes compliance management through the utilization of innovative technologies is urged upon enterprises in the paper, which also highlights the significance of developing comprehensive compliance frameworks to integrate and harmonize the diverse requirements across multi-cloud environments.

Zheng, Tao et al., (2022) [16] The term "edge computing" refers to a novel approach of delivering cloud computing resources close to mobile consumers, at the network's periphery. When dealing with computation-intensive and delay-sensitive activities, it provides a viable option for mobile devices. Nevertheless, the network's periphery offers a constantly changing setting characterized by a multitude of devices, users who are highly mobile, applications that are diverse, and traffic that is both continuous and intermittent. Problems with task failure and system performance are common outcomes of uneven resource allocation in such a setting, which impacts edge computing. With the aim of achieving workload balance, decreasing service time, and lowering the failed task rate, we presented a deep reinforcement learning (DRL)-based workload scheduling method to address this issue. At the same time, we tackle the high-dimensional and difficult workload scheduling issue by using Deep-Q-Network (DQN) techniques. Our suggested method outperforms competing techniques in terms of service time, virtual machine (VM) use, and failed task rate, according to simulation data. We have developed a method based on DRL that can efficiently solve the edge computing workload scheduling issue.

Cheng, Wei et al., (2022) [17] There is a lack of control over both internal and external data in the IT infrastructure due to the presence of several master servers, which leads to inefficient data management. Additionally, it becomes very difficult to monitor the platform's information access security. Consequently, a hybrid cloud-based multi-cloud management platform is developed for the purpose of overseeing IT infrastructure. Using data from both the internal and external IT infrastructures, as well as two master servers, the cloud management platform's general architecture and cloud structure design are put into action simultaneously. Desensitized access to resources inside the IT infrastructure is also guaranteed by the framework. An IT infrastructure cloud management procedure and an adaptation method for cloud data load balancing are also created. Cloud management for information technology infrastructure and the extraction of association rules from groups of frequently used

IT infrastructure items are both accomplished via the usage of the fuzzy information feature extraction approach. The experimental findings demonstrate that after 400 repetitions, the cloud link utilization is around 90%, the cloud management latency is less than 1.1 ms, and the cloud management error is less than 2%. Additionally, compared to cutting-edge frameworks, IT infrastructure packet forwarding accuracy is superior.

Kaur, Ramanpreet et al., (2021) [18] Scheduling resources in multi-cloud setups is no easy task. A lot of people are excited to work on multi-cloud technology since it has the potential to solve a lot of issues, such stability, interoperability, and suppliers lock-in. Distributing resources on demand became a difficult task due to the unpredictability of multi-cloud setups with diverse user needs. Predicting optimal resource allocation management from current rules in multi-cloud settings remains a primary focus of research. A comprehensive literature review on resource management in multi-cloud settings is the overarching goal of this study. Because of their adaptability and dependability in current settings, optimization approaches have been seen as one of many open topics and potential future difficulties. Covering current homogeneous/heterogeneous user needs, cloud applications, and techniques to handle it in multi-clouds is important for literature study analysis. This study introduces a comprehensive taxonomy for cloud resource management, as well as a description and categorization of multi-cloud resource allocation approaches. Last but not least, we go into the unanswered questions and potential future paths of resource management in a multi-cloud setting.

Mezni, Haithem et al., (2021) [19] Virtualized resources (such as software, business processes, databases, platforms, mobile and social apps, etc.) may be easily hosted, managed, and made available as on-demand services via cloud computing, which has become more popular in recent years. New service models, such as big service, have emerged as a result of the massive amounts of data generated by this "servicelization" across many different sectors. The latter is what really helps to conceal the complexity of massive data via encapsulation, abstraction, and processing. Nevertheless, the necessary management facilities and tools are still missing from this intriguing approach. Big services and the data they access need constant management operations to keep them in a steady condition and ensure high-quality execution in the face of the

unpredictable and ever-changing nature of the cloud environments where they are hosted. Frameworks for planning, creating, launching, and overseeing large-scale services are crucial in this setting. This study aims to shed light on the new large service model by discussing it from the perspective of the lifecycle management stages. We also investigate how multi-cloud methods and big data frameworks contribute to the delivery of large services. At the conclusion of this article, you will get a synopsis of the research strategy for this subject.

Chatzithanasis, Georgios et al., (2021) [20] By using cloud computing resources from many providers, known as a hybrid cloud solution, significant benefits including improved risk management, operational efficiency, and freedom from vendor lock-in may be realized. Hybrid techniques and using many cloud providers are therefore relied upon by many organizations. Optimal performance/cost balance is achieved by multi-cloud adoption, which involves strategically distributing cloud management and resources across different providers. This allows for improved efficiency by taking advantage of economies of scale. Within this framework, the study examines 23 different IaaS providers' pricing practices and uses DEA methodology to assess the efficacy of various multi-provider service packages. The findings are promising since they show that multi-cloud solutions are becoming more efficient and that cost reductions are possible. For customers looking at hybrid solutions, which include several IaaS services from multiple providers, they provide a quantifiable driver.

El-Sharawy, Enas. (2021) [21] A computer's central processing unit (CPU) is its most vital component. One definition of CPU scheduling is the process that decides which processes will run in parallel with one another by determining which ones enter the CPU and which ones wait. The primary function of operating systems in achieving maximum CPU usage is scheduling algorithms for CPU management. The purpose of this article is to compare and contrast various CPU scheduling methods by reviewing the relevant research. It was found via our examination of the Round Robin, Shortest Job First, First Come First Served, and Priority algorithms that many researchers have put forward several methods to improve CPU optimization criteria such reaction time, turnaround time, and waiting time. However, no algorithm has proven to be superior in all of these areas.

Ebadifard, Fatemeh & Babamir, Seyed Morteza. (2021) [22] System stability, response speed, and resource productivity may all be improved by using the load balancing approach to distribute requests that dynamically enter the cloud environment. An issue with dynamic load balancing is that it raises communication overheads between virtual machines, which may be problematic when moving data between them. The problem of communication overheads is disregarded in the majority of load balancing approaches. Here, we try using Autonomous Load Balancing to fix this issue. The majority of the research on cloud computing task scheduling has been on requests that are tied to the central processing unit (CPU). Here, requests are categorized as either CPU-bound or I/O-bound according to the available resources. There is no way to use the existing load balancing techniques when both kinds of requests are considered. The suggested technique is tested using the CloudSim program, and its results are contrasted with those of Round Robin, Autonomous, Honey-Bee, and Naïve Bayesian Load Balancing methods. This proposed algorithm can evenly distribute the workload among the VMs and allocate requests to appropriate ones based on the resources they require. As a result, there is a decrease in communication overheads and an increase in load balancing degree. The results show this to be true for both the actual workloads of the NASA and Calgary servers and for the sample workload.

Arora, Neeraj & Banyal, Rohitash. (2021) [23] Thanks to innovations like on-demand processing, resource sharing, and pay-per-use, cloud computing is quickly becoming one of the most exciting new areas in computer science. Security, managing quality of service (QoS), data center energy use, and scalability are some of the cloud computing problems. One of the many difficult issues with cloud computing is scheduling, which entails allocating resources to various activities in order to maximize quality of service metrics. In cloud computing, scheduling is a famous NP-hard issue. A good scheduling method is going to be needed for this. An effective method for allocating user tasks to available cloud computing resources was suggested using a number of heuristics and meta-heuristic algorithms. The use of hybrid scheduling methods has grown in the cloud computing industry. This study provided a comprehensive overview of cloud computing scheduling algorithms, including those that are hybrids of two or more algorithms. Hybridizing an algorithm essentially means taking the best parts of both the original and the modified versions.

The paper goes on to categorize the hybrid algorithms, examine their goals, QoS parameters, and where the field of hybrid scheduling algorithms is headed in the future.

Bucur, Vlad & Miclea, Liviu-Cristian. (2021) [24] the foundation of information technology is the handling of data from several sources. Software initiatives depend greatly on the volumes and kinds of data that are absorbed by one or more networked systems. These systems might be as basic as apps or as sophisticated as self-driving automobiles. Not only is data consumed, but it is also changed or altered, necessitating a large computer power. Autonomous cars, one of the most interesting applications of cyber-physical systems, heavily on AI and deep learning to simulate the very complicated behaviors of a human driver. A lot of data is needed to try to map human behavior, which is a big and abstract idea. AIs utilize this data to learn more and become more adept at solving complicated issues. A comprehensive method for resource management in a multi-cloud context is described in this study. By using multi-cloud technology and commercially available solutions to scale resources and improve system resilience while being as cloud agnostic as possible, this API aims to meet the ever-increasing resource needs. In light of this, the work presented here will include an architectural analysis of the resource management API, a high-level overview of the implementation, and an experiment designed to demonstrate the viability and relevance of the systems discussed.

Ayyadapu, Anjan Kumar. (2021) [25] this study investigates the security issues that arise in multi-cloud setups and the potential applications of machine learning (ML). Multi-cloud systems provide enterprises scale and flexibility, but they also make data and application security more difficult. The difficulties with dispersed data, identity and access control, interoperability, and compliance are covered in the article. The use of multi-cloud security techniques to reduce these threats is then examined. Lastly, the study looks at how anomaly detection, intrusion prevention, and data encryption provided by machine learning may improve multi-cloud security.

Mohammad, Naseemuddin. (2021) [26] within cloud computing, multi-cloud settings provide special potential and problems. This abstract explores the crucial elements of improving security and privacy in Multi-Cloud environments by means of an extensive analysis that centers on access control and encryption methods. The goal of

the research is to address the rising significance of data security and privacy assurance across cloud providers. This study clarifies the challenges and solutions related to information security in Multi-Cloud systems by examining encryption techniques and access control schemes. This research offers important insights into strengthening Multi-Cloud settings against possible threats and weaknesses by a thorough investigation of security procedures and privacy protections, eventually leading to a more secure and resilient cloud computing ecosystem.

Heilig, Leonard et al., (2020) [27] These days, cloud users looking for the most affordable and compliant solutions have a difficulty due to the wide choice of cloud services that cloud providers provide, particularly when running highly scalable microservice architectures. An intelligent cloud brokerage mechanism may provide a solution by choosing cloud services from many clouds on behalf of customers, taking into account their unique needs and aspirations. We introduce the Cloud Service Purchasing Problem (CSPP) in this work, which seeks to combine unique consumer and application job needs with cost minimization. We provide two big neighborhood search strategies in addition to a mixed-integer programming paradigm to address this issue. We execute various computational experiments to assess the efficacy of the suggested methods using well-defined issue cases that include data from actual cloud providers. Their performance is competitive as they can solve every case in a short amount of computing time. In conclusion, the importance of this issue and decision support methodologies is examined via a comparison of use trends concerning several modern cloud providers and the virtual machine varieties available for diverse situations.

Priyadarshini, Aliva. (2020) [28] Among the many distributed computing technologies, cloud computing is among the most popular and in high demand. Cloud computing has evolved into several varieties, such as single, hybrid, and multi-cloud architectures. Because the cloud can process hundreds of requests per second, resource sharing, information loss reduction, server-side data storage elimination, and a host of other benefits, task scheduling will be a major topic in distributed architecture and all forms of cloud computing. In this section, we go over the architecture of several clouds and the scheduling strategies used to divide the workload equally across them. In the past, many algorithms have been proposed to

discover a near-optimal solution for task allocation. These include Cloud List Scheduling (CLS), Cloud Min Scheduling (CMMS), Minimum Completion Cloud (MCC), Median Max Algorithm (MEMAX), and Multi-objective scheduling (MOS).

Aldailamy, Ali et al., (2020) [29] in recent years, there has been an increased need for greater computing power. Grid computing came about as a result of the ever-increasing shortage of processing power to meet these demands. The growing need for resources, bandwidth availability, storage capacity, and processing power was met by this technology. Grid computing is regarded as a distributed system that makes use of resources from many computers spread out over different geographic locations. Workloads with large amounts of data and no interaction are often handled by this system. Researchers are now faced with the difficulty of figuring out the best job scheduling strategy that offers the best resource utilization in this very varied environment. Presenting an assessment of resource usage for certain heuristic scheduling methods in a Grid Computing Environment is the primary objective of this paper. Out of the three heuristic scheduling algorithms that were examined, the suffrage method yielded the best resource utilization, according to the findings of the two testing cases.

Chimakurthi, Venkata Naga Satya Surendra. (2020) [30] The acceptance of zero-trust security models and architectures has surged lately for a number of reasons, including the growing popularity of off-premises cloud technologies, the diversity of IT devices, and the growth of the (IoT). Constructed on the idea of various authentication and trust levels, this strategy assumes that users, devices, applications, and networks are all untrustworthy. The phrase "zero trust" refers to the transition from static network perimeters to safeguarding people, property, and resources as new cybersecurity paradigms emerge. Zero trust concepts may be used to the design of corporate and economic architecture and operations. According to the concept of zero trust, assets or user accounts are assumed to be implicitly devoid of confidence due to their physical placement on the network (local networks vs the Internet), as well as their ownership (personal or business). Before establishing a connection to an organizational resource, authentication and authorization must be completed. Cloud computing comes in a wide variety of forms, including public, private, hybrid, and on-premises. A multi-cloud deployment approach for data centers replaces reliance on a private cloud or on-

premises architecture with a wide range of public cloud service providers. Hybrid multi-clouding is a multi-cloud deployment that combines on-premises technology with both public and private clouds. This article addresses the challenges associated with implementing the zero-trust security paradigm for multi-cloud systems and applications.

Tomarchio, Orazio et al., (2020) [31] As more and more businesses and individuals enter the cloud computing industry, it is clear that this model is delivering on its claims. However, the persistent expansion of the cloud sector is creating difficult obstacles for those involved. One of the most difficult things for a service provider to do is to manage their resources well enough to maximize profits without sacrificing client happiness. Customers must make an efficient resource selection when deciding which of the many similar possibilities provided by different providers the best fit for their needs is. Several research projects suggest using software frameworks, more especially cloud resource orchestration frameworks (CROFs), to effectively manage the different resources offered by numerous cloud providers in accordance with the needs of the client in a complicated multi-cloud context. This research aims to provide a thorough analysis and comparison of the most important Cloud Resource Orchestration Frameworks (CROFs) that have been published in the literature. It also seeks to highlight the open issues in multi-cloud computing that the research community should focus on in the next years.

Kritikos, Kyriakos et al., (2019) [32] the cloud promises endless scalability and cost savings, along with improved flexibility in resource management for all types of applications. Thus, a recent trend towards cloud-based business process (BP) migration might be attributed to these benefits. Currently, only one Cloud may be moved at a time, and it must be done manually. Nonetheless, a multi- and cross-Cloud setup of a BP can be advantageous since it can leverage the best deals from several Clouds and prevent the lock-in effect by allowing the deployment of various BP instances in various Clouds near BP customers' locations. In this regard, this paper offers a unique environment design that embodies the multi-Cloud BP provisioning concept. This environment includes novel components that provide cross-level monitoring and adaption of business processes (BPs) and cross-level orchestration of

cloud services. Additionally, it makes use of a language known as CAMEL, which has been modified to enable adaptive provisioning of multi-Cloud BPs.

P, Dr.Suresh. (2019) [33] A method of delivering IT resources via the Internet, including networking, storage, databases, and computer servers, is called cloud computing. It provides the resources as services with speedier, more flexible, economies of scale, and a focus on demand. Because of the scalability of cloud users and services, the main issues in cloud computing are managing workloads and allocating resources. Despite the abundance of options for allocating resources and managing workload, the majority of them fall short in this regard. A reinforcement learning–based approach to improved resource allocation and workload management is proposed in this research to boost cloud performance under heavy user and activity loads (RL-ERAWM). It uses the Q-Learning technique, which takes the virtual machine's workload and request arrival rate into account. In terms of reaction time, makespan, and virtual machine usage, experimental findings demonstrate that the suggested solution improves task scheduling and workload management process performance when compared to other methods.

Di Martino, Beniamino et al., (2019) [34] In order to automate cloud performance optimization and anomaly detection, cloud customers throughout the globe are examining the next generation of (AI) based cloud management solutions. AI tools need support for machine learning optimization aimed at numerous goals and a consistent representation of cloud services in order to function well across clouds. We propose that both can be supported by ontology-based models.

Vijayashree, J & Jayashree, J & M., Dr. (2019) [35] A new method of delivering IT services that is enabled by net protocols may be cloud computing. Typically, it entails the supply of resources at the infrastructure, platform, and software system levels that are dynamically climbable and sometimes virtualized. It covers entirely new foundations, such as failover mechanisms, quality of service, virtualization, measurability, and ability. The very characteristics that set the cloud atmosphere apart from more traditional environments are that it is highly climbable, encapsulated as an abstract entity that provides customers outside the cloud with entirely different levels of services, driven by economies of scale, dynamically organized (through virtualization or other methods), and delivered on demand. Cloud environments come

in three varieties: public, non-public, and hybrid. A cloud that provides cloud resources and services to the wider public may be referred to as a public cloud (also known as an external cloud). A personal cloud, also known as an internal cloud or an enterprise cloud, is a rented or in-house cloud. Generally speaking, a hybrid cloud is made up of two or more different model clouds. However, we characterize a hybrid cloud as consisting of one public cloud and one private cloud. In such a cloud, an organization maintains its own private cloud that supplies and oversees certain internal resources and only pulls in external resources from the public cloud when needed.

Celesti, Antonio et al., (2019) [36] these days, the ability to store and retrieve data via the cloud is essential to the survival and expansion of both individuals and companies. The ability to store enormous volumes of data and files on distant, third-party cloud storage providers is becoming a more practical option in this context. Regretfully, there is no assurance as to the dependability of these suppliers or the availability of data. In this scenario, all the Cloud has to do is trust the other Cloud if it want to utilize its storage as a service. Using a Multi-Cloud Storage (MCS) system is one potential fix. It is far from simple, nevertheless, how businesses should design their own MCS system based on geo-distribution. In this work, we especially address how to test and validate an MCS system made up of three major cloud storage providers: Dropbox, Google Drive, and Copy, in order to optimize the system as a whole with regard to data retrieval and storage. Research has shown that the selection of cloud storage providers for file storage is contingent upon the data transmission performance based on file chunk size. Furthermore, we showed that the provider with the highest upload speed is not always the one with the best download performance.

Cao, Ronghui et al., (2019) [37] these days, cloud-supported Internet of Things, or cloud-IoT, is widely used in smart medical systems. Cloud computing architectures are used to overcome the limits of IoT-related medical devices with regard to data access, storage, scalability, and processing. However, it will be very challenging to construct or administer such an expanding and enormous medical IoT system in typical single-cloud platforms due to the fast development of medical equipment and the growing number of medical devices. In this study, we develop and deploy Tri-Storage Failure Recovery System (Tri-SFRS), an Open Stack-based platform for

medical IoT built on a multi-cloud architecture. We use a combination of methods to accomplish this effort reduction in Tri-SFRS implementation: a low-overhead native testing framework, a multi-cloud cascading architecture, a medical data storage-backup mechanism, and snapshot-volume cascaded operations for b-ultrasonic data. Tri-SFRS may also allow resource management specialization at the same time. Tri-SFRS is a native component of the Open Stack platform that showcases our suggested cascade framework's level of native Open Stack multi-cloud platform administration. By up to 20%, Tri-SFRS may lower the resource-request processing latency from B ultrasonic machines as compared to the conventional single-cloud Open Stack architecture. Additionally, our tests show how broadly applicable Tri-SFRS is.

Mandati, Sasikanth. (2019) [38] the research outlines the multi-cloud strategy's architecture for the computer and IT industries. The paper outlines the methods used in the cloud approach, as well as its applications and significance in the expanding global community. The report also outlines the components and methods of the multi-cloud approach as well as its potential uses in the future.

Markus, Andras & Dombi, József Dániel. (2019) [39] Cloud computing's assistance for handling sensor data is one of its main priorities, and the Internet of Things (IoT) concept is strongly related to cloud technology. Because IoT-Cloud systems are often used to control sensors and other smart devices that are linked to the cloud, a lot of data is created by these devices and has to be processed and stored effectively. One benefit of using simulation platforms is that they make it possible to study intricate systems without having to buy and set up actual resources. In our earlier work, we modeled IoT-Cloud systems using the DISSECT-CF simulator and included provider pricing models to allow cost-conscious experimentation strategies. The purpose of this study is to enable multi-cloud resource management, thereby expanding the tool's simulation capabilities even further. In order to minimize application execution time and usage costs, we provide four cloud selection methodologies in this study. We describe in detail our suggested approach to multi-cloud extension and assess the tactics provided by using situations from a weather application.

Chaudhary, Rajat et al., (2018) [40] the number of people using smart devices is growing exponentially, and this is resulting in a rise in the volume of data being generated from different smart devices. This data is different for each of the five key

variables that are used to classify it as big data. The majority of service providers, such as Google, Amazon, Microsoft, and others, have set up several geographically dispersed data centers to handle the massive volume of data produced by different smart devices, enabling consumers to get fast response times. These service providers analyze big datasets in this way by using Hadoop and SPARK extensively. The network through which data travels, or the underlying infrastructure, is one of the most crucial elements for the effective execution of any intended solution in this context, but it has received less attention. In the worst situation, performance deterioration may occur from the underlying network infrastructure's inability to transport data packets from source to destination owing to high network traffic related to data migrations between several data centers. With an emphasis on all of these problems, we provide in this paper a unique SDN-based big data management strategy that optimizes the use of network resources including bandwidth and data storage units. We examine several elements at the data and control planes that might improve the big data analytics optimization across numerous cloud data centers. We examine the effectiveness of the suggested approach, for instance, by using Bloom-filter-based insertion and deletion of an entry in the flow table kept at the Open Flow controller, which uses the rule-and-action-based method to classify network traffic most of the time. Developers may implement and examine real-time traffic behavior for upcoming big data applications in MCE by using the suggested approach.

Ferry, Nicolas et al., (2018) [41] Although there are an increasing number of cloud options available, developing and managing large-scale, distributed cloud applications continues to be challenging. The lack of interoperability between the existing cloud systems is one major barrier. Due to this, creating and maintaining complex apps that can be used on several cloud platforms and infrastructures becomes more difficult. This article shows how the Cloud Modelling Framework manages the complexity by providing: (i) a specialized language for defining the provisioning and deployment of multi-cloud applications, and (ii) a runtime environment for their continuous provisioning, deployment, and adjustment. The Framework leverages model-driven engineering and embraces the DevOps concepts.

Guo, Mian et al., (2018) [42] The delay-optimal virtual machine (VM) scheduling issue in cloud computing systems, where the resource pool's capacity for CPU,

memory, and storage is fixed, is the subject of this paper's investigation. The cloud computing system offers consumers virtual machines (VMs) as services. Over time, cloud customers may request different kinds of virtual machines (VMs) at random, with widely differing periods for VM hosting. First, we use a queueing approach to account for the dynamic and diverse workloads. Then, we construct the scheduling of virtual machines (VMs) in such a queueing cloud computing system as a decision-making process, where the optimization goal is the delay performance in terms of average task completion time, and the decision variable is the vector of VM configurations. To get the answers, a low-complexity online technique called SJF-MMBF—a combination of the min-min best fit (MMBF) scheduling algorithm and shortest-job-first (SJF) buffering algorithm—is suggested. To prevent the possibility of work starvation in SJF-MMBF, a different system called SJF-RL is further suggested. It combines the SJF buffering and RL-based scheduling algorithms. The simulation findings demonstrate that by providing a low delay at different work arrival rates for different shapes of job length distributions, SJF-RL is able to accomplish its aim of delay-optimal scheduling of virtual machines. The simulation findings further show that SJF-MMBF is efficient in throughput performance in terms of the average job hosting rate provisioning, even if it is sub-delay-optimal in a heavily loaded and highly dynamic environment.

Jambunathan, Baskaran & Kalpana, Dr. (2018) [43] the burgeoning field of cloud automation and IT infrastructure management is called DevOps. Businesses are investing time and resources in infrastructure management through continuous improvement and optimization. They are also looking for ways to increase productivity and efficiency, cut down on redundancy, and optimize costs associated with developing and implementing enterprise applications. More significantly, organizations need to have a way to better collaborate with different teams inside the business and continually analyze and optimize operational tasks since they have many environments to manage. Furthermore, a lot of businesses are moving to the cloud, and some of them have numerous cloud environments spread across different locations. This makes managing cloud migrations and devOps even more challenging, necessitating the need for a system to manage the main risks and issues. In this post, we'll talk about the main issues and how to solve them in a multi-cloud context. We'll also provide an integrated solution that tackles every issue and effectively supports

devops in multi-cloud environments by using the appropriate tools and framework architecture.

Babu, A & Latha, M. (2018) [44] One of the top emerging technologies in the distributed computing space today is cloud computing, which enables payments based on models that meet client demands and specifications. A collection of virtual computers that include every computational and persistent capability is included in the cloud. Enabling effective data access to distant locations over a network is the core tenet of cloud computing. Cloud is dealing with a lot of difficult issues every day. The main one that it faces is scheduling. Scheduling is the technique that allows a job to be completed on a computer device in a certain sequence. The term "scheduling" refers to a collection of guidelines for managing the task sequence for executing a known operation via a network using computers. An excellent scheduler modifies the scheduling approach to transform job execution into a variety of distinct jobs. In the study article we submitted, the researchers' new focus is on data management across several virtual machines in cloud computing. In order to solve some challenges, scheduling algorithms for various jobs are crucial. The goal of scheduling is to complete various activities accurately and in a shorter amount of time. We also spoke about the several components that cloud computing relies on, such as time, load balancing, and CPU utilization.

Alexander, Kena et al., (2017) [45] Many strategies have been used to address the issue of orchestrating application components across diverse cloud providers; as a result, cloud orchestration and management standards like TOSCA and CAMP were developed. An end-to-end solution that can define, deploy, and manage apps and their components across diverse cloud providers can only be provided via standardization. In contrast, TOSCA and CAMP serve distinct purposes in relation to cloud applications. Whereas CAMP is largely concerned with application deployment and administration, TOSCA is primarily concerned with topology modeling and orchestration. In addition to combining the two new standards, TOSCA and CAMP, this article offers a unique method that adds extensions to CAMP that enable declarative policy-based multi-cloud application orchestration. Additionally, the CAMP platform is extended, bringing the standards closer together to facilitate a smooth integration. A cloud application may span and be delivered across many

clouds thanks to our proposal's end-to-end cloud orchestration solution, which facilitates cloud application modeling and deployment procedures. Our validation research shows that our strategy is both feasible and beneficial.

Mezni, Haithem & Mokhtar, Sellami. (2017) [46] Utilizing cloud environments to provide a variety of resources as services has grown rapidly in recent years. Reusing and composing services is seen as a potent way to increase software development efficiency. Effectively assembling services from several clouds is still a problem, however. In fact, current solutions presuppose that all of the services included in a composition originate from a single cloud. This strategy is impractical since better services may be hosted by other clouds that are currently in use. The user request must be compared to the services in the multi-cloud environment (MCE) or, at the very least, clouds within the user's availability zone, in order to provide high-quality service compositions. In this research, we offer a Formal Concept Analysis (FCA) based multi-cloud service composition (MCSC) technique. To represent and aggregate data from several clouds, we use FCA. The foundation of FCA is the lattice notion, a potent tool for categorizing cloud and service data. Initially, we represent the multi-cloud setting as a collection of formal contexts. Next, we take formal notions and extract and merge candidate clouds. Ultimately, the best mix of clouds is chosen, and the MCSC is converted into a traditional service composition issue. Experiments conducted validated the efficacy and capability of the FCA-based approach in regrouping and identifying cloud combinations with a low communication cost and minimum number of clouds. Additionally, the top-down technique that was taken allowed for the quick selection of services hosted on the nearest and similar clouds, which immediately lowered the inter-cloud communication cost when compared to current ways. This was shown by a comparison with two well-known combinatorial optimization approaches.

Lahmar, Fatma & Mezni, Haithem. (2017) [47] Over the last ten years, cloud computing has emerged as the go-to option for hosting and delivering different types of computer resources as on-demand services. Cloud services may be merged to better meet user needs while taking into account the limitations of the virtualized environment, such as security guidelines, resource availability, and interoperability. Numerous studies have been undertaken in order to investigate the main concerns

around the subject of cloud service composition. Very few studies, nevertheless, have looked at these problems in a multi-cloud environment. In order to close this gap, we provide a thorough literature assessment on multi-cloud service composition in this work. First, some background on service composition in isolated clouds is provided. Next, we introduce the multi-cloud taxonomy and examine the approach used by researchers to address service composition in multi-cloud settings. In conclusion, we delineate the obstacles and prerequisites associated with the creation of multi-cloud services, along with the forthcoming paths. The necessity for efficient cloud service reuse techniques in multi-cloud settings has increased due to the spread of multi-cloud setups and the growing complexity of needs for cloud users. Facilities for multi-cloud service composition are absent from current solutions. We conduct a thorough literature study on multi-cloud service composition in this research. The multi-cloud taxonomy is presented. Next, we provide an overview and a classification of current multi-cloud service composition strategies. In conclusion, we delineate the fundamental prerequisites and the forthcoming paths.

Chimakurthi, Venkata Naga Satya Surendra. (2017) [48] in the last several years, new programming applications have been planned using a changed programming engineering methodology. Particularly suitable use cases for this engineering methodology come from an autonomously deployable programming administration in the avionics industry. These services are freely deployable using fully computerized equipment and are designed around business and mission capacity. Some types of uses become easier to build and maintain using micro services when they are divided into smaller, cohesive components that work together. This is not a traditional, "solid" application that has developed all at once. A few aspects of micro services-based engineering will be covered in this article, along with some possible applications for the aviation industry. The attributes of microservice-based engineering, such as endpoints, associations, componentization, and informatics. The expert application of micro services via containerization audits, administrative communication, and other design components. Specific open-source projects and elements that are helpful in building the architecture based on microservices. A sample collection of use scenarios.

Mimura Gonzalez, Nelson et al., (2017) [49] Utility computing, which is the providing of computer and storage resources as a metered service, is where cloud computing originated. Multitenancy is another important aspect of cloud computing that allows for cost and resource sharing across a large number of users. Features like flexibility and multitenancy are ideal for the demands of contemporary scientific research and data-intensive projects. Simultaneously, scalable and automated data processing and administration are necessary since research depends on the study of exceedingly big data sets. As a means of formalizing and organizing data analysis, workflows have grown in popularity as a model for managing intricate scientific procedures. Cloud computing and scientific procedures are made feasible in large part by effective resource management. To be a true high-performance infrastructure for scientific computing, clouds must overcome a number of obstacles, including data-intensive loads, complex infrastructures like hybrid and multi-cloud environments to support large-scale workflow execution, performance fluctuations, and reliability. This article is a survey on cloud resource management that offers a thorough analysis of the industry. A taxonomy is suggested for the examination of the chosen works, and the analysis in the end results in the identification of future issues and gaps that need research and development.

Heilig, Leonard et al., (2016) [50] Cloud computing is fast becoming a commonplace technology delivery strategy that businesses and researchers hope to benefit from. There is a great deal of opportunity to optimize the choice of cloud services in order to better meet the needs of users, or consumers, and applications, given that different cloud providers provide cloud services in different formats. With the introduction of multi-cloud environments recently, apps may now be run utilizing resources from many cloud providers in addition to those from a single provider. Owing to the increasing intricacy of cloud markets, clients may get decision help via the employment of a cloud brokerage mechanism that interacts with several cloud providers on their behalf. This work tackles a current optimization topic called Cloud Resource Management topic in multi-cloud contexts, which aims to minimize the financial cost and execution time of consumer applications that use Infrastructure as a Service (IaaS) from several cloud providers. We provide an effective Biased Random-Key Genetic Algorithm in response to customer demands for high-quality, real-time solutions to automate cloud resource management and related deployment procedures

at a reasonable cost. The results of our computational tests on a large benchmark suite created using actual cloud market resources show that our methodology performs better than the methods suggested in previous research.

Kritikos, Kyriakos et al., (2016) [51] Adaptive application provisioning across multiple clouds helps address the issue of vendor lock-in and optimize user needs by choosing the most suitable option from the plethora of services provided by many cloud providers. In light of this, new and old research platforms and prototypes are increasingly supporting this kind of provisioning. One big worry, which is really keeping consumers from migrating to the cloud, is security, which becomes trickier in multi-cloud environments. This issue is divided into two primary areas: (a) appropriate access control over user data, virtual machines, and platform services; and (b) scheduling and modifying application deployments in accordance with security specifications. In order to address these security concerns, this paper proposes a novel model-driven architecture and approach that secures multi-cloud platforms, gives users access to their own private space, and ensures that application deployments are both built to and capable of maintaining the level of security that the user has specified. Modern security software, secure model management techniques, and security standards are all exploited by such a system. It also discusses several access control situations that include programmatic, web-based, and external user authentication.

Moura, Jose & Hutchison, David. (2016) [52] Through the Internet, cloud computing provides virtualized networking, storage, and computing resources to both individual users and businesses in a fully dynamic manner. Compared to sets of local, physical resources, these cloud resources are more elastic, less expensive, and simpler to administer. Customers are encouraged to use the cloud for their apps and services as a result. Cross-platform and cross-technology functional issues arise when data and applications are moved into a shared environment outside of customers' administrative domains. This paper offers a modern analysis, mostly from a network viewpoint, of the most pertinent functional issues related to the present development of cloud computing. Additionally, a succinct explanation of cloud computing technology and ideas is provided in the article. It begins with a synopsis of the origins of cloud computing. Then, cloud service architecture models are explained, and a

thorough literature study is included along with a short discussion of the most relevant cloud computing products. The most important and useful network issues—security included—that are relevant to the delivery of high-assurance cloud services over the Internet are outlined and examined in this article. Lastly, future research directions and trends are also discussed.

Ferrer, Ana et al., (2016) [53] The cloud computing segment that has had the most increase in terms of acceptance and market share over the last several years is Platform-as-a-Service (PaaS). Simultaneously, cloud models are shifting toward hybrid clouds by taking into account the proliferation of services across various cloud providers. Most multi-cloud strategies to date have just looked at the IaaS layer, ignoring the remainder of the cloud stack. Due to PaaS multi-cloud's flexibility, this prevents businesses from taking use of the advantages of an automated, flexible, and agile infrastructure. The findings of two research projects—ASECETiC and SeaClouds—that address PaaS multi-cloud architectures are presented and contrasted in this article.

Gill, Sukhpal Singh & Chana, Inderveer. (2016) [54] the task of allocating the right resources to cloud workloads is difficult, and it is influenced by the quality of service (QoS) demands of cloud applications. The cloud environment presents resource allocation challenges due to heterogeneity, unpredictability, and dispersion that cannot be resolved by the current resource allocation strategies. From the body of current literature on resource scheduling algorithms, researchers continue to struggle to choose the most effective and suitable method for a given workload. This study presents a thorough, comprehensive literature review of cloud resource scheduling in particular as well as resource management in general. Using a comprehensive selection of 110 research articles from a larger collection of 1206 research papers published in 19 prestigious workshops, symposiums, conferences, and journals, a conventional, systematic literature analysis strategy is used in this study. The state of resource scheduling in cloud computing nowadays is broken down into many groups. A methodical examination of resource scheduling in cloud computing is provided, along with descriptions of resource distribution rules, kinds, types of resources, methods and administration for scheduling resources, and advantages. Also included is the literature on the thirteen different kinds of resource scheduling methods. Eight

different categories of resource allocation policies are also explained. Researchers will be able to identify key features of resource scheduling algorithms and determine which method is best for scheduling a given workload by carefully analyzing this research project. This study effort also includes recommendations for future research areas.

Faniyi, Funmilade & Bahsoon, Rami. (2015) [55] the ability to dynamically acquire, grow, and release computer resources in response to variations in workload is made feasible by cloud computing. A growing number of academic and commercial stakeholders are investigating cloud deployment alternatives for their vital applications. One unresolved issue is that the cloud ecosystem's service level agreements (SLAs) haven't developed to the point where crucial apps can be dependably installed in clouds. The goal of this article is to understand the state of the art and identify outstanding challenges by conducting a comprehensive study of the field of SLA-based cloud research. The resource allocation phase of the SLA life cycle is the focus of the study, however it also highlights implications for other stages. The findings show that: (i) the majority of studies account for the minimum number of SLA parameters; (ii) the most widely used methods for allocating resources are heuristics, policies, and optimization; and (iii) the monitor-analysis-plan-execute (MAPE) architecture style is most common in autonomic cloud systems. The findings support the principles of cloud service level agreements (SLAs) and their autonomous management, spurring more study and the development of practical, industry-focused solutions.

Debnath, Jayanta & Serpen, Gursel. (2015) [56] the viability of developing a real-time scheduling system for a multi-story, fully automated parking complex with many elevators is examined in this work using simulation. A single vehicle (car, small truck, SUV, or minivan) may be transported between floors using each elevator. Using the waiting line model of the queuing theory, the elevator count for a certain parking structure with 400 parking spots on each level and a number of floors between 4 and 20 is calculated under the assumption of a mean service rate and a customer arrival rate. Through the simulation research, a scheduling method based on layered partitions and genetic algorithms is assessed. During morning rush hour, the simulation environment simulates the mean customer arrival time and elevator

dynamics for crowded multi-story metropolitan commercial parking facilities. MATLAB was used to develop the elevator scheduling system's performance assessment. The maximum customer waiting time, mean elevator service times, and performance indicators were tracked. The results of the simulation show that the suggested design allows for reasonable customer wait and service times and high elevator usage rates.

Munteanu, Victor et al., (2014) [57] currently, existing technologies and services provide an underlying complexity that is hidden behind cloud service abstractions in an attempt to make it easier to implement Cloud Federations and Marketplaces. Specifically, resource management systems that interact with many cloud providers must provide wrappers for the Cloud service APIs and present a consistent interface for different services. In this article, we explore the approach used by a newly created open-source and vendor-neutral platform-as-a-service to facilitate the deployment of Multi-Cloud applications. A multi-agent system for automated Cloud resource management is part of the middleware. The solution's modular architecture allows for flexibility in accommodating new resource kinds and Cloud service options. The modules that allow for resource abstraction and automated management are the main topic of this study.

Jofre, Jordi et al., (2014) [58] the speed, latency, packet loss, and jitter of a network have a big impact on how well users perceive cloud apps. The performance of cloud applications is impacted by network services, and this influence is amplified in circumstances where the cloud infrastructure spans many administrative domains, as in federated or hybrid clouds. Supporting sophisticated managed networking features across dispersed cloud infrastructures is very valuable, especially considering the significant correlation between network and cloud application performance. Future production clouds and the Future Internet (FI) experimental community would both benefit from these features. This article presents an architecture and a set of protocols for integrating cutting-edge features for regulated networking into a multi-cloud environment. Through this connection, experiments operating in a multi-cloud test-bed may be supported with the deployment of customized network functions and services. Enhancing the performance of the cloud apps used by the experimenters requires the ability to manage network connection. We concentrate on the specifics of

integrating the BonFIRE multi-cloud architecture with three cutting-edge networking facilities. These three networking resources are: OFELIA, which leverages Open Flow to provide Software Defined Network capabilities; GÉANT's Bandwidth-on-Demand service; and FEDERICA, which facilitates regulated routing. While the connections with FEDERICA and GÉANT are now operational, OFELIA is planned as a future project for the connecting of a third facility.

Grozev, Nikolay & Buyya, Rajkumar. (2014) [59] because cloud data centers provide several advantages over private in-house infrastructure, they are increasingly being used as the deployment environment of choice for a variety of commercial applications. However, there are a number of drawbacks to the conventional method of using a single cloud, including availability, preventing vendor lock-in, and offering globally accessible services that are compatible with laws and have an acceptable Quality of Experience (QoE). Using many clouds, or a multi-cloud, is one option for cloud customers to lessen these problems. In this paper, we provide a method for distributing three-tier apps over many clouds to meet their primary nonfunctional needs. We provide resource provisioning and load distribution algorithms that are adaptable, dynamic, and reactive. These algorithms heuristically minimize overall cost and response delays while adhering to crucial legal and regulatory constraints. With a reasonable trade-off between cost and latency, our approach outperforms the state of the art in terms of availability, regulatory compliance, and quality of experience, as shown by our simulation using actual workload, network, and cloud characteristics.

Agarwal, Amit & Jain, Saloni. (2014) [60] in distributed computing, cloud computing is a new technology that allows pay per model based on user requirements and demand.A cloud is made up of many virtual machines that have storage and processing power. The main goal of cloud computing is to provide effective access to resources that are dispersed geographically and remotely. Scheduling is one of the numerous difficulties that cloud computing is facing as it develops daily. A set of rules called scheduling governs the sequence in which a computer system completes tasks. A skilled scheduler adjusts their approach to the work at hand and the ever-changing environment. A Generalized Priority method for task execution efficiency and comparison with FCFS and Round Robin Scheduling was described in this

research study. The technique should be evaluated using the cloud Sim toolkit; the results indicate that it performs better than other conventional scheduling algorithms.

Škrinárová, Jarmila. (2014) [61] the fundamental idea of elastic clusters as a hybrid approach to high-performance computing jobs for cloud and grid computing is examined in this study. The context of managing activities and resources in the elastic cluster is the main focus of the investigation. This study describes the scheduling algorithm's design, concept, and implementation. The scheduling approach is based on hill climbing (HC) and particle swarm optimization (PSO), and it is a suitable mix of the best aspects of both techniques. The Torque resource management integrates the algorithm on an HPC cluster. The algorithm's measurement and assessment approach is presented. The techniques of algorithm behavior verification for various task requirements—which are common for grid or elastic clusters—are presented in this work. We assess how well the suggested algorithm fits in comparison to existing solutions. Based on the analyzed data, it is verified that the suggested method more closely matches certain elastic cluster requirements.

Singhal, Manish et al., (2013) [62] Without pre-established cooperation agreements or defined interfaces, a suggested proxy-based multi-cloud computing architecture addresses trust, policy, and privacy concerns by enabling dynamic, on-the-fly collaborations and resource sharing across cloud-based services.

AlZain, Mohammed et al., (2013) [63] A large number of companies have begun to rely heavily on cloud computing. The cheap cost and easy access to data of cloud computing are only two of its numerous advantages. A key component of cloud computing is security assurance since users often save sensitive data with cloud storage providers, some of whom may not be secure. Consumers are concerned about assaults by malevolent insiders and outsiders on the availability and integrity of their data stored in the cloud, on top of the possible unintended consequences for cloud services. Despite the gravity of these issues, cloud computing security research has a lot of room to grow. Due to the risks of service availability failure and the possibility of malicious insiders in the single cloud, customers are predicted to become less interested in dealing with providers who use a "single cloud" model. There has been a significant surge in the trend for "multi-clouds," also known as "interclouds" or "cloud-of-clouds." This study aims to review current research on security in single

and multi-cloud environments and provide potential remedies. It is discovered that the usage of single clouds has drawn more attention from the research community than the use of multi-cloud providers in maintaining security. The goal of this effort is to encourage the use of several clouds since they may lower security concerns that impact cloud computing users.

Innocent, A. Anasuya. (2012) [64] The user may dynamically access cloud services over the Internet at any time and from any location thanks to the new computer age known as cloud computing. The cloud is made up of data and resources, and its services include software, infrastructure, applications, and storage that are delivered over the Internet in response to customer demand. To put it simply, cloud computing is a commercial and economic paradigm that enables customers to virtually access high-end processing and storage with little to no infrastructure required on their part. Three service models are available for cloud computing: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The administration of cloud infrastructure services is covered in detail in this article.

García-Gómez, Sergio et al., (2012) [65] the goal of the 4CaaSt project is to provide a PaaS framework that will allow for flexible Cloud-based service and application creation, marketing, deployment, and administration. The main obstacles that 4CaaSt has overcome to enable the thorough administration of services and applications in a PaaS are discussed in this article. These issues include developing a blueprint language to define cloud applications and their lifecycle management, providing a one-stop shop for cloud services, and managing resources at the PaaS level with sophisticated Network as a Service capabilities and flexibility. A portfolio of ready-to-use cloud native services and cloud-enabled immigration technology is also offered by 4CaaSt. There is also a description of the assessment procedure used to gauge 4CaaSt progress.

RESEARCH GAP

Managing end-to-end operations in multicloud environments presents a complex set of challenges that are yet to be fully addressed by existing research. While multicloud adoption is growing due to its benefits such as increased flexibility, redundancy, and the avoidance of vendor lock-in, comprehensive management techniques are still in

their nascent stages, particularly regarding interoperability, security, data management, performance optimization, and cost control.

One of the primary gaps in current research is the lack of robust solutions for interoperability and integration across diverse cloud platforms. Existing studies often focus on technical aspects of integrating different cloud services using APIs and middleware. However, there is a need for more sophisticated frameworks that ensure seamless interoperability without degrading performance or security. These frameworks should enable different cloud services to communicate efficiently and consistently, while also simplifying the integration process. Future research could explore the development of standardized protocols and advanced tools to facilitate better interoperability among heterogeneous cloud environments.

CHAPTER 3
DYNAMIC RESOURCE MANAGEMENT STRATEGIES FOR MULTI-CLOUD ENVIRONMENTS

3.1 MULTICLOUD MANAGEMENT STRATEGY

Today's information technology sectors are placing a greater emphasis on the development of applications that are more distributed, scalable, secure, and comprised of highly decoupled components, and that are deployed across many locations all over the world. The ability of applications to be highly available in nature, portable between platforms, capable of communicating across heterogeneous systems, and more responsive in nature is one of the most important requirements for businesses. In addition, businesses are placing a greater emphasis on developing creative solutions, increasing the amount of automation they implement, and continuously optimizing their environments in order to cut down on operational overhead and increase cost savings.

Given the increased level of rivalry among businesses in terms of accomplishing their business goals, it is of the utmost importance for these businesses to fulfil the demand and continuously enhance the user experience and the level of happiness experienced by their customers in every facet. The evolution of technology that enhance the nimbleness of the corporation is bolstering these commercial goals, increasing return on investment (ROI), and reducing the amount of time it takes to bring a product to market.

To address multi-cloud distributed scenarios, The development of new technologies is an ongoing process. These technologies include Docker containers, Micro services, container orchestration tools, DevOps, automations, and many others. The goal of these technologies is to make this application more distributed in nature.

Furthermore, in order for industries to achieve their primary business goals, such as lowering costs and enhancing operational excellence, it is vital for them to optimize the distributed environment in which they operate. In the course of this investigation, these technologies are investigated and tested in great depth in a variety of different contexts pertaining to the issue. In the following section, how these technologies are being used in managing the multi cloud environment is discussed in detail.

3.1.1 TECHNOLOGIES SUPPORTING MULTI-CLOUD MANAGEMENT

In this section, we will explore some of the most important technologies that have emerged over the past several years for the purpose of managing and supporting operations in multi-cloud environments. Among the most important concerns that need to be addressed in multi-cloud:

a. Portability of application across clouds.

b. Service discovery and management across clouds

c. Orchestration of services available across clouds

d. Operation management in terms of build, release and deployment

e. Resource management and optimization.

After a few years had passed, it was challenging to address the points that were listed above. However, with the introduction of new technologies like as containers, devops, micro services, and orchestration tools such as kubernetes, it is now simple and efficient to manage these operations in a seamless manner while reducing the amount of time and money involved. These technologies are going to be addressed in greater detail in the following sections, along with how they are solving the difficulties that were described earlier in relation to multi-cloud administration.

Micro services

The latest paradigms of software development approach are known as micro services. These services are designed to facilitate the development of applications that are scalable, portable, and distributed across several locations. That the services can be created isolated, that they can be loosely connected, and that they can be merged with applications that are polyglot in nature are all examples of how it differs from traditional application development. Developers are able to create their services using their preferred programming language, such as Java, .NET, Python, and so on, and these services can easily be integrated with one another. When each service is put on any system, it is possible to make it functional independently and does not need to depend on the functionality of other services. The management, maintenance, and

monitoring of such services are relatively simple, and developers may quickly adapt to being able to deal with them.

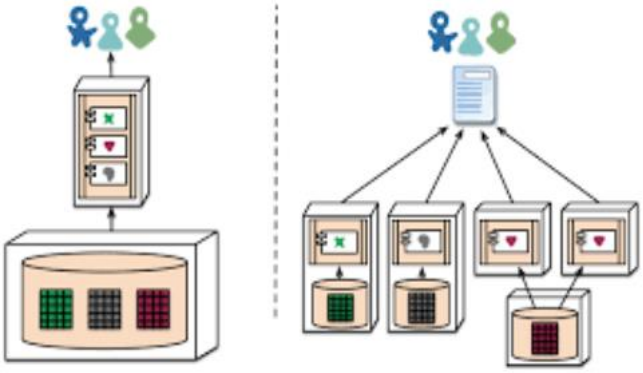


Figure 3. 1 Monolithic and Micro services

The distinguishing characteristics of monolithic and micro services are illustrated in Figure 3.1. In distributed systems, where scalability and reliability are the most important factors to take into consideration, micro services are designed to handle the workload. Services that are small and finely grained are required for these systems.

These services must also have their own lifecycle management, and they must work together in order to complete a specific task or feature. Micro services, in general, are created based on the business domains, and they assist in overcoming the typical challenges associated with multi-tiered architecture in terms of tight integration between service tiers. Additionally, micro services are more adaptable and flexible, and they may be quickly integrated with any new technologies and methods that have evolved in recent years.

They are also helpful in resolving important risks and concerns that are associated with the development of much service-oriented architecture due to the highly disconnected nature of these implementations. It is one of the most important programming concepts that is utilized in multi-cloud environments, which are characterized by the presence of key characteristics such as distribution, scalability, independence, security, and availability.

The design of such a micro service application is relatively straightforward and straightforward to install on any cloud environment.

Docker and Container

In comparison to virtual machines, containers are a lightweight component that are the new evolution in the area of infrastructure modernization. Containers are also the new evolution in the space. When it comes to containers, the operating system is removed and then made common, so that it may be shared throughout all of the containers. The only components that are included in containers are runtime libraries, binaries, and dependencies; as a result, containers are lightweight in nature. Containers, in general, are designed to improve the deployment process by containing all of the details for an application. This makes it easier to automate the deployment process, manage versions, and deploy applications quickly. This technology of containers is not new; in fact, it has been around for quite some time. Docker containers are the most extensively used and highly acknowledged containers in the business. They facilitate easy portability and deployment across a variety of cloud platforms and come with a comprehensive ecosystem that allows for their management.

Containerization is the new mechanism for packaging and deploying programs in any cloud environment where it is necessary for the application to be portable and predictable. The packaging of components and the relationships between them is accomplished through the use of containers, which are standardized, separated, and lightweight process environments. There were a lot of businesses that wanted to deploy applications that were bundled in the form of containers so that they could be dispersed and so that the system could quickly scale to deal with situations where applications failed.

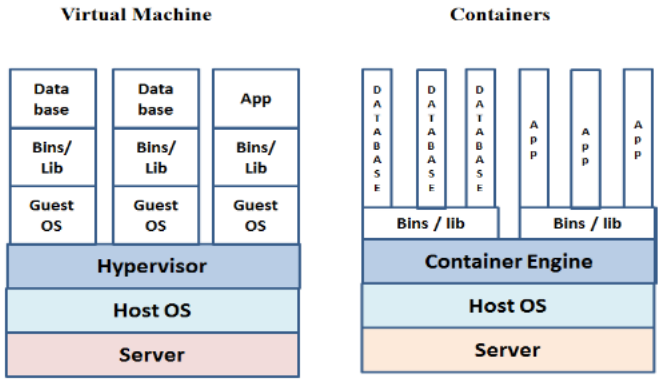


Figure 3. 2 Virtual Machine and Container (Docker)

Figure 3.2 illustrates the distinction between a virtual machine and a container that is associated with Docker. The Docker platform is an open-source platform that allows developers and operations teams to construct, ship, and run applications that are distributed. Docker makes it possible for businesses to standardize deployment in a variety of contexts and assists them in simply adapting to this kind of service design and management.

Docker-based infrastructure modernization has been accomplished at a number of businesses by utilizing its ecosystem, and it has shown to be an extremely efficient method for managing distributed applications. The Docker registry allows for improved registration and management of micro services, and the capabilities that are enabled by Docker for micro services deployment help with service management and archiving efforts.

When they were first developed, containers were intended to provide support for stateless applications. The term "stateless" refers to the fact that the information that is produced during one session is not saved for use during another specific session.

In the modern world, there are a lot of applications that need to record user activity, and because of this, the application needs to be stateful. The most recent version of the Docker container helps to support both stateless and stateful parts of an application. These aspects can be customized in a manner that is appropriate for the requirements of a complicated application. The Docker container paradigm makes it possible for an application to keep its persistent state in a separate storage that can be shared among several Docker application containers in a distributed environment. Docker has proven to be the most successful deployment methodology in the development of modern applications due to its unique support for both stateless and stateful natures.

3.1.2 DOCKER ECOSYSTEM

The Docker Ecosystem offers a comprehensive ecosystem that enables users to construct, deploy, and execute Docker images in an environment that is genuinely distributed. The fact that it is composed of a small number of components enables them to properly control the application component. Figure 3.3 provides an explanation of the entire ecosystem that facilitates container management.

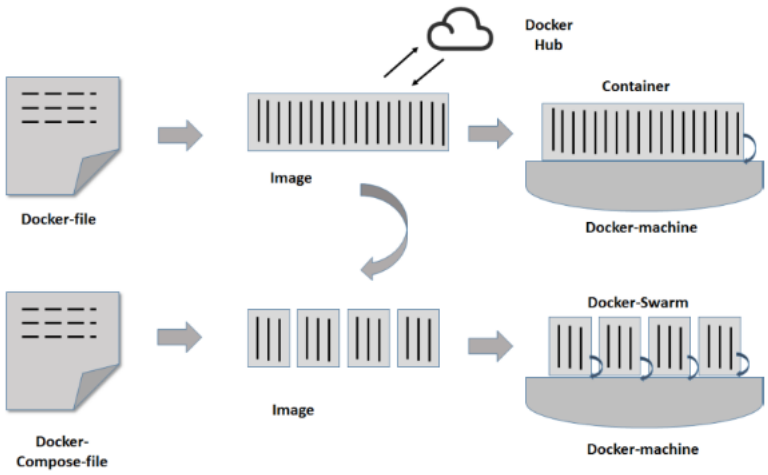


Figure 3. 3 Docker Eco-System

• Docker engine - creates and runs docker containers as live running instance in the form of docker image.

• Docker image is a file created in the form of a text document which are used to run a specific service or program in a selected operating system. User can create a docker image with all the commands which the docker file contains through command line.

• Docker Hub is a docker registry and is hosted in a cloud. User can have different version of docker registry to create, test, and store and distribute container images.

• Docker machine helps in provision of docker engine on virtual hosts, and Docker machine commands are used to manage the hosts.

• Docker Swarm is an orchestration tool, used for clustering, scheduling and it automatically optimizes a distributed application's infrastructure based on the application's lifecycle stage, container usage and performance needs.

• Docker-compose enable orchestration across multiple containers and helps in balancing load by writing container dependencies in a YAML file and managing via docker UI.

Container as a Service (CaaS)

Containers-as-a-Service, also known as CaaS, is a type of container-based virtualization in which the container engines, orchestration, and the underlying

compute resources are provided to customers by a cloud provider in the form of a service. CaaS, which stands for containers as a service, is a self-service approach that allows for the construction and deployment of applications. This model assists any IT organization in managing and securing application content on any infrastructure. CaaS is a type of container-based virtualization in which the container engines, orchestration, and the underlying compute resources are provided to customers "as a service" from a cloud provider. CaaS is also known as containerized application virtualization.

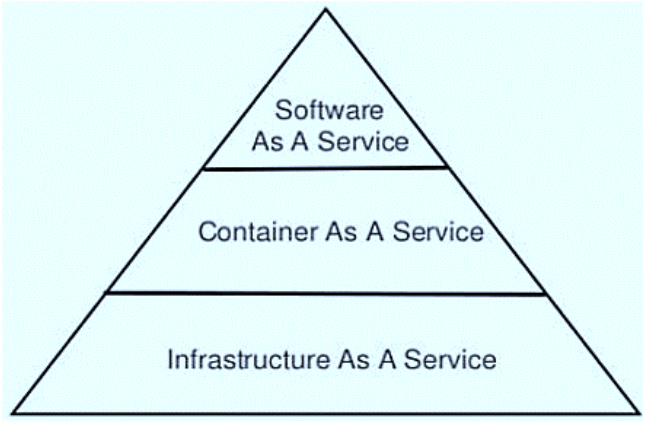


Figure 3. 4 CaaS Service Model

(CaaS) and (IaaS) are shown in Figure 3.4, which explains their relationship. Container services are one of the several deployment options offered by every single cloud operator. Most people consider (CaaS) to be a subset of (IaaS), particularly given its position between IaaS and PaaS. You may use CaaS on a bare metal system or within a (VM). When it comes to (IaaS) environments, CaaS is the way to go since it is container-driven. CaaS products give the agility that is necessary for architecting solutions that make use of containers. They also make it possible for DevOps teams to automate the check-in process and assist the deployment of containers onto any production system in a short amount of time.

Container Orchestration

In the context of to provide users with a collective reaction, it is essential to install applications in a dispersed manner and then coordinate and orchestrate them. Because these apps are often housed in groups called clusters, it's crucial to have a good grasp on cluster administration, scheduling, load balancing, and service management. The

goal of container orchestration is to facilitate the early deployment of containers in multi-container packaged applications so that they may be coordinated in the cloud. Moreover, it makes it easier to control the availability of several containers as if they were a single entity, scaling, and networking. An explanation of how the multi-cloud orchestration is carried out by means of Kubernetes is provided in Figure 3.5.

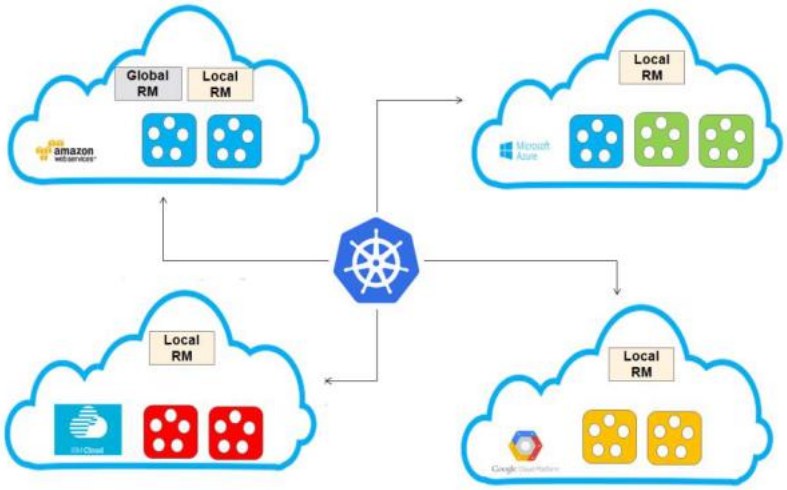


Figure 3. 5 Orchestration with kubernetes in multi-cloud

Service discovery

The development and launch of a truly distributed application necessitates the distribution of various services over several data centers and physical locations. Finding the right service, binding and carrying it out, and seeing it through to a successful completion is a major challenge in managing the execution of a job. The produced services are essentially recorded in a registry that is dedicated to services. Any request for a particular service component will search the registry to find out the details of that service, including where it is available, deployed, and functioning. The request will be able to find and bind with the services much more swiftly because of this. Also, as mentioned earlier, the orchestration tool can handle service discovery and administration, so the user won't have to worry about that either.

Resource Management

Containers are designed to run applications, and in order to ensure that these applications are successfully executed, they require a collection of resources that can

be controlled and monitored in an effective manner. The monitoring and control of the resources that are being allotted to these containers on a periodic basis, such as the central processing unit (CPU), network, input/output (I/O), and memory, has to be done eventually. Doing so ensures that these containers can function autonomously with the resources they contain. Incorporating it within the CaaS paradigm allows for full control and oversight of these assets. It also provides a dashboard that, with the help of these CaaS services, can measure, monitor, and manage a number of indicators.

Scheduler

When it comes to container as a service, the scheduler is a crucial component that helps with the execution of tasks and batch operations. It is possible to load a certain container into a host system using scheduling. The process of loading a service file does this which provides the administrator with assistance in managing and maintaining the container. The scheduler is a component that is built into the majority of service providers that include container managed services. In order to execute a container service on a cluster, schedulers are used to maintain state, perform various back-end operations, and aid in host selection. When it comes to running containers, schedulers are in charge of implementing the restrictions that administrators have specified and defined.

3.1.3 STRATEGY TO MANAGE MULTI-CLOUD

One of the most significant challenges that businesses are currently facing is the management of multi-cloud environments. Managing a single cloud is something that many people do, but the situation is different when it comes to managing many clouds. Companies are confronted with a number of obstaclesthe majority of which are to matters pertaining to operations, performance, security, and collaboration, and orchestration, discovery of services, scheduling, and management of resources, storage, and workloads. The fact that each service provider has its own distinct rules, procedures, and processes makes it exceedingly difficult to collaborate on the creation of industry-wide best practices for managing multiple clouds. Although there are ongoing efforts to build frameworks and tools, no one-size-fits-all solution has been put into place to address the problem thoroughly. To have an end-to-end coverage that

handles all of the difficulties, the adoption of these technologies presents both obstacles and solutions, which will be described in the next portion.

3.1.4 MANAGING RESOURCES IN MULTI-CLOUD

One of the biggest challenges in managing several clouds is optimizing and managing resources. The creation, management, and destruction of resources such as virtual machines (VMs), containers, networks, storage, and services, among other things, are essential tasks for operational engineers. When it comes to multi-cloud, the task becomes even more difficult because each cloud has its own unique resource types and operational procedures, and multi-cloud management is accomplished through a single interface.

Furthermore, virtual machines (VMs) and containers are considered to be vital resources. As a result, the management, consolidation, and migration of these resources to various clouds must be properly addressed by a system that assists in the effective management of critical resources.

When it comes to managing and keeping tabs on your resources, a dashboard is an essential tool that every cloud provider provides. The user may also customize resources for improved functioning and track their lifespan using the dashboard.

VM consolidation

Virtual machines (VMs) are a great way for businesses to manage their workloads and applications, and it's likely that any organization using cloud services has done just that. Due to the fact that different departments use cloud resources, there is a possibility of deploying separate resources in the cloud.

Any company serious about getting the most out of its cloud resources in terms of usage and cost must make cloud monitoring, administration, and optimization a top priority. Overall resource optimization will include dynamic monitoring and consolidation of these virtual computers.

With the use of virtual machine consolidation methods, a typical cloud data center may optimize resource utilization and minimize energy consumption. Intelligent decisions on whether to sleep and wake up virtual machines (VMs) are made possible

via this consolidation by tracking workload changes on the VMs. In order to save energy consumption, hosts will go into sleep mode when virtual machines (VMs) are not in use or are only partially employed. What follows is a discussion of a few of the difficulties.

- Migration: The host determines which virtual machines (VMs) to use based on the workloads and applications' utilization and consolidation. Transferring virtual machines from over-utilized to under-utilized states and releasing unused, empty, or inactive VMs are all part of this. Second, transfer the virtual machines to a different host or cloud in order to consolidate them.

- Planning for capacity is something in order to ensure effective management of resources. Depending on the quantity of work that needs to be handled, it is necessary to predict how many virtual machines (VMs) will be needed to match the workload capacity. If the number of virtual machines (VMs) is more than the task's capacity, then go forward with the workload migration and release a limited number of VMs.

- Provisioning: To provide or de-provision virtual machines (VMs) based on their continual consumption, as well as to monitor VMs It is necessary to deploy new virtual machines (VMs) in order to handle the workload if there are further workloads waiting for the availability of resources.

Therefore, having capacity management solutions that can anticipate and control resource requirements is critical for effective virtual machine (VM) administration. Virtual machines (VMs) should also be clustered in an area with enough of power and other resources to back them up.

In order for the system to work as intended, it has to include prediction algorithms that consider the use of virtual machines (VMs) across several clouds and provide a forecast based on the capacity needed for future migrations.

See Figure 3.6 for a rundown of how virtual machine consolidation works in its entirety in any cloud setting.

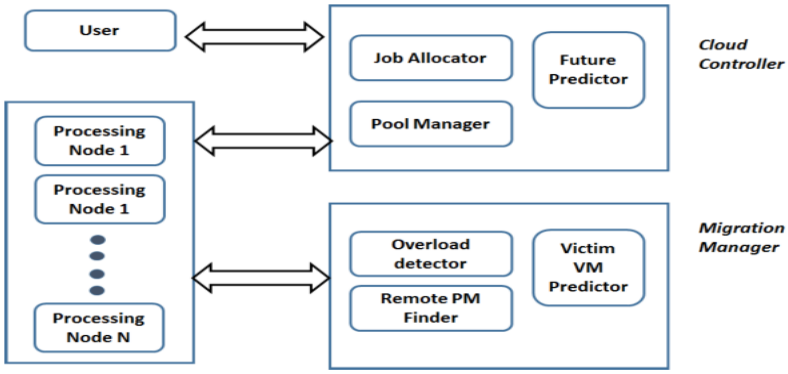


Figure 3. 6 VM Consolidation

The suggested structure incorporates the following critical components for virtual machine consolidation.

- Uses usage detector to identify system overload circumstances
- Uses candidate VM detector to identify which virtual machine requires migration.
- Identifies the workloads that need to be redistributed for VM performance optimization and decides which cloud the group of VMs to move to should reside in.

The following sections go into great depth on each of the aforementioned criteria.

Container Migration

Container migration is now the main focus of the company since it is more portable, lightweight, and polyglot in nature. Containers, which are built using dockers and used for different workloads, are then placed in virtual machines (VMs) on different cloud hosts. The convenience of containers has led to their widespread usage; in fact, many product suppliers now produce their products in container form. Transferring containers across different cloud platforms or even between different virtual machines (VMs) is a breeze. The problem has prompted the development of a plethora of solutions.

Because of containers' inherent ability to isolate resources across numerous containers that are contained inside the same kernel, their use is on the rise when it comes to hosting applications. More easily accessible and scalable containers are in high demand due to the rising popularity of container technology. One possible answer to

this issue is a method for moving containers quickly from one computer to another without affecting the applications' functionality. Still, containers should be able to move to a new server in real time without affecting the programs running on the old server. In order to reach the goals of load balancing, OS updates, and hardware replacement or maintenance, container migration is required. The simplest approach may be cold migration, which entails stopping the container at the source, copying it to the destination, and then restarting it there. Essential steps in cold migration may include storing the state locally, moving it to the destination, and then restoring it. For live migration, the migration technique presupposes that there will be zero downtime. There are three steps to transfer a container's state to a remote host. Process state, the file system, and RAM are these entities. Whenever a container is migrating and memory transfer is involved, the majority of its memory pages are moved to the remote host. Immediately after the transfer of the memory state, the file system is copied. The next step, after transferring the file system and memory, is to provide the remote host the process state of each individual process. This includes information about their registers, open files, and sockets.

The Docker engine manages and creates container images in the background. There are tools available to help construct the image and add it to the registry, which is also known as the Docker trusted registry (DTR) and may be public or private.

The finished photographs are saved to a registry, and anyone with an internet connection may access them and use them. Container status monitoring is necessary according to the choice to move, whether it's a container at rest or a live migration, or the technique to be used in the migration plan. Whatever choice was taken, this is done accordingly. Important concerns that must be addressed throughout the migration process include scheduling and deployment of containers.

Workload Placement

A critical component of multi-cloud management is the approach for scheduling and placing workloads. There are often two tiers to managing workloads and tasks.

a) Optimal performance and efficient management of the current workload via redistribution between virtual machines and the cloud.

b) Keeping track of incoming workloads and tasks and ensuring their proper distribution.

The primary difficulty in both situations is determining which virtual machines (VMs) are best suited to run the specified workloads.

Task scheduling is an important role in cloud computing. For this assignment to provide the greatest results, it is necessary to complete a number of activities within the given start and end times while keeping to certain constraints. Time limits or resource constraints are only two examples of the many potential types of limitations. Task scheduling allows for the distribution of work among several processors, which in turn maximizes resource consumption and minimizes execution time. People usually think about things like make-span, energy usage, service level agreement (SLA), cost of computation, cost of data transmission, and similar things while trying to manage workloads efficiently. Cutting down on execution costs and waiting times should be high on the list of priorities, along with addressing resource availability and scheduling issues that hinder efficient use of available resources. Before deciding on the right virtual machines (VMs) and cloud services to deploy, it is critical to ascertain if the workload is CPU or RAM intensive.

The difficulty of task scheduling in the cloud is addressed by using a diverse range of scheduling strategies. Cloud task scheduling makes use of a number of techniques to improve server speed and throughput while making better use of available resources.

- The popular scheduling method known as the Cuckoo search algorithm (CSA) is undergoing development and experimentation in the realm of cloud computing. Job scheduling involves watching and mimicking the cuckoo bird's typical behavior. The cuckoo will choose a nest at random and put its eggs in it one at a time. After then, the host bird's chances of finding an alien egg are somewhere in the middle. The algorithm's speed and coverage reach new heights when the probability value is low.

- Growing cloud providers' bottom lines is the major goal of another popular approach to work scheduling that relies on genetic algorithms (GA). The GA function evaluates the population by making use of the fitness function, which considers user happiness and the availability of virtual machines as criterion. In

addition, it comes up with a set of methods for scheduling tasks that boost performance. In order to generate the optimal timetable for tasks, the function iteratively reproduces the populations.

- Other well-known algorithms for task scheduling include the Ant colony optimization (ABC), the Bee Colony Optimization (BCO), the Particle Swarm Algorithm (PSA), and many others. The heuristic approach is the foundation around which these algorithms are built. They are used to search for and locate the appropriate resources by utilizing a variety of methods. Within the scope of this investigation, an enhanced version of these methods is utilized for the purpose of resource placement and optimization within the cloud. The scheduling of applications to cloud resources is accomplished through the utilization of these heuristic approaches, which take into consideration the costs of computation as well as the costs of data transmission.

In the context of a multi-cloud architecture, each of the aforementioned methods is helping to improve speed and throughput by elucidating the challenges of resource optimization. When deciding on a scheduling algorithm and approach, the following are the main factors that are considered.

a) The minimum acceptable distance or network delay to the specified resource.

b) There need to be a bare minimum of time between the execution of the task or its transfer to the cloud resource.

c) There ought to be no placement fee.

d) It is expected that SLA, Compliance, and Performance fulfil the requirements.

Within the scope of this investigation, following a substantial amount of analysis and investigation, We found the best way to schedule jobs and deploy them on cloud resources while meeting all the requirements listed above.

The Support Vector Machine (SVM) is one such approach; it uses a weighted sum of each virtual machine's power to choose the one that's best suited to do a given job, cost, and execution time.

Knowledge management and Decision making

The management of knowledge is a crucial component of applications in general. Either the prior knowledge of a system that was comparable to the one being studied or the knowledge gathered from the current system via the process of observing its behaviour over a period of time was utilized by the researchers. The important features of the present trend in research are the construction of the knowledge database and the application of the gained knowledge to the process of making a crucial decision through the utilization of machine learning algorithms. The key factors considered are

a) Instructions for creating the database for managing knowledge

b) Ways to discover new things and put that information to use for forecasting

c) The best way to assess the findings in light of how the information was used to enhance the results' accuracy.

There is a wide variety of algorithms accessible, which can be implemented in a deterministic or heuristic manner. The initial step, which is known as "learning" or "training" the system, involves the system learning with a predetermined outcome. As time passes, the number of mistakes that occur during the learning process decreases, and after a few rounds, it reaches a point of saturation. If you want to predictions based on previous knowledge, predictive analytics is utilized. There are alternative sets of algorithms, such as the Markov Decision Process (MDP), that perform learning and prediction based on the behaviour that is occurring right now rather than on the behaviour that occurred in the past. Either strategy can be utilized, depending on the circumstances, in order to arrive at a choice that is corrective.

Storage management

The container technology offers a straightforward approach to the creation, management, and execution of applications, characterized by a high degree of flexibility and efficiency. When containers were first introduced, they were believed to be stateless by default. This meant that when they were used to run applications such as databases, which require persistent storage, they would lose all of their data

when instead of preserving the data, they were erased. Persistence storage was not a default feature of earlier versions of Docker. Therefore, it becomes an issue for clients that want to execute many kinds of workloads in containers. This occurs because each time a new container image is created, all of the data is erased. In contrast, when it comes to managing storage across many clouds, there are approaches to establish persistent storage in Docker that are more urgent. Docker offers technologies that make it possible to store data in a permanent manner during the lifespan of a container. Mounting a host directory, data volume containers, data volumes, and docker storage plug-ins are the four methods by which Docker containers may be given with persistent storage. Extending the discussion beyond the scope of this article would make it impossible to cover all of these methods in depth. Despite these solutions' best efforts, they fall short of fully addressing the requirement of data permanence in managing an environment with numerous clouds. Docker volumes, as default features, can only access data stored on the file system of the server hosting the container.

Among Docker volumes, this is the biggest drawback. In most cases, this should work just fine for testing and development environments; however, it is not suitable for environments that are used for production for the reasons that are listed below.

• Any data stored on the original host will not be accessible during server rebuilding.

• Data protection and management strategies are not easy to implement with local volumes.

With Swarm or Kubernetes, application owners can deploy containers to multiple servers, but they have limited control over which server is used.

There will be difficulties and dangers associated with moving massive amounts of data from one cloud to another. When it comes to storage management in multi-cloud, these are the main issues that this research has found.

a) Access to equivalent resources in a different cloud, data migration, and confirmation of data after processing.

b) Ensuring the safety of data during cloud migration.

b) Target cloud data compliance and integrity.

d) Data storage in the desired cloud.

e) Information corruption while in route

The purpose of this study is to provide a system that will handle the issues mentioned above and guarantee that the management of data and storage will be considerably addressed in an environment that utilizes several clouds. There are storage services that are exclusive to each cloud service.

Data loss during transit due to security holes happens, nevertheless, when storage is transferred from one cloud to another. This is where the issue starts. The constant use of data apps and the transferability of data between platforms will also provide substantial obstacles. A methodology for safe storage management within a multi-cloud system is proposed by the results of this research.

3.1.5 OPERATION MANAGEMENT

DevOps in multi-cloud

The DevOps methodology is an approach to software development that brings together the domains of software engineering and IT operations. Consistently providing additions, fixes, and updates while shortening the systems development life cycle is the purpose of DevOps, which is in line with the company's goals. DevOps is not a technology in and of itself.

DevOps is a method that brings together the development team and the operation team. It also assists in bringing the development and operational tasks much closer together and makes them more collaborative.

There are a great number of tools that are frequently used, and some of these tools include continuous integration and continuous delivery or continuous deployment, specifically with an emphasis on task automation. In addition to collaboration platforms, there are only a handful of other solutions that can provide assistance for DevOps. Among these offerings is an incident response system and real-time monitoring capabilities.

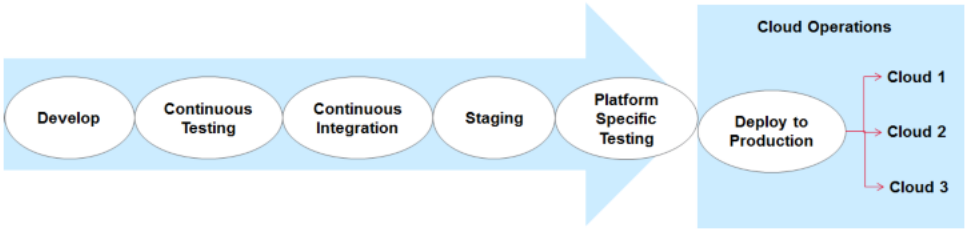


Figure 3. 7 Devops in multi-cloud

Before deploying to any cloud environment, the DevOps lifecycle must be completed, as shown in Figure 3.7. The details of these steps will not be addressed here because they are outside the scope of the current conversation. Developed in-house by each cloud provider, DevOps services are there to help with the management of all the operational tasks that happen in the cloud. Managing development, releases, and deployments in multi-cloud environments presents a number of challenges for DevOps. Consequently, there is a plethora of multi-cloud operations management frameworks and tools to choose from.

Automation

Increasing the automation of manual processes in order to boost production, decrease the number of errors, increase efficiency, and lower costs is the primary goal of many firms in the modern era. Modern technologies, such as automation that is based in the cloud, provide solutions to these issues and enhance operational excellence in accordance with the business objectives of the consumers and users. Numerous automation technologies, such as ANT, maven, bamboo, Jenkins, and cloud bee, among others, are available for the purpose of automating the process of building, deploying, and releasing application packages. There are many steps in the software development lifecycle that these technologies help automate. Supply chain management, configuration oversight, and deployment across infrastructure components of numerous cloud resources are all made easier with the assistance of technologies such as Chef, puppet, and Ansible. Many organizations are making automation a continual priority area in order to have a greater focus on optimizing processes, standardizing procedures and governance, improving security and performance, and reducing errors that are caused by tasks that are performed manually. Automation allows an organization to concentrate more on the creation of

new solutions and reduces a measure of the frequency with which existing operations are monitored and optimized; with increasing automation, companies may optimize their operational and financial advantages; and they can also maximize their capabilities to concentrate more on the creation of new applications and increase their "Time to Market" for their solutions in comparison to those of their competitors. There are several problems that must be properly studied and handled in order to successfully manage the DevOps process in an environment that utilizes multiple clouds. In this research, a number of these operational problems are investigated, and solutions are presented as a component of the framework architecture for the multi-cloud.

Networking and Interoperability

Multi-cloud networking makes it possible for many cloud platforms to be efficiently had a common ground and worked together. Because of this, apps running in various clouds are able to interact with one other effectively. Overlay networks are a concept in cloud networking that allow containers to interact with one other in a cloud environment. It is a way to run many separate virtualized network layers on top of the actual network. This approach combines software engineering with IT operations; it is used to build network abstraction layers. By using this approach, additional security advantages are often added to the cloud-deployed application. To put it simply, Flannel and similar technologies make it easy to set up an overlay network nationwide in the cloud. The Docker engine provides multi-host networking as a pre-installed feature by using the idea of an overlay network. By extending the overlay network to hosts running on multiple clouds, it becomes feasible to provide seamless communication between containers running in all of those clouds. For the sake of argument, let's say that Open Stack hosts the web layer of a typical multi-tier application, Microsoft Azure hosts the database tier, and Amazon Web Services hosts the application tier. Then, by utilizing an overlay network, it is possible to make these three entities communicate with one another.

The term "interoperability" relates to the capacity of cloud echo systems to exchange data and information across different levels of cloud service providers to comprehend each other's intentions concerning the exchange of communications and the facilitation of portability between them is what makes this capability possible. A set

of guidelines and standards for describing interoperability is currently in development.

a) The capacity to transfer applications across different cloud service providers.

b) Capacity and skill to transfer data between different cloud services.

c) Reduce certification and compliance costs as much as possible.

d) Implement open access/open source policies via API extensions and make them available.

a) Steer clear of becoming stuck with a single cloud provider.

f) Make technology ineffective.

Standards are necessary at every level, including infrastructure, platforms, applications, services, data, and management, in order to get the necessary degree of compatibility. As an example of one of the few standards that now exist, consider the (OVF) and the (CDMI). To simplify portability, the former only means enclosing virtual machines (VMs). In order to access and manage the cloud storage solution, the latter must choose which interface will be used. Connecting multiple hosts using Docker is achieved by use of an overlay network, as seen in Figure 3.8.

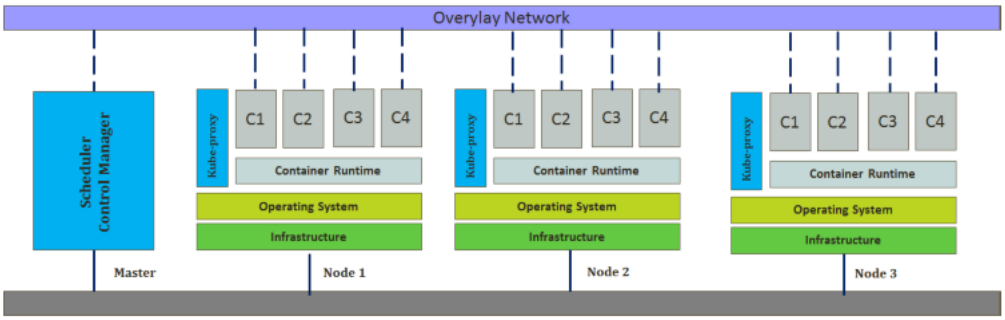


Figure 3. 8 Multi-cloud networking with overlay network

Effective multi-cloud management relies heavily on networking and interoperability as its two most important components. There are a lot of cloud orchestrating platforms available nowadays that efficiently address these aspects and take care of the majority of the issues that were stated previously. A key component of every cloud service provider's offering nowadays is an orchestrator like Kubernetes, which helps with

managing and orchestrating services in a distributed cloud environment. Handling the administration of application development and deployment across several clouds is a top priority for modern businesses. In addition to the languages they support, API services, and deployment choices, each cloud provider has their own unique set of offerings. Managing several clouds presents a number of challenges, the most significant of which is portability. However, with the assistance of micro services and dockers, this challenge is substantially addressed, allowing applications to be moved across clouds with ease. Nevertheless, Services are made accessible in different clouds during the distributed application deployment in multi-cloud. So, it's not easy to find services, bind them, and collaborate with all the different service providers. Use of orchestration solutions like Kubernetes can resolve most of these issues. A unified deployment platform will manage the whole application lifecycle, from development to release and deployment, and will execute CI/CD procedures across all cloud environments. A unified, self-service enabled dashboard is necessary for people to be able to install applications across cloud environments and monitor them. Building and deploying in an orchestrated manner allows for the efficient coordination and administration of several concurrent deployments. CI/CD is performed across a number of different cloud environments by tools such as Jenkins, Travis, CircleCI, and Terraform, amongst others.

The architecture of applications that utilize micro services enables the creation and deployment of distributed applications effectively. By registering these services with the registry, other apps and services will be able to discover them, bind to them, and start using them in a coordinated fashion. As an integral aspect of operations, managing the build is one of the most critical challenges in multi-cloud application administration, release, and deployment process.

Monitoring and Optimization

Key operational duties include monitoring cloud resources across all cloud environments and participating in multi-cloud. This is because the consumption of cloud resources, utilization of cloud resources, performance of cloud resources, it is essential to constantly analyze and change various elements of cloud resources to manage expenses and improve operational effectiveness. The use of monitoring systems that keep tabs on all resources is continuous monitoring, including

applications, networks, storage, and infrastructure, among other things, and assess metrics based on these characteristics at regular intervals. In accordance with the organization's standards, these metrics are evaluated in comparison to the performance benchmark, and decisions are made on the management of the resources. The deployment of operational engineers allows for the monitoring of resources, the making of choices and the implementation of corrective actions in response to the measured metrics, and the optimization of the environment on a periodic basis.

Security, policy and governance

The most critical aspect of multi-cloud management is the control and oversight of security measures across all cloud services. This is because it could be challenging to create a centralized standard method when every service provider has their own distinct set of regulations and procedures. There are a lot of security concerns when switching clouds. This is because infrastructure and applications are cross-platform. No matter the vulnerability, it is critical to protect apps and data during transit and storage from any kind of hacking. Properly encrypting and decrypting data is essential for a trouble-free data transit across clouds. A number of security standards are now in development with the goal of achieving uniformity in cloud computing. Governance is an integral part of multi-cloud administration, much as application, data, and infrastructure management. Governance includes a number of essential components, the most critical of which are process monitoring, protocol confirmation, and anomaly discovery. Organizations must enforce the rules and regulations, including security policies, to guarantee that they are being followed and handled in a standard and normative manner. Anyone found to be in violation of these procedures is to face severe consequences, and any change from the norm must be documented and reported to the proper authorities.

Every sector has its own unique set of security rules and laws, and every company has its own internal policies, guidelines, and guiding principles for leadership. A number of businesses are striving to establish these standards as industry-wide recommendations by staying true to the security idea and, in due time, being more forthcoming and honest with their customers. Raising awareness of these standards is a primary goal for these groups. Due to a lack of full alignment between corporate policies and processes and the common standards, the industry is finding it difficult to

either fully embrace or disregard these standards. The industry is currently in the early stages of developing these standards. Additionally, owing to the abundance of security rules, the governments of all countries and regions have instituted their own unique sets of security constraints.

3.2 MULTI-CLOUD MANAGEMENT

When it comes to multi-cloud management, there are a lot of issues that need to be handled because it involves a variety of cloud services, each of which has its own set of characteristics and operating methods. There are a few frameworks that have begun to emerge in the market for managing multi-cloud, and some of these frameworks are successful in solving only a few of these difficulties.

The purpose of this research effort is to design a strong framework that can address the majority of the critical difficulties, such as continuously monitoring, managing, and optimizing the multi-cloud environment, as well as performing continuous optimization in order to improve the outcome in terms of cost reduction and optimal performance. In order to satisfy the target functions, the framework that was built is intended to address the end-to-end management of the heterogeneous cloud environment.

This will be accomplished through the utilization of a centralized monitoring system, which will measure the metrics, conduct analysis, and optimize the system using a variety of methodologies.

In addition to addressing optimization on a global scale across several clouds, the framework was built to also do optimization on a virtual machine (VM) level within a single cloud. When it comes to application deployment, containers are taken into consideration, and metrics are monitored at the infrastructure, virtual machine (VM), and container levels on a periodic basis.

These measurements are then archived for subsequent analysis. Five primary components make up this framework, all of which are strongly coupled and have functions that are highly interrelated to one another. Here, we take a close look at the framework at every level and explain how it's helping with the continuous optimization of cloud resources.

Analysis on existing framework

In the past, a great number of scholars have conducted a great deal of study on cloud optimization and experimented with a variety of heuristic and deterministic statistical approaches in relation to cloud optimization on a single cloud. A small number of researchers have extended their work in multi-cloud administration optimization in the last many years. This has occurred because multi-cloud and hybrid cloud are gaining a lot of momentum in terms of avoiding vendor lock-in. For the purpose of managing several clouds, there are only a few standard frameworks and technologies that are made available by various firms and research organizations. Such tools as Right Scale and Clickr are examples. These frameworks include features such as cloud brokerage, service management, capacity management, and operational management, among others; however, none of them address how to select or rank the cloud for the purpose of handling a specific application (workload or tasks) for deployment, continuously optimize and use the prior knowledge of the system, and make use of the system's exiting state. In the course of this investigation, doing so allows for a thorough assessment of the system's present status: continuously monitoring cloud resources in order to gather parameters; the selection of the appropriate subset of parameters by means of Particle Swarm Optimization; the ranking of clouds by means of Analytic Hierarchical Processing (AHP) for cloud selection; capacity management with prediction; and resource optimization by means of Reenforcement Learning (RL).

Not only does the suggested framework analyse the present state and make decisions, but it also makes use of the prior information gathered through the utilization of the knowledge management framework in order to make the decision more effective. As a result, it is exclusive in comparison to other frameworks that are comparable and are now accessible on the market.

By utilizing a strong framework that includes a comprehensive methodology, techniques, and approach that has been tested against a variety of scenarios and has been demonstrated to be effective in real-time settings, the primary purpose of this research is to address all of the significant problem areas that are associated with managing operations in multi-cloud environments. It is acknowledged that many people have machine learning in terms of prediction and optimization on cloud

parameters. This is something that is being investigated as part of this research. In a multi-cloud environment, it was discovered that there was not a lot of effort done on utilizing the previous information by learning about the system and using the knowledge management to decide on decision making.

This went beyond a simple analysis of the system's current status. In addition, this research makes a unique contribution to the field of multi-cloud management by building a framework that follows a specified procedure and incorporates new components based on historical behavior knowledge in order to predict future outcomes given the present condition.

Challenges in Multi-cloud management

There are a handful of critical issues with controlling the multi-cloud that must be resolved for the system to function at its optimal level and achieve maximum efficiency. Here are a few of the most important obstacles:

a. Continuous monitoring of participating cloud environment in a multi-cloud scenario and measure metrics boost the effectiveness of daily operations.

b. Continuous resource optimization to have an optimum resource running across clouds for cost optimization.

c. Portability of resources / applications across cloud and choosing the right cloud for particular set of application to perform better.

d. Resource consolidation and moving to a different cloud to maximize efficiency and enhance (ROI).

e. Bringing standards, best practices in terms of operations and management.

f. Improve decision making and accuracy in computation as the system is highly dynamic and random in nature.

In order to build a framework that can handle these problems, this study thoroughly analyzes them and then employs the best machine learning and optimization methods. comprehensively, from beginning to conclusion.

3.2.1 PROPOSED MULTI-CLOUD FRAMEWORK

The proposed multi-cloud management framework is unique in its approach to problems like monitoring, workload migration from one cloud to another with proper analysis and decision making, workload scheduling according to the strategy, and so on. It does this by utilizing algorithms that understand and analyze the current conditions and make corrective decisions regarding scheduling and optimization across environments. The suggested structure consists of five separate phases of operation. At least one problem with managing several clouds is addressed in each level of the system, which together provides a complete solution.

Stage 1: Monitoring multi-cloud - All of the resources in the participating cloud may be easily monitored by the centralized system. The goal of periodically acquiring these measurements for all of the participating clouds' primary resources is to facilitate further investigation.

After that, a feature selection method based on PSO algorithms is used to narrow them down to a minimum viable candidate list. Kubernetes is a tool for managing the interactions across participating clouds, which helps services work together more efficiently.

Stage 2: Ranking of Cloud - The choice of cloud provider to host new deployments and migrate existing apps to is a challenging one. The company's goals, which include things like performance, availability, network latency, and cost, will determine this choice. So, with the help of predefined business metrics like price, availability, consumption, network latency, and response time, a ranking-based system is put in place to choose the optimal cloud.

Stage 3: Scheduling and Placement - Once the right cloud has been selected for more research, it is critical to choose the right set of (VMs) to install the apps. We need to solve this problem next. The goal of the workload scheduling and placement strategy is to deploy the right set of applications onto the right set of virtual machines (VMs) within the selected cloud using optimization and machine learning techniques. Figure 3.9 depicts a task scheduler in addition to centralized monitoring and orchestration, which demonstrates how these two components contribute to the management of multi-cloud environments.

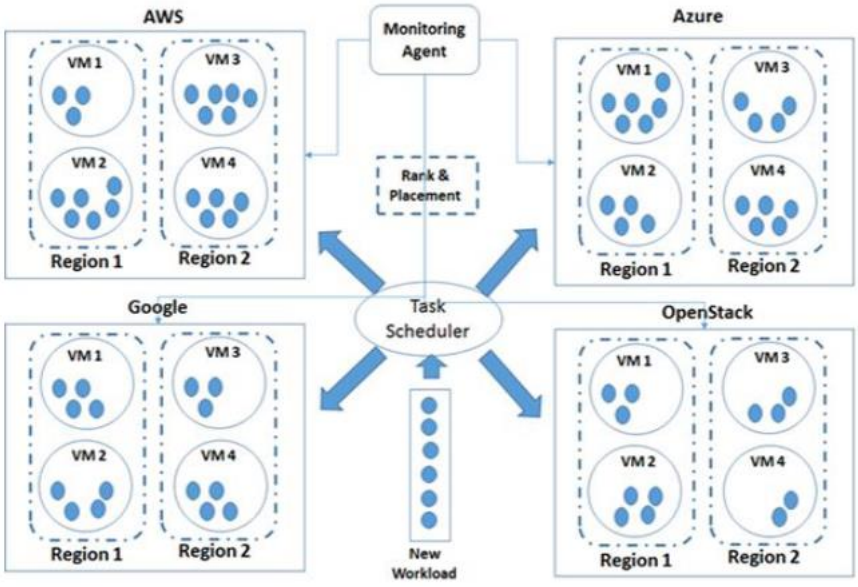


Figure 3. 9 Simulated Multi-cloud environment

Stage 4: Prediction and Optimization - A cloud environment's (virtual machines') resource requirements change dynamically over time in response to demand. Managing resource demand successfully requires frequent addition or removal of resources. In order to handle the workload (application) demand, we may recommend the amount of (VMs) needed with the use of a good forecasting method. Prediction methods are used to address this substantial challenge. Every cloud's (VMs) should share apps and workloads equally while using as few resources as feasible. An efficient optimization approach is one that facilitates the system's ability to accomplish this objective.

Stage 5: Knowledge Management - It makes use of both the historical data and the present state of the system to provide assistance for the decision-making process. The knowledge management component of the framework is responsible for accomplishing this goal.

The final point is that the speed at which decisions are made and analyses are performed ought to be high because there are several parameters involved in the various stages of operations. Accordingly, a resilient processing unit that is capable of running computation in parallel and in an asynchronous manner is designed to enable speedy decision making in light of the fact that the environment is constantly changing over time. Following the completion of ongoing research and analysis to

achieve the ultimate objective, algorithms are identified for each stage of the framework. These algorithms are then put into action and put through their paces in a number of controlled environments to ensure they provide the desired results. The following sections will go into detail on the algorithms and the results of the testing.

3.2.2 KEY RESEARCH AREA

Following a thorough examination of related studies and methods used in previous work, this investigation takes into account the following primary developing themes. Depending on the challenges that will be tackled in this study, the idea and principle of some of these techniques are expanded.

Machine Learning

(AI) is used in machine learning, which is a kind of artificial intelligence that enables computers to learn and grow in response to new data without human intervention. Computer programs allow algorithms for machine learning to access and process data for the purpose of enhancing their own learning capabilities. It makes use of statistical models, which are utilized by computer systems, in order to successfully carry out a certain task without the need for explicit instructions. Additionally, it employs scientific algorithms that are based on patterns and inference. In order to uncover potentially lucrative possibilities or potentially hazardous threats, it permits the processing of enormous amounts of data while simultaneously producing results that are both faster and more precise. The algorithms that are used in machine learning are computationally costly, which means that additional time and resources are required to train them correctly during the learning process in order to produce acceptable results. Machine learning, artificial intelligence, and cognitive technologies are typically combined by researchers in order to accomplish the goal of making AI even more efficient in the processing of enormous amounts of information. Within the business sector, there are two primary varieties of machine learning that are utilized.

a) Unsupervised Learning: The ability of computers to infer a function describing a hidden structure from unlabeled input is the focus of research on unsupervised learning. Research like this falls within the umbrella of education. These algorithms extract insights from datasets by mining unlabeled data for previously unknown

patterns. However, the majority of the times, these algorithms are not able to accurately anticipate the event that is sought.

b) Supervised Learning: An algorithm of this type makes use of labelled data in order to take in data from the past, process it, and then use that knowledge to forecast what's to come. The method requires a training phase and a testing phase. Upon completion of the training phase with labeled data, the system may provide objectives for all subsequent inputs. Additionally, the algorithm could check whether it's using the right intended result during the testing phase, which allows it to identify flaws and make adjustments to the model in accordance with those findings.

Re-enforcement Learning

RL, or RL, is a technique used in the field of machine learning. This approach uses an agent to continually engage with a system, which helps analyze and understand its present behaviors. The agent may gradually learn its behavior using RL, which relies on environmental input. Convergence is a common challenge with this RL approach, however certain RL algorithms may converge to the global optimum solution. If the issue is taken into consideration while modeling it. Rewarding desired behaviors and punishing undesirable ones is the foundation of reinforcement learning, a training approach. As a tool for managing unsupervised machine learning with incentives and punishments, the learning approach has gained popularity in the AI community. This automated learning method states that a human expert with domain knowledge is minimally necessary to guarantee that the agent understands the system to an acceptable degree. Figure 3.10 shows a picture of the Reinforcement Learning structure.

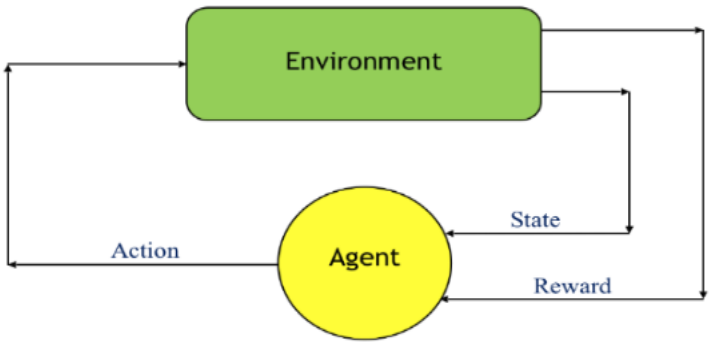


Figure 3. 10 Reinforcement Learning framework

RL can represent either a fully supervised learning experience or an uncontrolled learning experience. This method of learning is unique in that the agent learns from its surroundings while simultaneously carrying out its assigned duty. In this paradigm, the agent receives a reward when it learns the behaviours that are wanted, and it receives a punishment, which is negative actions, if it learns different behaviours. When the system achieves the intended behaviours, it receives positive reinforcement in the form of positive values. On the other hand, when undesirable behaviours are achieved, the system receives negative values as a kind of negative reinforcement. In order to arrive at the best possible solution, the agent is trained to look for the greatest possible overall payoff over the long run. Two steps comprise this method-

a) Training

b) Prediction.

An unusual kind of instruction called Q-Learning exists; it is defined by a set of regulations and a table that depicts a state-action pair. Through repeated interactions, the agent learns the system's behavior, and the table is updated with new rewards and penalties, allowing it to get closer to its objective. As part of the Markov decision process, the following sections provide detailed explanations of extra information. Reinforcement learning is a challenging approach, and researchers are always looking for ways to improve it. Many people believe that storing values for each stage is a very memory-intensive and complex operation, which in turn demands a lot of computer resources. When dealing with issues with large quantities of complex data, it is important to look at various value approximation approaches like Neural Networks and Decision Trees. Many problems arise when these erroneous value estimates are introduced; so, researchers strive to lessen the effect of these assumptions on the solution quality.

Markov Decision Process

To efficiently tackle a broad range of optimization issues, a Markov Decision Process (MDP) provides a framework for reinforcement learning. It describes a setting where a monitoring agent keeps tabs on things all the time, where the output is somewhat dependent on the agent's or decision maker's activities but also largely random. You

can see the progression from a Markov Process to a Markov Decision Process as seen in the picture below, which details the Markov Reward Process:

• A series of random states s1, s2, etc., that obeys the Markov property is called a Markov process or markov chain. Instead of looking at the system's history of states, it's more common to look at its present state while trying to forecast its future state.

• (MRP) is a kind of Markov Process that includes values, often known as a Markov chain.

• Decisions in a Markov Reward Process are known as a Markov Decision Process.

In order to achieve a goal, the MDP model is designed to learn the system via constant interaction with an agent. There is always a state-action pair associated with every single interaction that occurs with the system, and the agent's state is changed in reaction to the action that has been done. Every time the agent interacts with the environment, something changes, presenting the agent with a new set of circumstances. When a complete view of the environment is available, the MDP is used to describe it so that reinforcement learning may take place. The vast majority of RL problems are capable of being modeled as MDPs with the assistance of RL approaches. The subsequent section provides a comprehensive discussion of the MDP learning and prediction model, which is broken down into its component parts.

Recursive Neural Network (RNN)

A Recursive Neural Network, often known as an RNN, is a type of artificial neural network that was developed specifically to address the issue of particular recursion structures that are present in everyday events. In this model, the output of one neural network is given to the input of another neural network in a multi-stage model, and that output is occasionally given back to itself. This model is an extended form of a standard neural network. A significant amount of neural networks of this kind are utilized in the Deep Learning method, which requires one to comprehend many patterns present in an object and then merge these patterns into a single entity through the process of learning recursively. In addition to learning the knowledge of the present time, RNN also makes use of the information from the sequences that came before it. Similar to the back propagation process, the RNN algorithm may also turn

the propagation mistake around. The RNN process is also time-dependent, similar to the back propagation method. As a result of its one-of-a-kind architecture that uses chain components, RNN can deal with the issue of data preservation. On top of that, it has several clear benefits when working with time series. Also, the recursion vanishing gradient issue may be effectively handled with its help. If you have a little amount of data and do a lot of matrix multiplication, you'll see that the gradient decreases exponentially and goes away after a while. The matrix undergoes this transformation. The network becomes unable of learning data situated at long distances due to the gradient vanishing issue, which becomes apparent as the time interval increases... However, the gradient vanishing issue has been solved in the past using a variant of RNN called long short-term memory (LSTM). To improve the accuracy of the forecast, these solutions are used.

3.2.3 COMPONENTS OF THE FRAMEWORK

In accordance with the diagram presented in Figure 3.11, the designed framework is comprised of five primary components. Monitoring, ranking, optimization, placement, and knowledge management are the stages that this process goes through. With regard to the analysis and comprehension of the system, each component carries a greater significance, and it also contributes to the process of making the appropriate decision on the execution of certain actions on the system in order to accomplish the intended objective. The purpose of this research is to ensure that all of the cloud resources that are being considered in a multi-cloud environment are optimized in such a way that the system's performance is both high and steady.

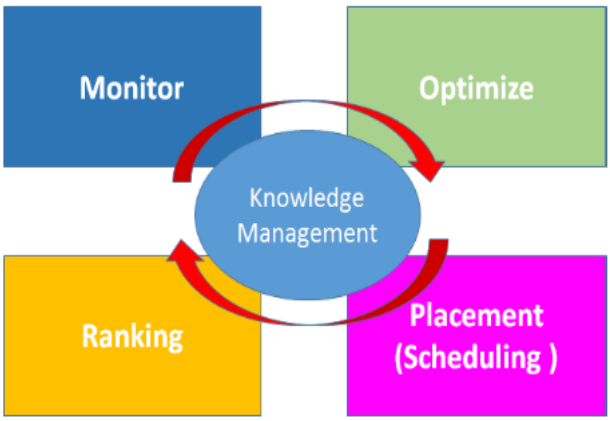


Figure 3. 11 Proposed Multi-cloud management framework

Monitoring

Monitoring the resources available in a multi-cloud system is an essential task that must be carried out in order to ensure efficient cloud management. Monitoring a variety of resource factors, including as virtual machines (VMs), containers, networks, storage, and memory, for the purpose of deploying applications in an effective manner and achieving to understand and improve the system's performance, it's important to get the best feasible result for a given situation. To establish an ideal system, it is necessary to continuously monitor these critical aspects, analyze these parameters, evaluate the metrics, and then, based on this, make a corrective decision about migrating to the right cloud environment to increase throughput. In a cloud context, containers optimized for Docker are placed within virtual machines (VMs). Keeping tabs on how well each cloud in a multi-cloud setup is supporting the virtual machines (VMs) and containers running within them is critical. This occurs because the deployment procedure involves the deployment of many clouds. In this example, the orchestration engine is kubernetes, and its job is to oversee the Docker containers that are meant to run on the VM using the Docker engine. Monitoring the performance of containers and virtual machines (VMs) would be much easier with the ability to track these system metrics. Based on the measured metrics, corrective measures might then be done. In order to keep tabs on containers, virtual machines (VMs), and other attributes, monitoring technologies like dynatrace and Prometheus are used. Information about these factors is stored in a database, and further analysis uses the generated metrics to deduce more details make decisions and actions that are corrective, and so on.

Analyzed key metrics include

• CPU Level (Host VM) – system resources used, memory utilised, storage space used by the host, memory

• Container level – Processor, memory, swap, throttle, and traffic between containers.

It is possible to define these metrics in a file, and then use that file to regularly get data from the cloud environment within a given time period. This will allow us to do further research. The data collected from the several clouds is compiled in a database for further study.

Optimization

In terms of cloud optimization, several researchers have found the system to be stable, and they have also looked at the system's present state generally. A lot of study has focused on reinforcement learning (RL) recently since it's a way to make decisions that doesn't need previous knowledge but can still identify a goal, run a functional model, and apply policy. Reinforcement learning (RL) is based on a system of penalties and rewards. Everything in this system is constantly geared toward the activities that provide the most profit. RL is a method of learning that, in a constantly changing environment, use trial-and-error interactions to generate the best possible action plans for a given state of the environment. Reinforcement learning describes this procedure. Through the use of an agent and optimization models based on RL and the Markov decision process, this work aims to comprehend the system's unpredictability. Instead of considering every possible state of the system when making a prediction, the agent uses its state action pair to learn the system's behavior and then uses the current state to create its prediction. As the designer may depict situations where the outcomes are partly dictated by chance and partially within the control of the decision maker, the Markov decision process finds application in several decision-making difficulties. Decisions made with a Markov model include

• State (S): This collection encompasses every conceivable state.

• Action (A): It stands for a series of potential outcomes starting from state "S",

• Probability of transition P(s'|s, a): The likelihood of changing states is shown by it "s" to "s' " at time "t+1" when a representative acted “a” in state “s” at time “t”,

• Reward R(s'|s, a): It stands for the benefit that the agent receives in the event that state "s" changes to state "s'" as a result of action "a."

• Discount (Y): It stands for the discount factor in situations when the current value of a reward is lower than its future value.

Many other kinds of Markov models are available, and the standard definition of a Markov process is a set of states represented by the letter "S," with certain actions denoted by the letter "A" being conceivable from each state with a certain probability

denoted by the letter "P." Within the context of solving (MDP) is a common application of RL. One formal technique to describe a reinforcement learning environment is using Markov decision processes. The surroundings is completely visible in this setting, which means that the current state entirely characterizes the process.

In this study, an optimization technique that makes use of MDP is developed in order to learn how the system is acting right now and see how it could behave in the future. The second goal is to keep an eye on the cloud in a multi-cloud setup by measuring it and rating it constantly. To achieve the desired throughput, the workload will be allocated according to the workload using scheduling, and deployment will be carried out as quickly as feasible. Furthermore, in a multi-cloud setup, the optimization challenge is divided into two parts: (a) optimizing at the virtual machine and cloud levels, and (b) optimizing across all clouds to fulfill performance, cost, and network latency criteria, and throughput.

Ranking

As part of the framework, the ranking process helps choose the best cloud to deploy according to the criteria. Furthermore, it helps in arranging the selected cloud's (VMs) in a ranking for future optimization and study. The objective of this study is to rank the cloud globally using the (AHP) approach and to classify (VMs) using the support vector machine (SVM). With the help of the ranking procedure, one may methodically sift through all of the potential targets and arrive at the best decision. Beyond this, it is more environment- and conditionally-oriented, which implies the ranking parameter will be quite essential. Another crucial part of our research is the data collected to filter and rank the metrics with the most importance. Researchers have invested a lot of time and energy into creating a large number of ranking algorithms for various settings, and there are a lot of statistical and machine learning methods that are accessible. Because it allows us to change the criterion for improved performance and outcomes and because it considers the decision criteria and priority, the AHP was chosen for this study. The second benefit of support vector machines is their improved classification accuracy and results, which are a consequence of their operation on non-linear decision making in multidimensional space. Using AHP, businesses may choose the best cloud for their workload deployment and migration

needs in a multi-cloud context. This tool helps with both new application deployment and existing application migration. We use a combination of (SVMs) and (PSO) to find the best set of (VMs), rank them, and then organize workloads into those VMs so they run most efficiently. A dissection of these algorithms and the framework that employs them is presented.

Placement of workload

To make sure there's enough room in the chosen virtual machines (VMs) for the new workload to be deployed, optimization is done after ranking on both global and local characteristics is successful in choosing the right cloud and the set of VMs. To maximize space and capacity utilization, a mechanism must be in place that permits the deployment of suitable workloads (tasks) on suitable virtual machines (VMs). Results from (SVMs) informed the scheduling and placement algorithms that locate VMs according to available free space and assign jobs according to size and priority. The goal of developing these algorithms was to find the optimal combination of (VMs) to run the workload in terms of both cost and runtime. In addition to allocating resources for newly arriving tasks, we want to transfer workloads from one set of (VMs) to another, or perhaps to another cloud entirely, based on the given rating. Additionally, workloads are delivered as containers, which make it easy to transfer containers to other clouds or (VMs).

Knowledge management

The most notable feature of this framework when compared to similar ones is its use of prior results and information derived from the system's present state via the Q-Learning and MDP process. In order to make accurate and timely forecasts and judgments, it is important to have knowledge about the system's present and historical conditions. Two distinct knowledge management tiers are included into this architecture.

• Predictive analytics methods, such as Long Short-Term Memory (LSTM) neural networks, are essential for managing both current and future workloads, and one such method is the capacity prediction of virtual machines.

• Decision making of shifting the containers from one VM to another inside cloud or across clouds utilizing MDP-RL approaches based on policy based judgments to have an optimum system performance.

Within this architecture, the anticipated number of virtual machines (VMs) is determined by averaging the daily workload over different time intervals. To predict the needed capacity of virtual machines, the Neural Network - LSTM method is used. The amount of virtual machines (VMs) needed for deployment on that specific day may be more easily planned. For predicting the needs in the near future from the outcomes of the past, the general statistical approach called ARIMA (Auto Regressive Moving Average) is helpful. On the other hand, this method ignores contextual information in favor of the premise that all employment will come at regular intervals. In order to increase prediction accuracy, this LSTM neural network considers contextual information, forgets some prior knowledge, and allows for the addition of additional information. When building the new capacity to automate the provisioning and de-provisioning of resources according to the work schedule, it is helpful to use information on the system's past and current states in this way.

Workload migration based on MDP

Re-enforcement learning, Markov decision process, and Q-Learning were we covered everything in depth in the part before this one. The present state of the system is inherently random, but these learning approaches help us comprehend it. On top of that, they help figure out what to do next based on this knowledge so that you may maximize your return. In this part, it draws out a state-action map and finds the best policy that will lead to the most benefits in reaching the goal. Applying Q-Learning allowed the system to learn and was able to anticipate the next probable action. Following the construction of the Q-Table, which includes the maximum rewards, it is utilized for the purpose of forecasting the subsequent action that can be carried out, taking into account the current condition of the system. Using the data presented by this Q-Table, the workload migration—also called the container migration—is executed in this instance. This transfer might happen in the cloud or between clouds. The local optimization method may be used to optimize the workload distribution by moving virtual machines (VMs) into or out of containers (to better distribute the

burden). Utilizing knowledge management gained via learning is very helpful for predicting the next likely action based on the current state of the system.

Data volume processing using R, sparks, Scala and Kafka

One of the trickiest parts of this task is organizing all the monitoring data for analysis, prediction, and optimization. One of the biggest problems with this study is working out how to handle, process, and infer information from such a huge amount of data quickly enough, given that data is collected from four different clouds in a multi-cloud scenario, and that data is collected periodically over a longer period of time. This data includes things like collecting multidimensional metrics, assigning different values to virtual machines and containers running inside VMs across different regions on four different clouds, and so on. For the sake of making judgments that are relevant and helpful in solving the situation at hand, this is done. In order to allow for the performance of several actions simultaneously, it is also important to run these algorithms in parallel. So, for maximum efficiency, the designed system has to handle massive amounts of data and provide more cross-system cooperation.

3.2.4 KEY DIFFERENTIATOR OF THE PROPOSED FRAMEWORK

There are a lot of benefits to using the suggested framework instead of existing alternatives. Here are the main points that set this method apart from others:

a) Establishing a structure for interconnecting clouds for end-to-end multi-cloud management and providing a unified interface for this purpose.

b) Gain insight into the Cloud's present status using Markov decision process (MDP) and make near-term predictions using reinforcement learning (RL). As an added bonus, you can utilize the system's historical behavior to forecast its future actions.

c) Define the procedure for managing and deploying workloads, and use a ranking algorithm for both global and local optimization utilizing an AHP (Arithmetic Hierarchical Processing) and SVM (Support Vector Machine) method.

d) Implement parallel processing of algorithms and data to expedite data processing and decision-making outcomes; make use of modern technologies such as R, scala, spark, and kafka for efficient handling of massive data sets.

e) Applying DL techniques to recurrent neural networks allows algorithms like RNN-LSTM to mimic human prediction and classification abilities.

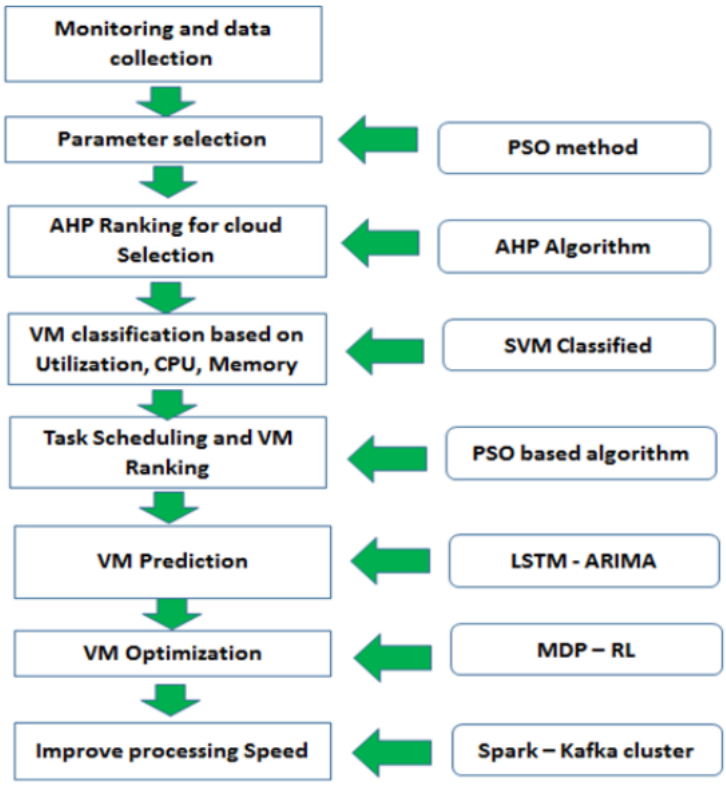


Figure 3. 12 Multi-cloud management process

Explanations of the various steps of the framework are provided in Figure 3.12, the algorithms used for analysis and decision making at each level, and a thorough depiction of them.

Here we cover why a solid foundation is necessary for managing multi-cloud operations and why many businesses are beginning to realize this. Research toward a framework for managing operations in multi-cloud environments is now underway. Machine learning, optimization, and scheduling are new approaches that are crucial for managing operations across several clouds. Furthermore, new technologies like as containers, micro-services, orchestration, etc., are assisting with the resolution of some issues, such as portability, interoperability, service discovery, etc. What makes this study stand out is that no prior system has ever used a ranking technique to choose the best cloud and, within that, the best resources for efficient deployment. Furthermore, efficient multi-cloud management makes advantage of the insights obtained from earlier analyses as well as the technique to boost computing speed. The

overarching goal of this study is to provide a solid framework that includes comprehensive methodology, methodologies, and an approach that has been tested in real-world settings and shown to be successful in managing operations in multi-cloud environments.

CHAPTER 4
MONITORING, PARAMETER RANKING, AND OPTIMIZATION FOR PREDICTIVE KNOWLEDGE MANAGEMENT

4.1 MONITORING AND PARAMETER SELECTION

The initial stage of analysis in this study is the monitoring of cloud resources and the measurement of metrics, which is based on the framework that has been proposed. Monitoring a variety of resource factors, such as virtual machines (VMs), containers, networks, storage, and memory, in order to deploy applications in an effective manner and get the best possible outcome for a certain circumstance is the goal. Constant vigilance over these critical parameters will lead to the most efficiently run system, the analysis of these parameters, the measurement of the metrics, and the decision-making process based on this information on migration to the appropriate cloud environment for improved throughput capabilities. Each and every cloud in a multi-cloud system has its own collection of resources, components, and services, all of which are managed and controlled by the cloud service providers (CSP) that are responsible for that cloud. When it comes to managing their own cloud resources, each cloud service provider (CSP) has their own monitoring dashboard and control parameters. On the other hand, when it is deemed to be a multi-cloud environment, which means when the clouds are connected to one another, there must be a centralized monitoring system that is capable of monitoring individual cloud resources and has a common controlling and decision-making mechanism. Having a system and rules in place is critical for managing operational problems collectively and fixing them. Doing so is essential for resolving the operational issues. Centralized monitoring systems that can keep tabs on all the participating clouds and regularly collect their parameters is the ultimate goal of this work in progress, and perform additional analysis for the purpose of continuous optimization.

4.1.1 MONITORING PROCESS AND DATA COLLECTION

One of the most critical things that must be done for this research endeavor is the regular gathering of data and the monitoring of cloud resources. Within a setting with several clouds, each of the participating clouds has a unique collection of resources

and is in a different condition and state. As a result, it is necessary to perform continuous monitoring and analysis of data, as well as to measure key metrics and take corrective actions based on the decisions that are reached based on these metrics. As shown in the following picture, this study involves the interconnection of four participant clouds, including Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and Open Stack, with an overlay network and a centralized monitoring tool known as Prometheus. As a result of the fact that Prometheus is an agent-less monitoring tool, it is simple to configure this tool with any cloud, and it is also simple to monitor resources with any cloud that is participating. The process of storing the data that has been acquired for subsequent analysis is described in Figure 4.1, which also provides an explanation of how the monitoring and data collecting from the multi-cloud environment is functioning.

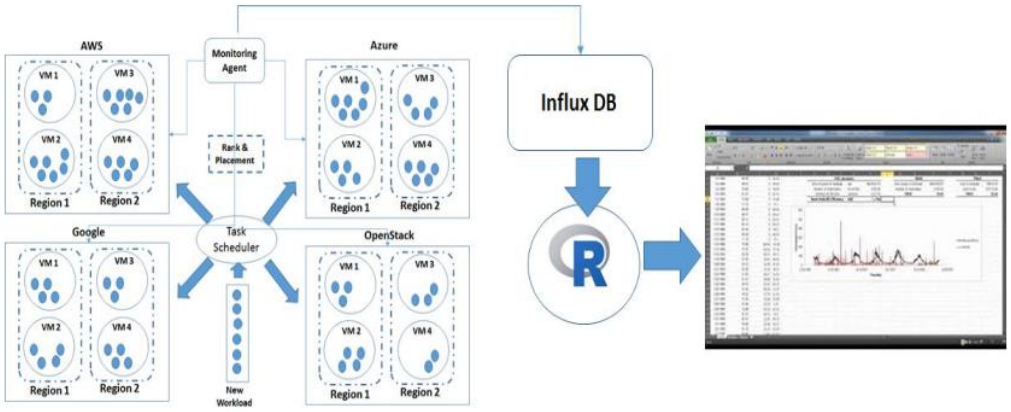


Figure 4. 1 Process of Multi-cloud monitoring

Within the scope of this investigation, the following information is gathered at every level of the resources, such as the virtual machine level and the container level, at a rate of ten minutes per hour for a period of three hours.

This information is then compiled in the inflow database from each cloud and the application that is operating within each cloud. For the purpose of ranking, positioning, and optimization, there are approximately twenty different parameters that are first taken into consideration for study. Details on the twenty characteristics that were taken into consideration from each cloud are provided in Table 4.1.

These parameters were collected at the virtual machine and container level for further analysis.

TABLE 4. 1 Parameters of Virtual machine and Docker container

Docker Parameters	VM Parameters
Total Containers Running	VM CPU
Container CPU	VM Memory
Container Memory	VM Disk space Utilization
Container Memory Usage	Disk I/O
Container Swap	Network Latency
Container Disk I/O	Number of cores
Container Network Metrics	VM Power
Application Metrics	Disk Swap
Inter-Container traffic	Energy / Temperature

With the monitoring tool, you have the ability to configure these parameters. Following the configuration of the Prometheus tool with cloud resources, the collection of data is carried out automatically, and it is then pooled in the influx database for purposes of future analysis. Due to the fact that there are twenty parameters, the dimensionality of this data is high, and there are also four clouds that are participating, the volume of data that has been collected over a very short period of time is enormous, and it has to be managed and examined. In the following section, it is detailed how the data that have been collected are further processed in order to reduce the dimensionality and, as a result, increase the speed and accuracy of analysis and decision making.

4.1.2 DATA ANALYSIS AND PARAMETER SELECTION

It is necessary to have a defined framework in order to automate the process of data collection and analysis because it is an ongoing process. Additionally, it is necessary to have an integrated system in place. The following picture provides an explanation of the process of data gathering, which is then combined with an algorithm for

machine learning and a decision-making system in order to facilitate efficient data management. During this step of the analysis, In order to streamline and improve the decision-making process, the framework accounts for parameter selection according to their weightage and importance in the overall decision-making procedure. Particle Swarm Optimization (PSO) is the technique used in this work, which will be described in more detail in the section that follows.

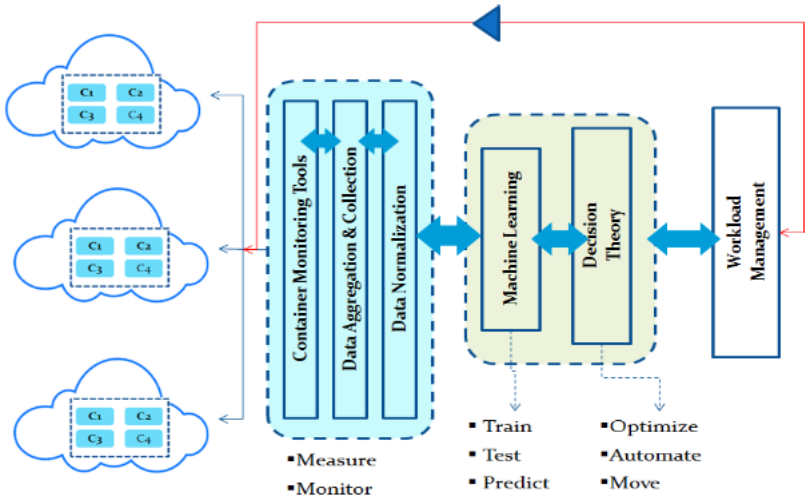


Figure 4. 2 Procedure of Data collection

Figure 4.2 lays out the steps to take in order to gather and handle data from a cloud environment. Here, we evaluate the categorization's efficacy using the support vector machine (SVM), a machine learning classification method. The test's findings establish the criticality and relevance of the participating vectors in the twenty aggregated parameters, paving the way for more research. The objective, as previously stated, is to minimize the data's dimensionality and volume by selecting the best combination of parameters that improves classification accuracy while simultaneously reducing the number of parameters from twenty to eight.

4.1.3 PARAMETER SELECTION USING PSO-SVM ALGORITHM

Parameter selection (sometimes called feature selection) is a process that is seen as one of the optimization challenges in machine learning. The purpose of parameter selection is to shorten calculation times without sacrificing classification accuracy by reducing the number of features and removing irrelevant, noisy, or duplicated data. Data mining, pattern classification, machine learning, and medical data processing are

some of the main fields that use the challenge of feature selection from multidimensional data. This problem is generally employed in the field of pattern classification. Therefore, it is necessary to have a good strategy for selecting features, which will assist in enhancing the processing rate, the accuracy of prediction, and the avoidance of incomprehensibility, while simultaneously enabling classification accuracy. When it comes to optimization, there are numerous approaches that have been tried out by numerous study scientists in the past, including parameter selection and dimensionality reduction. Findings indicate that, among the currently available algorithms, Particle Swarm Optimization yields the best results since it employs a multi-objective decision-making procedure. Here, we use particle swarm optimization (PSO), an optimization method, to solve the feature selection issue. On top of that, SVMs are used in tandem with the one-versus-rest approach, which is a fitness function of PSO for the classification issue, and PSO itself.

PSO Overview

When it comes to managing the data volume from numerous cloud resources, Parameter selection and dimensionality reduction were two of the most critical considerations in this study. The process of selecting parameters is based on a great deal of prior study. Following extensive evaluation of most techniques, it was determined that the most effective classification method was a hybrid of Particle Swarm Optimization (PSO) and Support Vector Machine (SVM). Therefore, as part of the current research, this technique is being considered as a possible alternative to handle the parameter selection process.

One stochastic optimization method that was inspired by swarm behavior is Particle Swarm Optimization (PSO). When birds forage for food in large flocks, scientists study their behavior. Both its speed and the direction it looks for food are dictated by the knowledge it has previously received and its present position in the search space. Any collection of swarming organisms, such as a flock, is regarded as a particle. Every single one of the flock's individual particles has a possible solution to the problem whose resolution is being considered. Pbest stands for "personal best," and it refers to the particle's favorite location that has yielded the best possible outcome, as defined by the classification system in use. The local best, often abbreviated as lbest, is the location of the strongest particle member in the immediate vicinity of a given

particle. The global best, or gbest for short, is the best particle's location across all swarms. For a particle to improve its current position, the velocity is the vector that indicates the direction in which the particle must fly (move), and when one particle is used to direct another particle toward more favorable portions of the search space, that particle is called the leader. In order to regulate the effect of a particle's past velocities on its present velocities, the inertia weight, represented by the symbol W, is used. Cognitive learning (C1) and social learning (C2) are two components of learning. One way to think about a particle's learning factor is as an attraction toward its own or its neighbors' success. Common practice dictates that C1 and C2 are synonymous constants. In conclusion, the set of particles used to determine the optimal value of a given particle is decided by the neighbourhood topology.

Support Vector Machine

For problems involving classification and regression, Support Vector Machine (SVM) functions in a multidimensional space, making it one of the successful machine learning challenges.

There are few machine learning jobs as common as this one. In order to solve a broad range of real-world problems using a small number of kernel functions, it may be characterized as both linear and non-linear. A line or hyperplane is supposed to be generated by the model in support vector machines (SVM) to split the data into its relevant classes.

The supervised machine learning method known as a support vector machine (SVM) has several applications, including classification and regression analysis. Based on the idea of decision planes, which define decision limits, it finds principal use in categorization problems.

A decision plane is a type of plane that differentiates between a group of objects that belong to various classes and makes use of the theory of machine learning to achieve the highest possible level of forecast accuracy while simultaneously avoiding over-fitting to the data being used. Utilizing a non-linear separable model, which is the most popular strategy in classification, is something that can be done in a vector space that has multiple dimensions.

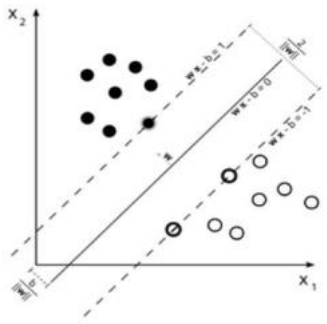


Figure 4. 3 SVM classification with hyper plane

Using a hyper plane, Figure 4.3 shows the results of using the support vector machine (SVM) to categorize two distinct groups. When it comes to classification and prediction, the support vector machine (SVM) is the model of choice. As mentioned earlier, it accomplishes its goal by reshaping the data into a different space, which enables the data to be partitioned using a linear or non-linear function. Additionally, the modified space has the largest distance between its vectors and the smallest error between the various types of data points represented in multidimensional space. Support vector machines are used when a non-linear kernel function is required and the decision boundary cannot be isolated from the support vectors. For the goal of handling non-linearly separable data, this inquiry makes use of a specific class of functions called the kernel function. This is seen in Figure 4.4, which explains how classes in a non-linear model are thematically separated.

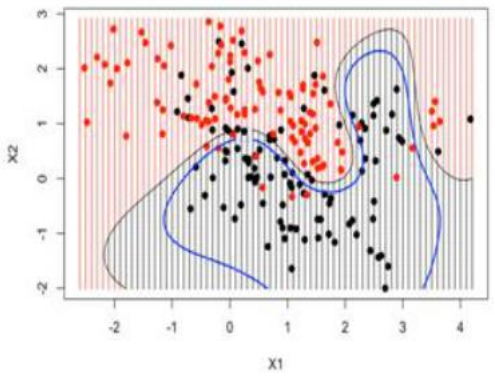


Figure 4. 4 SVM classification with non-linear kernel

Kernel functions are functions that take two mapped feature vectors and return the dot product of those vectors. The initial feature vectors are sent to these functions, which are referred to as kernel functions. A liner model for classification, it reduces the

computing burden of deriving the higher dimensional vector space from the basic vector space that is provided. Because of this, information may be linearly segregated in three-dimensional space. The kernel function is often used since it is a linear model.

PSO with SVM

When doing particle swarm optimization (PSO), each possible solution is like a bird in a flock; each one is considered a particle in the search space. In Particle Swarm Optimization (PSO), a fitness function is optimized based on the fitness values of individual particles.

Furthermore, the velocities of the particles govern their motion. As the particles wander around the search area, they constantly adjust their positions according to what they've learnt from their own experiences and the experiences of the particles around them. This allows the particle to learn and update the knowledge it has gained from the experience it has gained.

As the search progresses, every particle remembers the best possible location that it and its neighbour can discover. After that, it traverses the problem space by tracing a collection of existing optimal particles.

Here we will go over the PSO-SVM method in depth:

1. Set the starting population size to "n" when you launch PSO.

2. In the problem space, specify the number of particles and their dimensions.

3. Establish and maintain the particle-specific learning rates.

4. Determine each particle's fitness value by factoring in the training set's cross validation error.

5. By comparing the fitness values, determine the particle's local best and global best.

6. Revise the velocities, positions, and inertia masses of all particles.

7. When the fitness function converges or the entire iteration count is achieved, repeat steps 4-6.

8. The SVM classifier is fed the set of particles that were determined to be the best globally when condition 7 is satisfied..

Pseudo Algorithm

Initial Population to "n"

While (Not equal to number of iteration or stopping criteria)

For p=1 to number of particles

If X_p> fitness of $pbest_p$

Then update $pbest_p$ to X_p

For k ∈neighborhood of X_p

If the fitness of $X_p > gbest$ then

Update $gbest = X_k$

$Next_k$

For each dimension d

$$v_{dp}^{Curr} = \mathrm{w} \times v_{dp}^{Curr} + C_1 \times rand_1 \times (pbest_{dp} - X_{dp}^{Prev}) + C_2 \times rand_2 \times (gbest_{dp} - x_{dp}^{Prev})$$

If $v_d \neq (V_{\max,} V_{\min,})$ then

$v_{dp} = max\ (min\ (V_{\max,} V_{dp}), V_{min})$

$x_{dp} = x_{dp} + v_{dp}$

$Next_p$

$Next_d$

Next generation until stopping criteria.

The feature dimension's particle characteristics are based on the monitoring parameter values from Table 4.1, which includes 18 parameters total (nine for containers and nine for virtual machines). An SVM, or support vector machine, was used to classify this data in order to improve the accuracy of the classification. In order to find out how well the PSO fitness function solves the parameter selection problem, the support vector machine (SVM) is mainly used as an assessor. Using particle swarm optimization with a fitness function, one may pick the ideal collection of features. This can be shown with an 18-dimensional data set (n=18) and a "m" (say, 50,000) set of samples collected from a monitoring system, where Sn = F1, F2, F3, (...)F18. In this case, we'll use: The purpose is to choose a small number of features (less than n)

from the set of 18 features that may be further analyzed. The set of features is represented by Sn = F1, F3, F5, F7, F9, F10, F13, F15, with 8 subsets of samples from the features. We use the K-Fold Cross-Validation Method to partition the "m" data set into 10 separate pieces, called as D_1, D_2, D_3, … D_n}. For example, a dataset consisting of 5000 data points is divided into ten parts, and each part is subjected to a ten-time cycle of training and testing with the other nine components. 10 classification accuracies are produced after 10 training and testing cycles. The total classification accuracy of the dataset is calculated by averaging these 10 classification accuracies. This method relies on binary PSO algorithms the entire time.

Each particle's new value is based on the fitness function's calculated value, which is called the adaptive value. The largest adaptive value inside a group of pbest is referred to as gbest, where pbest stands for the best adaptive value for each particle renewal. Following data collection for pbest and gbest, one may track the properties of these particles in relation to their velocity and location. The feature selection procedure is made easier by representing each particle's location in the search space as a binary string. Here are the equations that illustrate how the adjustments affect the velocities of the individual particles.

$$v_{dp}^{Curr} = \mathrm{w} \times v_{dp}^{Prev} + c_1 \times rand_1 \times (pbest_{dp} - X_{dp}^{Prev}) + c_2 \times rand_2 \times (gbest_d - x_{dp}^{Prev}) \quad (4.1)$$

$$S (v_{dp}^{Curr}) = \frac{1}{1+e^{-v_{dp}^{Curr}}} \quad (4.2)$$

If $\left(\mathrm{rand} < \mathrm{S}\left(v_{dp}^{Curr}\right)\right) then\ x_{dp}^{Curr} = 1; else\ x_{dp}^{Curr} = 0.$

The function S (v_{dp}^{Curr}) (Eq. 4.2) employed to ascertain the characteristic subsequent to renewal. The speed value is v_{dp}^{Curr} in this equation. If the feature S (v_{dp}^{Curr})is greater than a disorder number that is generated randomly and falls within the range of (0, 1), then the position value Fn, where n is equal to 1, 2,..., m, the feature is marked as {1}, indicating that it is selected as a required feature for the next renewal process. If S (v_{dp}^{Curr}) If the value of the position is smaller than a disorder number that is generated at random and comes within the range of {0~1}, then F_n $where$ n = 1, 2, …, m is

represented as {0}, which indicates that this feature is not chosen as a needed feature for the subsequent renewal.

4.1.4 RESULTS AND DISCUSSIONS

The monitoring and data collection lab environment is comprised of four interconnected clouds. A few examples of these clouds include Open Stack, Google Cloud Platform, Microsoft Azure, and Amazon Web Services (AWS). With Prometheus, a centralized monitoring tool, you can keep tabs on cloud metrics across all of your cloud installations. Every cloud that is being examined for data collecting has agents installed and this tool is set up. Each of the three cloud providers—AWS, Azure, and Google—has twenty virtual machines (VMs) set up specifically for the research. Two distinct regions, like the US and Europe, host these virtual machines. Twenty virtual machines (VMs) are available in our lab for monitoring reasons; these VMs are linked to the Open Stack private cloud via public clouds. Also, all cloud environments generate Docker containers, which are then deployed on candidate virtual machines distributed around the cloud for optimization and monitoring purposes. In Figure 4.5, we can see a flow diagram of the data gathering and storage process from containers.

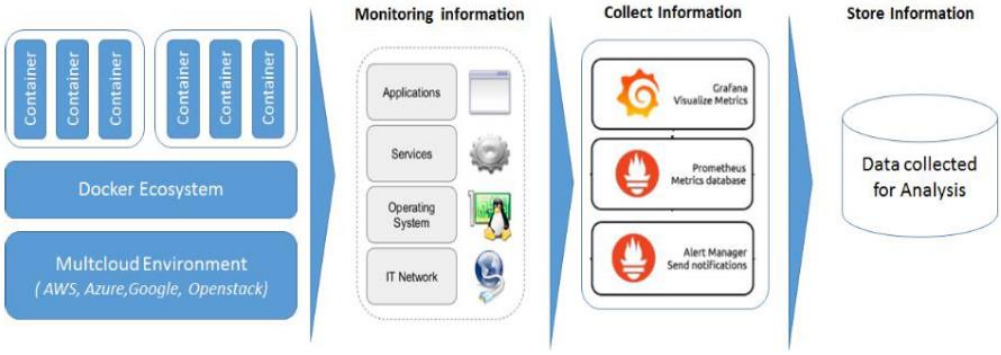


Figure 4. 5 Data monitoring and storage mechanism

By using this API, Prometheus will be able to scrape metrics at regular intervals, such as every set of seconds. It is possible to export these metrics to a Prometheus-connected influx database. The initial set of settings for containers and virtual machines is as follows. Twenty virtual machines and containers in each of the four cloud environments routinely provide about eighteen parameters to the cloud data set. To facilitate their retrieval for further examination, these parameters are suitably

indexed. The virtual machine level collects nine of these characteristics, whereas the container level collects nine more. Using the 18 VM and Container parameters and the aforementioned technique, we improved the SVM classification accuracy by calculating the fitness function using the binary PSO algorithm using equation 4.1. The output was a binary string with the value of, as predicted by equation 4.1. S (v_{dp}^{Curr}), whereby it is represented as {1} if it exceeds the threshold and as {0} if it falls short. Using this, the fitness value obtained by binary SVM iteration according to pseudo code, and the final qualifying set of values are considered for the purpose of evaluating SVM classification accuracy.

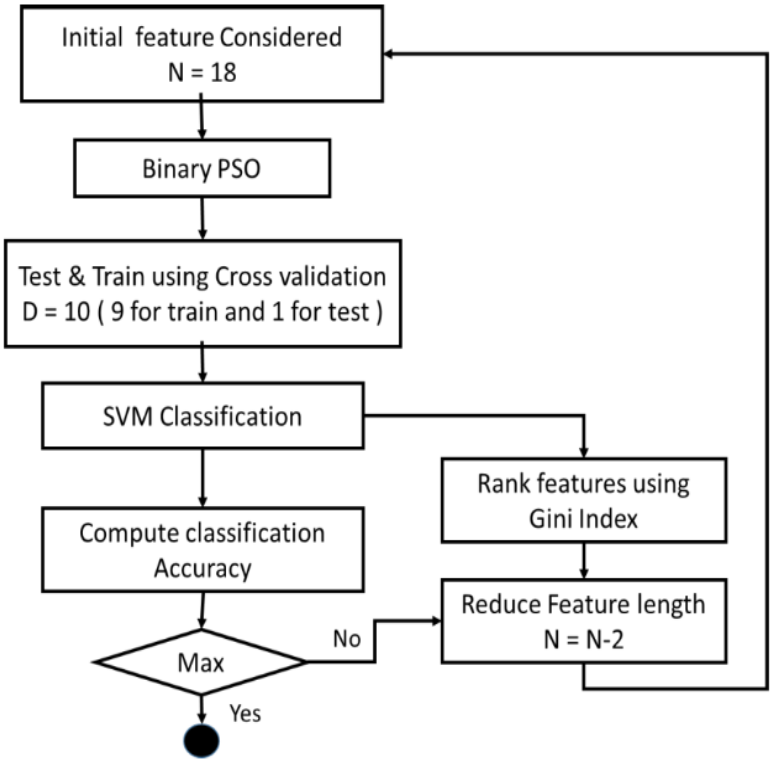


Figure 4. 6 Parameter selection Process using PSO-SVM

Figure 4.6 shows the whole process, from start to finish, of parameter selection using the PSO algorithm. According to the results, identifying a set of eight parameters improves the accuracy of classification using SVM, comprised of four for virtual machines (VM) and four for containers.

As a result, these parameters are utilized for further study. On the basis of this analysis, the parameters are chosen, and the Gini index is utilized for the purpose of feature ranking for the support vector machine.

Through the use of the Gini coefficient, one may determine the degree of inequality that exists within any distribution. No matter what, the value of the coefficient will always fall somewhere between 0 and 1, where 0 denotes complete equality and 1 denotes complete equality throughout.

At this point, the percentage value is obtained by multiplying it by 100. We sort it according to the Gini score and use SVM to choose the best values for categorization. About six distinct groups (clusters) make up the whole.

The reduction in dimensionality from 18 to 8 and an improvement in classification accuracy to 91.9% are both achieved by feature selection using the combination of the PSO-SVM approach, as shown in the summary of table 4.2.

As seen in Table 4.2, the SVM classification accuracy is going up even while the number of parameters examined by the PSO method has decreased.

TABLE 4. 2 Classification accuracy with reduced Parameters

Number of Parameters	SVM Accuracy
18	72.6
16	77.8
12	82.9
10	88.3
8	91.9

The dimensionality of the issue may be decreased and its accuracy improved using a feature selection strategy based on particle swarm optimization (PSO). Reducing the number of parameters from 18 to 8 seems to have been more effective and yields more meaningful findings. Careful parameter selection using Particle Swarm Optimization (PSO) allowed for this decrease.

According to the PSO selection approach, Table 4.3 shows the eight parameters that were ultimately considered for further study and investigation.

TABLE 4. 3 Shortlisted parameters using PSO using parameter selection method

Docker Parameters	VM Parameters
Container CPU	VM CPU
Container Memory	VM Memory
Container Disk I/O	Utilization
Container Network Metrics	Network Latency

It is shown that the PSO method is successful for selecting parameters from data with greater dimensions. Further, by reducing the complexity of parameter sets and optimizing the functions used for parameter selection, it aids in performance improvement. Concurrently, it increases classification accuracy across the board while decreasing the amount of training and testing required handling multidimensional data via the use of a kernel function that finds a maximal margin hyper-plane in a high feature space.

Using the PSO technique, this section outlined the steps for determining the optimal set of parameters from a bigger dataset. The goal of minimizing the number of parameters is to make processing faster and the results more accurate by reducing them to a manageable set. It makes the data easier to work with by decreasing its size. Through this investigation and testing, the efficacy of the chosen parameters and their impact on the final outcome are shown. Particle swarm optimization (PSO) uses prior information and easily converges to the best solution, hence it produces better results than other techniques to parameter selection. In the final stages of this study, our method improved classification accuracy by helping to pick the optimal subset of parameters that satisfied the objective functions.

4.2 RANKING AND PLACEMENT TECHNIQUES

When working in an environment that utilizes many clouds, it is always tough and challenging to discover the appropriate cloud for deploying new tasks or workloads. Therefore, it is necessary to have a mechanism that can locate the appropriate cloud. One such way of discovering is referred to as the "Ranking process," and it is responsible for determining the appropriate cloud platform by taking into account a

set of criteria and forming a model that assists in ranking the cloud. In general, the ranking process involves the collection of a certain set of parameters for the purpose of analysis. The priority and relevance of these features are to be provided as input, and the resultant factor is arrived at based on particular weight factors. This factor is then used to assist in ranking the entity. According to the findings of various studies, there are various levels of ranking, such as ranking the cloud, ranking the virtual machines (VMs), and ranking the job. These levels are taken into consideration for analysis, and as a result, numerous distinct ways have developed over the course of time. For the purpose of gaining a knowledge of how ranking is accomplished through the use of statistical methodology, the following section will provide a detailed discussion of a few of these algorithms.

4.2.1 RANKING ALGORITHMS

AHP – Analytic Hierarchy processing

The Analytic Hierarchy Process, also known as AHP, is a multi-criteria decision-making approach that is utilized for the purpose of making decisions based on combined comparisons. Using AHP, the decision maker is able to reduce complex decisions to a series of pair-wise comparisons, and then synthesize the outcomes of those comparisons. This allows the decision maker to set priorities and make the best decision possible. It is the goal of the AHP to capture both the subjective and objective parts of a decision, as well as to enable a strategy for testing the consistency of the evaluations made by the decision maker, which would ultimately result in a reduction of bias in the process of decision making. The principle Eigen vectors are used to derive the ratio scales, and the principal Eigen value is used to derive the consistency index. Both of these are derived from the principal Eigen. In order to arrive at the most appropriate choices, the AHP algorithm is engineered with a collection of evaluation criteria, and it also employs a collection of alternative options. A weight is assigned to each evaluation criterion by the AHP algorithm for each pair-wise comparison that the decision maker does for each of the criteria that are taken into consideration. The significance of the criteria is determined by the weights that are assigned to them. In the event that the weights are high, the criteria will be of greater significance, and vice versa. A pair-wise comparison is done between specified parameters in the form of a matrix, during which the eigen values

and eigen vectors are computed. This is accomplished by selecting the set of decision criteria, assigning priorities to each of the criteria, and then performing the comparison. As a consequence of these values, the ranking parameters have been generated, which assists us in making decisions that are appropriate.

TOPSIS

The Technique for Order of Preference by Similarity to Ideal Solution, also known as TOPSIS, is a method of decision analysis that takes into account multiple criteria and was developed using the idea of geometric distance. With the longest value from the negative ideal solution (NIS) and the shortest value from the positive ideal solution (PIS), the geometric distance is computed for the solution that was chosen. This method compares a set of alternatives for each criterion by establishing weights through the use of compensatory aggregation. This method can assist in satisfying the objective and systematic evaluation of alternatives on a variety of criteria that are being examined for analysis. By using these strategies, it is possible to achieve a poor performance in one criterion, which can be compensated for by achieving a good result in another criterion, and to achieve a compromise between many criteria. The TOPSIS technique operates under the assumption that every criterion has preferences that grow in a monotonically increasing or decreasing fashion. As a result of the particular advantage it possesses over criterion modelling and compensating approaches, this method is utilized extensively in a variety of fields that involve making decisions based on several criteria.

Bipartite

With the assistance of learning, bipartite ranking is able to derive the ranking function from a training set that contains examples that have been classified both positively and negatively. When a ranking function is applied to a collection of unlabelled instances, it is anticipated that it will establish a total order in which positive instances come before negative ones. In this algorithm, the system assigns a real number to each instance of the scorer in such a way that positive instances have a higher score than negative instances, and the objective is to make the scorer learn continuously. It is anticipated that the scorer will be subject to a penalty of "1" every time he violates the condition. Applications of bipartite ranking include content recommendation, in

which the objective is to rank a set of items based on an individual's liking for them, and epidemiological studies, in which the objective is to rate a group of persons who are very likely to have a certain disease. Both of these applications are examples of bipartite ranking.

Rank SVM

For the purpose of learning classification, regression, or ranking functions, support vector machines (SVMs) are classified as classifying SVMs, support vector regression (SVR), or ranking SVMs (or Rank SVMs), respectively. SVMs are frequently employed for learning these functions. During the past ten years, support vector machines (SVM) have been utilized widely in the fields of data mining and machine learning communities, and they have also been actively applied to applications in a variety of other disciplines. The two special qualities of support vector machines (SVMs) that are commonly regarded for ranking processes are (i) the ability to maximize the margin while maintaining a high level of generalization, and (ii) the ability to allow efficient learning through the use of nonlinear functions such as kernel trick. Ranking support vector machines (SVM) through the approach of learning preferences (ranking) has resulted in the production of a variety of applications in the field of information retrieval. Additionally, it has improved the performance of generalization by increasing the contrast between the minimal rankings. It is responsible for computing the weight vector "w" that optimizes the differences between data pairs that are ranked closely together. SVMs that are used for ranking find a ranking function that has a high degree of generalization in this manner, which makes the ranking more successful in comparison to other algorithms that are similar.

Limitation of ranking algorithms in multi-cloud management

There were a few well-known ranking algorithms that were briefly covered in the part before this one. These algorithms were used to analyse multi-objective functions. When it comes to parameter selection, prioritizing, pairwise analysis, and weightage, as was mentioned before, every algorithm has its own set of advantages and disadvantages and does not share any similarities. There have been a few modified versions of the algorithms that were explained earlier, which have increased the decision accuracy even further by taking into consideration fuzzy parameters. In the

same vein, every single one of them presents a unique set of issues and is better suited for a specific situation and use case that is being examined for analysis and rating. The purpose of this study is to conduct an analysis of the multi-cloud environment. The researchers wanted to take into account various criteria and priorities when selecting the best cloud for a specific scenario within the multi-cloud environment. They discovered that AHP is more suitable for the situation, and as a result, they considered it as a potential option for the framework design that will be used for global ranking.

4.2.2 MULTI-CLOUD RANKING

As we go further with this methodology and this research, it is crucial to rank the clouds in a multi-cloud scenario in order to manage and decide which cloud to choose for the deployment of a new application that is scheduled to be deployed or for migrating any of the existing applications that are running in one cloud to be re-deployed on to any of the other clouds for optimization. It is the decision to select the appropriate cloud for the appropriate deployment, taking into consideration the relative importance of the various characteristics that are being evaluated. The AHP algorithm allows for this degree of flexibility in ranking, and it also allows for the random setting of priorities. AHP conducts additional analysis based on this priority and provides the ranking appropriately, which is detailed in greater detail in the section that follows.

Global ranking using AHP

We are able to make effective decisions on difficult situations by simplifying and accelerating natural decision-making processes, which is made possible by the Analytic Hierarchy Process (AHP), which provides a framework through which complex problems can be handled in a relatively basic way. Using both qualitative and quantitative approaches to decision making, it assists in the process of locating the best possible option. Instead of making hasty conclusions and attempting to defend them, the AHP aids in approaching the problem of decision making in an organized manner, which is in contrast to the traditional method. AHP is a framework that is designed to answer complicated problems that involve several criteria. It does this by combining the deductive method and the inductive approach into a single

integrated component of the framework. AHP is advantageous since it is designed to deal with circumstances in which the subjective judgments of individuals comprise a significant part of the decision-making process. This is one of the reasons why it is so important.

For the most part, the application of AHP is focused on the objective of accomplishing the overall goal, using the most essential choice possibilities in a pair-wise assessment as the basis for the business criteria and functions. It uses an underlying scale with values between 1 and 9 to find the relative preferences between two objects. The element represented by the letter i in the matrix indicates the degree to which the item in row i is favored when compared to the item in column j. In the pairwise comparison matrix, AHP assigns a value of 1 to anything on the diagonal. By applying the inverse of the transposed position, the AHP may ascertain the transposed position's preference rating. The number of entries actually filled out by decision makers is equal to $(n^2- n)/2$, where n is the number of things to be compared, according to the previously mentioned formula. The method, whose term is "synthetization," ranks the significance of each possible option as it pertains to the criteria that are being evaluated for analysis. Performing synthesis is accomplished through the application of a mathematical process known as computation of eigenvalues and eigenvectors. Table 4.4 illustrates the preference that was used for ranking, and it is possible for there to be alternative criteria depending on the problem definition that is being considered.

TABLE 4. 4 AHP Priority

Preferences	Ranking
Extremely Preferred	9
Very Strongly to extremely preferred	8
Very Strongly preferred	7
Strongly to very strongly preferred	6
Strongly preferred	5
Strongly to very moderately preferred	4
Moderately preferred	3
Equally to moderately preferred	2
Equally preferred	1

The following three-step procedure provides a good approximation of the synthesized priorities.

Step 1: Find the total values in each column in the pair-wise comparison matrix.

Step 2: Divide each element by its column total for the matrix which is alre4ady normalized,.

Step 3: Estimate the average of the elements for every row of the matrix which is normalized.

The averages that are being compared offer an estimation of the relative importance of the components that are being compared. The consistency of judgments that the decision maker displayed throughout the series of pair-wise comparisons is a significant factor in determining the quality of the decisions that their decision maker makes.

In the context of this study, one of the most important factors is the selection of the appropriate cloud from among several clouds for the purpose of analysis and deployment, with the goal of optimizing the utilization of existing resources and deploying new workloads.

For this reason, it is necessary to take into consideration the ranking of clouds (four) based on a set of critical factors, and it is also necessary to complete the business priority analysis in order to choose the appropriate cloud technology.

In this context, as was indicated previously, four distinct clouds—namely, Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and Open Stack—are taken into consideration for multi-cloud management and ranking process.

To determine which cloud is the most suitable for a given situation, business criteria are developed, and the ranking process is used to conduct an appropriate evaluation.

The arithmetic hierarchical process, often known as AHP, is adopted as the ranking method for this investigation.

Details regarding the five factors that are utilized by the AHP algorithm for the ranking process are provided in Table 4.5.

TABLE 4. 5 Criteria considered for Ranking process using AHP

	AWS	Azure	Google	Open Stack
Cost of VM (S) Per month	26.8	30.5	28.6	27.4
Average Network Latency	247	238	254	238
Average Availability (%)	99.5	99.8	97.5	96.3
Response Time (mSec)	245	276	298	283
Average Utilization %	76.4	71.7	67.3	76.3

Data Collection

Performing the AHP algorithm requires taking into consideration five business parameters for analysis.

These parameters are as follows: cost, network latency, availability for service level agreement (SLA), reaction time, and usage of virtual machines (VMs) in each cloud.

The most important factors that we use for our study and ranking are things like the average cost of virtual machines (VMs) per month for each cloud, the average network latency, the availability service level agreement (SLA) that we calculate for each cloud, the average usage of VMs over a month, and the response time of VMs in each cloud over a period of time.

Parameter selection

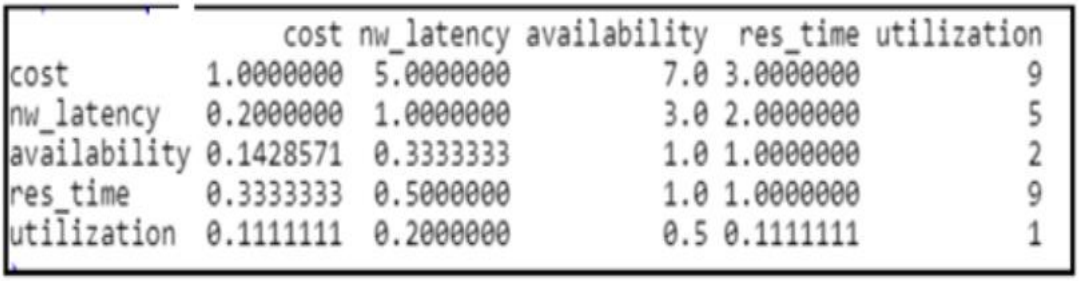
Details regarding the priority of the selected criteria for further analysis are provided in Table 4.6. These details are based on the recommendations made by the AHP. The AHP model provides an explanation of the factors that are prioritized and the weightage that is assigned to each parameter with preferences.

TABLE 4. 6 Priority of the selected criteria

Scale	Criteria	Priority	Reciprocal (divide by 5)
Extremely preferred	Cost of VM	5	1
Very Strongly preferred	Network latency	4	0.8
Strongly preferred	Availably	3	0.6
Moderately preferred	Response time	2	0.4
Equally preferred	Utilization	1	0.2

Experimental setup

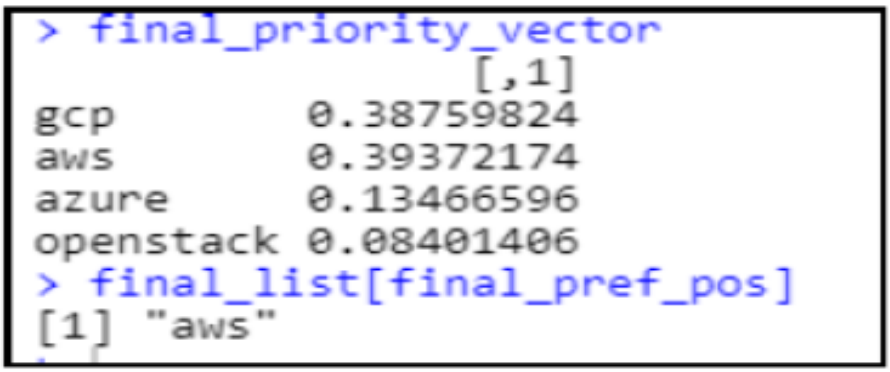
Execute the AHP algorithm in order to obtain the parameter comparison matrix with the five criteria that have been selected, as shown in Figure 4.7. This should be done based on the criteria and priority that have been decided upon.



	cost	nw_latency	availability	res_time	utilization
cost	1.0000000	5.0000000	7.0	3.0000000	9
nw_latency	0.2000000	1.0000000	3.0	2.0000000	5
availability	0.1428571	0.3333333	1.0	1.0000000	2
res_time	0.3333333	0.5000000	1.0	1.0000000	9
utilization	0.1111111	0.2000000	0.5	0.1111111	1

Figure 4. 7 Parameter comparison matrix for ranking

The final ranking of the four clouds using AHP is determined based on these selected criteria, as shown in Figure 4.8 (the output of the R application).



```
> final_priority_vector
                   [,1]
gcp        0.38759824
aws        0.39372174
azure      0.13466596
openstack  0.08401406
> final_list[final_pref_pos]
[1] "aws"
```

Figure 4. 8 Ranking output from AHP

Results and discussions

This data makes it very clear that, when the AHP ranking procedure is utilized, Amazon Web Services (AWS) is ranked first, and as a result, It is used for further processing. You may change the sequence in which these aspects are assessed and add new assessment criteria if you want to. To find the "Cloud selection" according to the ranking given by the AHP algorithm, the system's knowledge and experience are used to meticulously determine the importance and priority of each parameter. The algorithm is then run under different conditions.

Local Ranking of Virtual Machines using PSO-SVM

Prioritization and classification of VM

The previous section explained the goal of global ranking, which is to choose the right cloud and region for deployment at the cloud level. The next step, after deciding on the cloud and region, is to pick the right set of virtual machines (VMs) to deploy. The goal is to make a selection from the twenty virtual machines (VMs) that were first assessed by picking the five best ones. This opens the door for further research into how to improve the workloads running in those virtual machines (VMs) and how to get new workloads up and running at peak performance. There is a belief that the SVM and PSO algorithms work together to achieve this. As mentioned before, the SVM and PSO functions are decomposed into their component pieces. Part one involves classifying VMs using support vector machines (SVM) into "Under-utilized," "Normal," and "Over-utilized." The virtual machines' (VMs') CPU and memory use determines their classification. In this case, the PSO method is used to find the top five virtual machines that are suitable for further investigation and deployment based on the defined fitness function. The virtual machines' (VMs') performance and quality are compared to the jobs that are coming in for deployment. Based on the PSO fitness functions, we will choose the best workload to deploy within the top five virtual machines (VMs).

Data collection

For the proof of concept, we will utilize the data gathered from these twenty virtual machines (VMs) to rank and classify them using SVM-based classification using CPU, RAM, and utilization. From this list, we will choose the top five VMs for further study. In addition, we will choose the underutilized virtual machines for more research. This is due to the fact that "Over-utilized" and "Normal" virtual machines already have all the optimizations they need. Virtual machines (VMs) with a usage rate over 70% are considered over-utilized, whereas VMs with utilization rate below 40% are considered under-utilized. Virtual machines (VMs) typically have a utilization rate of 40–70 percent. At this stage, table 4.7 displays the sample data.

The objective of classifying these virtual machines (VMs) is to find a subset of them that can be studied further in relation to job placement into the VMS.

The right virtual machine (VM) or group of VMs must be used to deploy these activities in order to achieve increases in performance, throughput, and execution time. Virtual machines (VMs) use a plethora of techniques to find the right ones while load balancing these jobs.

We did this to continuously improve work scheduling and execution, which ultimately led to an improvement in the system's overall efficiency.

TABLE 4. 7 Sample data used for classification of VM

VM	CPU %	Memory (mB)	Utilization %	Power (W)
VM1	15	1024	35	185
VM2	35	2048	22	92
VM3	75	1282	47	113
VM4	68	1798	75	108
VM5	32	1540	52	128
VM6	25	1824	28	145
VM7	55	2748	29	112
VM8	77	1212	42	129
VM9	48	1398	85	168
VM10	82	1640	72	178
VM11	88	1824	45	185
VM12	39	1048	92	192
VM13	66	2282	27	213
VM14	85	1798	55	168
VM15	69	1580	59	137
VM16	28	1424	45	165
VM17	41	1648	72	192
VM18	84	1782	57	183

VM19	68	1198	65	238
VM20	37	1340	82	218
VM21	73	1724	95	168
VM22	47	1088	21	192
VM23	71	1782	43	133
VM24	58	1196	65	168
VM25	61	1546	32	198
VM26	52	1694	85	235
VM27	78	2218	42	192
VM28	91	1432	67	103
VM29	26	1098	35	148
VM30	42	1940	82	158

SVM classification for VMs (PSO-SVM)

Step1: SVM-based VM categorization based on CPU and memory use for 20 instances.

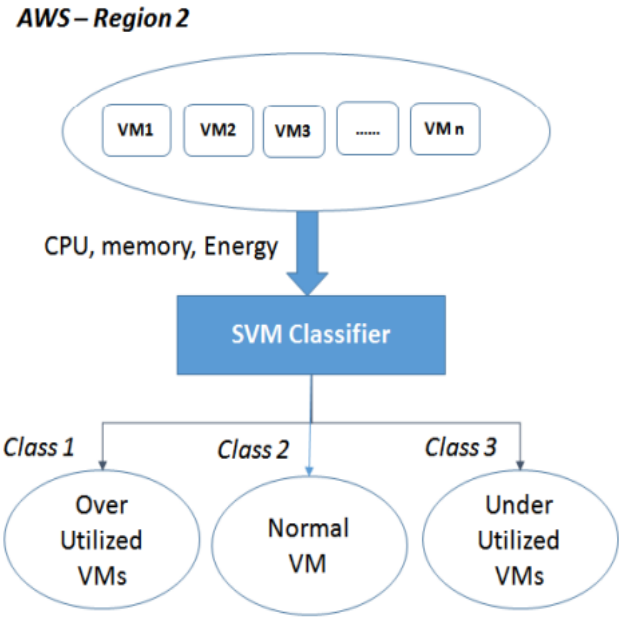


Figure 4. 9 SVM classification for VMs

Diagram 4.9 virtual machines (VMs) are shown to be classified using SVM. The support vector machine (SVM) classifier is being considered for this case's classification because it works well in non-linear space, uses a hyper plane in multidimensional space or a non-linear curve with any kernel chosen during the process, and contributes to improved classification and accuracy.

The SVM classifier was run using the classification criteria stated above as a first step in choosing under-utilized virtual machines (VMs).

TABLE 4. 8 Confusion matrix for SVM classifier for VMs

Category for SVM	Percentage	Number of VMs (total 20)
Under-utilized	0 - 40%	5
Normal	40 - 70%	9
Over utilized	70% above	6

Based on the results of the SVM experiments, five virtual machines out of twenty virtual machines are being examined for further ranking in the AWS cloud for the deployment of new workloads. According to the confusion matrix, the accuracy of the classification is 98.06 percent. The goal here is to go through all of the available virtual machines (VMs) and choose the ones that might benefit most from further training and optimization.

Ranking for VMs using PSO

In this part, we will go over the five virtual machines (VMs) and the additional tasks that need to be implemented in them using a technique called Particle Swarm Optimization (PSO). The PSO algorithm uses its local and global best to determine its fitness value, and it gives a better approach for finding the fitness criteria. Despite the availability of several task scheduling techniques, this remains the case. Consideration of the cost and execution time on a particular virtual machine informs the modified PSO algorithms used for scheduling critical jobs. Next, we'll look at how the PSO algorithm explains the method utilized for job allocation and scheduling.

Data collection for workload in pipeline for deployment

On the basis of the first-in, first-out (FIFO) principle, this section of the analysis identifies eight distinct tasks that are available in the queue, each of which has a different duration.

The goal is to identify the five virtual machine combinations that are most effective for deploying the eight jobs in a manner that is both cost-effective and minimal in terms of time.

The specifics of these eight distinct tasks, including their durations, are presented in Table 4.9 for the purposes of scheduling and deployment. The scheduling procedure for the work that is going to be deployed on virtual machines is shown in Figure 4.10.

TABLE 4. 9 Task and Length in MI

Task	Length
t1	3200
t2	1952
t3	6293
t4	3750
t5	5576
t6	4165
t7	6840
t8	4513

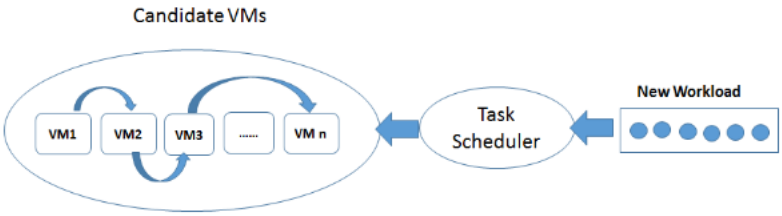


Figure 4. 10 Scheduling Process for tasks onto VMs

Workload scheduling and placement using PSO technique

To further rank the detected 5 VMs from the 20 VMs, we apply the following VM factors:

• Mean time it takes for a job to run in every virtual machine.

• The average cost of running a job in each virtual machine.

• Virtual Machine Power

• Virtual Machine Usage on Average.

Out of all the virtual machines (VMs) that were screened using the SVM classifier, the top five were chosen for this study. Virtual machine (VM) parameters including performance, bandwidth, input/output (IOPS), and cost are used for further ranking purposes. This study has shown that the five virtual machines can handle eight tasks, based on the number of tasks and the length of each work. The execution time, execution cost, and power of the virtual machines in MIPs (million instructions per second) are calculated for each case as part of the process of deploying each identified job onto each VM. This paves the way for finding the best match. Several factors are considered for assessment, including execution time, execution cost, average VM usage, and VM power. To assess these aspects, we apply the method that will be detailed in the section that follows. Choosing virtual machines (VMs) in this case is analogous to how swarm particles choose food in the search space, considering their present location and motion speed. In its quest for food, it draws on both its own wisdom and what it has learned from the outside environment. In the part that follows, we go into more depth about this behavior.

Here, we use the following equation to determine where the swarm will land.

$$X_i(t) = X_i\ (t\text{-}1) + V_i(t) \tag{4.3}$$

X_i (t) represents the particle's location (VM) during iteration "t"

X_i (t-1) is the position of the particle (VM) during iteration t-1

V_i (t) is the Velocity of particle during iteration “t”

$$V_i(t) - w * V_i(t\text{-}1) + r1*c1*(pbest - X_i(t)) + r2*c2*(gbest - X_i(t)) \tag{4.4}$$

Pbest represents the particle's optimal local location, whereas gbest represents the particle's optimal global location with regard to a population. The starting weight is denoted by W, and the random integers between 0 and 1 are found in r1 and r2. (1.6 is believed to be a random value) The acceleration coefficients are denoted by c1 and c2.

The following part provides an explanation of how the PSO algorithm is modelled by making use of multiobjective functions and how By using objective functions, it

helped optimize operations for the purpose of ranking virtual machines. This step involves calculating and validating the fitness function for each virtual machine in relation to the tasks that will be deployed.

Procedure for performing PSO

Procedure for PSO ( )

```
SetInitialValue(Swarm, Velocity, Position, Pbest)
        while ( condition is not met for stop criteria ) do
                for p ϵ P do                                    // iterate for all particles
                        f=fitness-value-compute(Swarm(p));    //compute fitness
                        Pbest =verify-fitness(f)
                        value-update(Velocity, Position)    //

                end for
        end while
return Pbest
end Procedure
```

The next section explains how the goal works, including how long it takes, how much power it uses, how much it costs to execute tasks, and how virtual machines (VMs) are used to schedule and deploy the right set of tasks onto the right set of VMs to get the best result. To establish the relative significance of each factor—cost, time, power, and utilization—the priority is utilized to give weights to each. The next section provides a representation of the pseudo method that is used for work scheduling of virtual machines (VMs) based on ranking and goal functions.

Task scheduling algorithm

Procedure for Task Scheduling T, VM

```
For each Task T_i do
        For each VM   do (v ∈ VMs)
                ExeCost(t,v) <-  EstimateExecCost(t,v)   =>  cost of each task
                ExeTime(t,v) <-  EstimateExecTime(t,v)    =>  task execution time
                AveUtil (t,v) <-  EstimateAveUtil(v)   => Average Utilization of VM
        End for
End for
```

Initialize (swarm, velocity, position, pbest)

For (i ∈T) do

For (v ∈ VMs) do

F = fitness-value-compute(VMs, ExeCost, ExeTime, AveUtil)

pbest(i) = verify-fitness(F)

value-update (velocity(i), position(i)) // using equation 1 & 2

End for

End for

return pbest;

end Procedure

Algorithm for Computing Cost-of-Execution(ExeCost)

Cost-of-Exe(i,j) = Total_ProCost(i,j) +Total-IOCost(i)

whereCost-of-Exe (i,j) is the task execution cost of the task i in VM_j

Total_ProCost(i,j) -> cost of execution of task i in VM_j

Total-IOCost(i) -> cost of transferring the data file of the task "i".

Algorithm for Task Processing Cost (TPC)

Total_ProCost(i,j) = Process-Time(i,j) + VM_Price(j)

Process-TimeT(i,j) => Processing time for task "i" in VM_j

VM_Price(j) => Price of VM_j

Process-Time(i,j) = Task-Length(i) / PSU(j)

where Task-Length(i) length of Task "i" in MI and PSU(i)Processing speed of VMj in MIPS (million instructions per second)

Total-IOCost(i) – FS(i) * BC

FS(i) = size of task I input and output file in MI

BC= Bandwidth cost

Algorithm for Averge Utilization of VMs

Average VM Utilization (AvgUtil): It is the average utilization of all the VMs in terms of the CPU, memory, storage and bandwidth used by all tasks to finish execution.

Pseudo algorithm

Procedure for Average Utilization

For each VM do (v∈ VMs)

$$VM\_CPU(i) = \sum_{i=0}^{n} CPU\_Used(i)/Total\_CPU(i) * 100$$

$$VM\_mem\_util(i) = \sum_{i=0}^{n} Mem\_Util(i)/Total\_Mem(i) * 100$$

$$VM\_storage(i) = \sum_{i=0}^{n} Storage\_Used(i)/Total\_storage(i) * 100$$

$$VM\_Bandwidth(i) = \sum_{i=0}^{n} Used\_BW(i)/Total\_BW(i) * 100$$

AvgUtil(i) = (VM_CPU(i) + VM_mem_util(i) + VM_storage(i) + VM_Bandwidth(i)) / 4.0

end for

end Procedure

Where

AvgUtil (i) is the utilization average of VM i

VM_CPU(i) represents the utilization value of CPU of VM i

CPU_Used(i) represents the used CPU for all tasks executed in VM i

Total_CPU(i) represents the total CPU of VM i

VM_mem_utilrepresents the memory utilization of VM i

Mem_Util(i) represents the used memory for all tasks executed in VM i

Total_Mem(i) represents the total memory of VM i

VM_storage (i) represents the storage utilization of VM i

Storage_Used(i) represents the used storage for all tasks executed in VM i

Total_storage(i) represents the total storage of VM i

VM_Bandwidth(i) represents the bandwidth utilization of VM i

Used_BW (i) represents the used bandwidth for all tasks executed in VM i

Total_BW (i) represents the total bandwidth of VM i

Here we will go over the ranking criteria and how to use them to calculate the fitness function, complete with a pseudo method.

Pseudo code for Compute Fitness functions for ranking VM

```
ComputeFitness( VMs, ExeCost, ExeTime, AvgUtil )
   Initialize (Rank)
        V1 ←RankSort(VMs,ExeCost)  =>  Sort ascending based on ExeCost
        V2 ←RankSort( VMs,ExeTime)  => Sort ascending based on ExeTime
        V3 <-RankSort(VMs,VMP)  =>  Sort descending based on the VM price
        V4 <-RankSort(VM,AvgUtil )  => Sort VM based on the Average Utilization.
        For v ∈ VMs do
              Compute-Rank(v) <-  V1(v) + V2(v) + V3(v) + V4(v)
        End for
        RankSort (rank)   // ascending order on rank
returnCompute-Rank()
 End procedure
```

The results are calculated in a systematic way by applying the fitness functions of PSO with various objective criteria, utilizing the aforementioned set of modified PSO algorithms. All of these fitness functions are averaged and then weighted to determine the ranking of the virtual machines. The next step, after calculating the weighted average, is to find the set of VMs that is the most valuable. Afterwards, these virtual machines are designated for the job delivery. The whole procedure will be detailed in this section, including every single detail.

4.2.3 TASKS SCHEDULING AND PLACEMENT

Crucial to cloud management as a whole is a process called "Task Scheduling" that ensures resources are balanced and assigned workloads accordingly. Determining in advance which cloud resource is suitable for deployment is vital since different users have varied application workloads that must be installed on the cloud. In order to function as efficiently as possible, this resource should have enough resources like CPU, RAM, and others. In order to gets the greatest potential outcome in cloud computing, work scheduling is crucial. Put simply, it's the act of allocating specific start and finish times to the various jobs that are bound by certain constraints. Both time and resource restrictions are examples of the constraints that may be present. By dispersing the workload among a number of distinct processors, task scheduling helps to maximize the usage of available resources while diminishing the amount of time

required for execution. People, in general, consider the computing cost, the data transmission cost, the makespan, the energy usage, the service level agreement, and other factors in order to effectively manage their workload. Cutting down on execution costs and waiting times should be high on the list of priorities, along with addressing resource availability and scheduling issues that hinder efficient use of available resources. Finding out whether the workload is memory-or CPU-intensive is the goal, and then to choose which virtual machines or cloud services are most suited for deployment.

When dealing with a multi-cloud environment, it becomes even more difficult to assign the appropriate workload to the appropriate cloud. This is due to the fact that there are numerous other aspects, such as cost, network latency, policies, interoperability, and so on, which were covered before. In the second place, when the workload is shifted to a different cloud, it should be portable and interoperable between the different cloud services. According to what was mentioned previously, container technology is providing efficient support for this portability between cloud environments. In addition, the deployment of applications across various clouds must be capable of communication and collaboration, and it must be coordinated in the appropriate manner in order to carry out a work without any interruptions. To manage and orchestrate a wide range of tasks—including load balancing, service management, build and release management, resource management, and many more—several orchestration technologies have evolved throughout the years. This simplifies the system's administration and user experience. Additional operational challenges include security, monitoring, integration, and deployment, among others, are assisted by a variety of tools and processes, such as devops, which make it easier to manage many clouds.

Two components make up this study's examination of work scheduling and placement strategies.

a) Relocating containers from underused virtual machines to overused ones, using ranking and predicted use.

b) Allocating new workloads (tasks) to the most suitable cloud and virtual machine (VM) while keeping cloud use and performance in mind Choosing the right resources

for the workload in a cloud system is no easy feat, what with all the customers requiring the cloud to handle their workloads simultaneously and their potentially different resource needs. This study takes the following crucial factors into account with respect to the placement and timing of tasks.

a) Locate a resource with greater available space for deployment and lower memory/CPU consumption.

b) If the usage rate exceeds 80%, you should relocate the task to a resource with a lower utilization rate.

b) Determine which resource has the lowest execution cost and the shortest waiting time.

The goal of the proposed method is to reduce overall processing cost in any chosen cloud by matching workloads (tasks) with the most suitable virtual machines (VMs). This is achieved by placing a premium on critical metrics including task priority, execution time, and virtual machine performance. Matching, allocating, and scheduling are the three parts that make it up. The goal of these procedures is to shorten the make-span, the total completion time, as much as feasible by mapping client requests or tasks to the cloud's virtual machines (VMs). Figure 4.11 details the process flow of scheduling jobs on VMs using the PSO approach; it is accessible below.

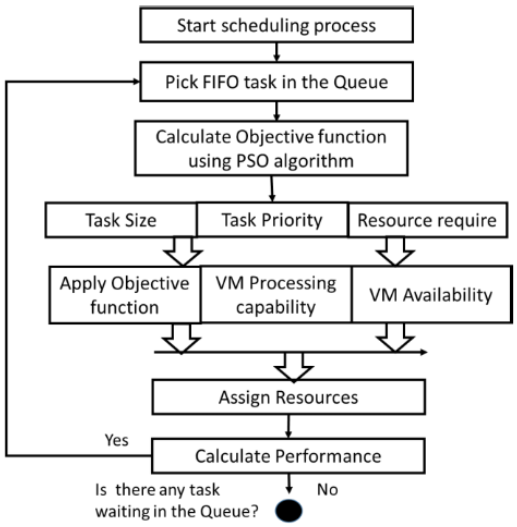


Figure 4. 11 VM Scheduling workflow

4.2.4 RESULTS AND ANALYSIS

Let T = {T_0, T_1, T_2, … . . T_n} identify the group of jobs that need to be sent to the clouds. For each j, let Vj = {P, B} represent the corresponding virtual machine with processor speed P in MIPS and bandwidth B in Mbps. In a similar vein, let Ti = {I, D} represent the i^{th} job with the dataset D in megabytes and the set of instructions I in million instructions. The following is the expression for the execution time of T_0 on Vj:

ETC_{ij} = (I/P) + (D/B)

In accordance with the PSO method described above, using the aforementioned pseudocode, we can calculate the job execution cost and the task processing cost. To put the approach into action, all of the virtual machines (VMs) are assigned a task to calculate the three parameters and then rank them based on the weighted average of the four parameters.

TABLE 4. 10 Ranking of VMs based on the fitness function

Task Id	VM1	VM2	VM3	VM4	VM5
Task 1	4	1	2	5	3
Task 2	3	1	2	5	4
Task 3	4	3	2	5	1
Task 4	4	1	2	5	3
Task 5	4	3	4	5	1
Task 6	4	2	1	5	3
Task 7	4	3	2	5	1
Task 8	4	3	2	5	1

Rank = W1*ExeCost + W2*ExeTime + W3*VMP + W4*AvgUtil Where W1=0.4, W2=0.3, W3=0.2, W4=0.1

As a result of using the equation shown above, For every deployment task, Table 4.10 shows which virtual machines (VMs) are prioritized. The study takes into consideration the aforementioned factors when determining the best virtual machines (VMs) to perform the given tasks. Cost is given 0.4 weight, execution time 0.3, quality/performance 0.2, and utilization 0.1. The weighing of these three factors is applied to every virtual machine (VM), and the VMs are ranked in order of

importance based on the specific job that needs to be executed. By arranging customer requests (or tasks) according to cloud resources, the ranking process hopes to achieve all of the following goals.

Use of the global and local ranking approach ensures consistent and meaningful results by guiding the selection of the best cloud among multiple clouds and, within that, the selection of the most suitable virtual machines (VMs) for scheduling and deployment tasks. To aid in the examination of the environment under defined circumstances in relation to a set of business criteria, cloud ranking is an essential component. The right choice can be made with the help of this analysis, which uses metrics based on the current state of the environment. There needs to be almost no human involvement in the decision-making process if it can be fully automated. Accurate decision-making, consideration of business priorities at different levels, and improved optimization for workload deployment on selected resources are all made possible by this framework's most crucial component—the ranking process. Based on this study, the ranking approach to selecting the best cloud resources is novel and different, and it is considered an effective way to manage tasks involving multiple clouds.

4.3 OPTIMIZATION, PREDICTION AND KNOWLEDGE MANAGEMENT

The resources available in the cloud are inherently unpredictable, with many resources undergoing rapid state changes in a very short amount of time. Continuous monitoring of cloud resources leads to optimization, which in turn reduces redundancy and makes the most efficient use of cloud resources for a specific set of processes. Infrastructure, networks, storage, and costs all have varying degrees of optimization. The goal of this study's cloud optimization is to achieve business goals with a little investment in cloud resources by eliminating duplication at every level.

Following are the key objectives of cloud optimization process.

- **Optimization of Resources:** Numerous resources, including network, storage, and computation, are essential in a standard cloud setup. Optimizing the use of these resources, which results in cost savings, requires continuous monitoring and

measurement of their parameters.

- **Cost Optimization:** By making the best possible choices when it comes to cloud services, APIs, functionality, network connection, software licensing, and more, it is possible to significantly reduce costs.

- **Optimal scheduling of tasks:** Important parts of task scheduling include determining which cloud resource is best suited to the workload and then equally dividing that resource among several cloud platforms according to performance and consumption of infrastructure resources.

- **Optimization of processes:** building, releasing, archiving, backing up, restoring, continuous integration, deployment, etc. are all examples of operational processes.

Better optimization may be the outcome of increased developer productivity and decreased mistake rates brought about by the automation of various operational procedures.

4.3.1 TECHNIQUES FOR OPTIMIZATION

Managing resource allocation is a critical component of cloud computing, but it may be challenging because of all the conflicting demands on the system. Optimization entails selecting the optimal solution from among all those that are applicable to the search space. The primary goal of developing such algorithms is to model solutions that rely on some objective function. To find the optimal solutions, these objective functions are evaluated using several methodologies. Objective functions may be of the single or multiple kinds, depending on the number of objectives used to assess the answers. In multi-objective optimization, each of the objective functions is assessed with the aim of finding their optimum value. To achieve each goal within the constraints of the search space, a vector consisting of decision variables is used. Modern programs must minimize resource use, making optimization strategies crucial. Since most modern apps function in real-time, cutting down on wait times and speeding up the distribution of resources are of paramount importance, which makes time a crucial factor when making optimization decisions. As a rule, optimization methods aim to meet resource demands by allocating resources as efficiently as feasible given the limitations at hand.

Typically, optimization algorithms are iterative, meaning they will continue to compare several solutions until they find the optimal one. Finding the optimal value for a collection of parameters in order to accomplish certain objectives is the main focus here. The goal to be achieved and the various solutions, or states, of an optimization issue are both defined. Optimization algorithms come in two main varieties.

- **Deterministic approach:** The assessment method relies on the computation of statistical parameters and the derivation of outcomes from predefined equations. In this case, the system will always go through the same states and generate the same output for a given set of inputs. The design of several technical challenges has been made possible by these methods.

- **Stochastic approach:** There are norms for probabilistic transitions in nature. Using a mathematical or statistical function with input parameters that are prone to randomness, it seeks to maximize or minimize the value of the function. The goal functions, design variables, and restrictions that these algorithms must adhere to are extensive. Maximization and minimization are the two possible forms of objective functions. Problems involving minimization or maximizing inspired the development of optimization algorithms.

- **Heuristic and meta-heuristics:** Useful for constructive algorithms, heuristics rely on local search techniques. The word "to find" describes them. In order to find what you're looking for, meta-heuristics follow a trajectory across the search space, which implies that it will be different every time.

Survey of Cloud Optimization Algorithms

Since the majority of cloud resources are already in use, it could be difficult to find enough of them to complete any given activity or demand.

Without proper attention, the resource allocation process, an essential component of cloud management, may have a negative effect on the system's overall performance.

Optimized resource utilization is a great strategy for allocating resources, which helps the cloud provider save money and make better use of their resources.

When it comes to cloud computing, two main allocation strategies are often used. In the first level, task scheduling divides up available virtual machines (VMs) across the available tasks.

In the second level, the duty is to allocate the available VMs to the hosts where they are scheduled to execute. What this means is that the most resource-intensive apps in the cloud will have their share of the available resources distributed among them.

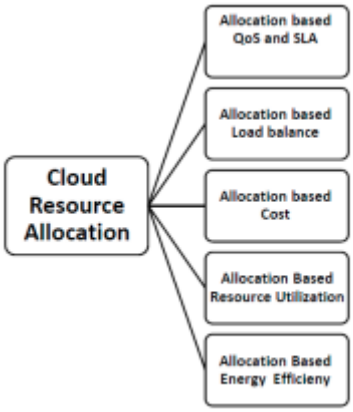


Figure 4. 12 Resource / task allocation process

The numerous choices for job allocation are shown in Figure 4.12. Because cloud resources in multi-cloud environments are geographically distributed across several locations and are subject to changing load circumstances, user needs, and pricing models, resource allocation is a challenging problem. When it comes to resources, pricing models, quality of service (QoS), and service level agreements (SLAs), cloud service providers are all different. Among the issues with multi-cloud is the need for optimization approaches to resource allocation, which includes things like load balancing, energy efficiency measurements, cost optimization, and resource usage. In order to better manage and monitor their systems, cloud providers often use optimization algorithms to increase resource usage and decrease costs. In order to solve the problems of resource allocation and optimization, several academics have created optimization algorithms. In order to meet one or more optimization goals, the following part provides a detailed discussion of a few algorithms that were created for resource allocation.

- **Greedy algorithms:** This method relies on heuristics to discover the best answer; it finds the local optimum option at each iteration value until it achieves the global

minimum. In its pursuit of global optimization, a greedy method often fails to provide an optimum solution as it is prone to hitting local minima. Efficient task scheduling across cloud resources allows greedy techniques to optimize costs in diverse cloud environments.

- **Genetic Algorithms (GA):** The method derives from meta-heuristics used to solve optimization problems, both with and without constraints. Mutation, hybridization, and natural selection are the processes upon which biological evolution is built. Its typical applications include finding near-optimal solutions to various combinational problems, reducing issues like make span in task deployment, and balancing workloads across cloud resources.

- **Particle swarm optimization (PSO):** Applying PSO in the cloud optimizes workload scheduling in a distributed heterogeneous environment across resources, with the goal of improving QoS. Several multi-objective models have been built using PSO and variants of PSO to optimize task scheduling issues such as minimizing execution cost, transferring time, and execution time. The algorithm begins with a population of randomly chosen solutions and iteratively changes its location and velocity in the search space to update its knowledge of the space and find the optimal solution. Each particle uses its own unique set of strengths, both internal and external, to identify the greatest possible answer.

- **Artificial Bee Colony (ABC) optimization:** The swarming foraging tactics of honey bees served as inspiration for this population-based heuristic. The amount of food source in the nectar corresponds to the quality or fitness of a potential solution in ABC, and its position is a possible optimization challenge. Using ABC to fairly distribute workloads across virtual machines is a common optimization strategy for the cloud. This helps to improve throughput and decrease makespan.

- **Ant Colony optimization (ACO):** This population-based algorithm uses pheromones generated by ants heading towards food to determine the optimum way to take, and then iteratively builds a solution by following that path. The model depicts the many ant pathways each ant follows the pheromone signal that leads to the food source, creating a connected network. by the one before it. The optimal option is the one that yielded the food in the least amount of time. The

ants add weights to values during runtime, making it a biased model. The ant's progress toward food determines the strength of the pheromone discharge. To find the best way to distribute resources across jobs in a dynamic cloud system in order to reduce the overall makespan of tasks, the ACO algorithm is used in the cloud setting.

- **Cuckoo algorithm:** Cuckoos, which deposit their eggs in the nests of other birds in the hopes that the host birds would hatch them, served as an inspiration. To stop the cuckoo from depositing eggs in their nest, some host birds may go to extreme lengths to defend their territory. When a host bird finds out its eggs aren't their own, it may do one of two things: either discard the eggs or start a new nest elsewhere. Search optimization for cuckoos. Cloud optimization issues, using an algorithm that imitates the behavior of cuckoo birds during mating might solve problems like job scheduling and load balance. Because it improves load balancing across resources and decreases overall execution time or cost, it solves the scheduling problem in cloud computing.

- **Fish Swarm Algorithm:** This program takes its cue from the way swarms of fish swim together to find food. If the school is still unable to locate the food source after swimming a certain distance, it will turn around and try again, it abruptly alters its course. Finding the origin and heading toward positive weights is the goal here. Some notable features of this method include a rapid convergence rate, insensitivity to beginning values, increased flexibility, include the ability to tolerate errors. In order to find the smallest possible resource allocation for executing tasks in the cloud, taking task priority into consideration, and the search approach is used to the visual area of fish behavior.

Cat Optimization, which relies heavily on the seek-and-run strategy, and Bat Optimization, If the food supply (object) is blocking signals, the process may be exploited, are two examples of such algorithms. Cats spend more time in seek mode, when they constantly scan their surroundings for potential food sources. As soon as it determines where its meal is, it races at full speed to seize it. Even though optimization with this sort of behavior is not common, it has been tried and tested on the cloud.

MDP-RL and deep learning

Reinforcement learning (RL) is a method of training where the agent gains knowledge by consistently interacting with the system. The agent adapts and performs new behaviors based on what it has learned from its past experiences in order to reach its aim. Using no prior data, the system's functional model decides the objective and applies several rules. It is the intention of the developers of the system that all parts are directed towards activities that provide the most profit by using a penalty and reward system. To maximize the decision maker's benefit, it mandates that an agent learn from its experiences in the environment and acts accordingly. A key function of RL in the Markov model is the identification of policies that facilitate the most efficient attainment of objectives. To achieve the objective, it may be necessary to use more than one policy. However, It seems that the agent has mastered the system when it follows the optimal policy, which is the best policy.

Limitations and challenges of Optimization Algorithms

job scheduling, load balancing across (VMs), optimizing cloud resources, and job allocation are all areas where the techniques covered up to this point find application. Their primary role is to search for food sources with optimum functions, much like how real species do it, by modeling their behavior after that of optimization algorithms. While there are benefits to each of these algorithms, researchers have tweaked the algorithms to make them even more optimized by changing their behavior and adding new features. While the majority of them are used for functions with a single purpose, there are a few that are also utilized for optimization in the search space with multiple objectives. Although several of these algorithms have recently been expanded to multi-cloud functions, Most of them were first evaluated for use with individual cloud providers.

The following are some of the main limitations of these methods when applied to situations involving several clouds.

- Any one of these algorithms, or even an improved version of it, cannot take a comprehensive view of the problem in a multi-cloud setting and handle the complexity entirely because there are so many parameters and objective functions to consider when optimizing for multi-cloud.

- When it comes to multi-cloud, optimization is key, both inside and between clouds. This means that virtual machines (VMs) in a single cloud need to be optimized locally, and optimization across clouds has to be optimized worldwide.

- All of these algorithms optimize job scheduling within individual virtual machines (VMs), but they don't take into account the effects of the VM's neighbors, such as the unpredictability and network latency that the VM experiences as a result of its neighbors.

Three tiers of optimization are used in this study.

a) **Prediction** of capacity required to handle future workloads, which may be achieved by removing underutilized virtual machines or by introducing more virtual machines than the current capacity allows.

b) **Optimization** of resources that are currently operating within (VMs), transferring or importing containers inside VMs, and releasing unused VMs.

c) **Knowledge Management** using the knowledge gained from prior optimization and prediction efforts in comparable contexts.

Markov Decision Process and Reinforcement learning

Key optimization strategies in this study include (MDP) with reinforcement learning as a support system. Keeping these optimization techniques in mind, as opposed to the other methods mentioned before, is important since cloud environments are inherently unpredictable and system states may change quickly. Decisions should be grounded in the present rather compared to earlier times if they want to achieve success, and regular monitoring of resources and systems is crucial for this. One would wish to train the system by continuous engagement with it and reward and punish it towards attaining the intended objective, which is why the Reinforcement approach is being considered. Systems that are inherently unpredictable and ever-changing, such as the cloud, benefit greatly from the application of reinforcement learning techniques, which, when coupled with Markov Decision Process (MDP), provide even greater results.

A machine learning approach known as Reinforcement Learning (RL) uses an agent

to continuously interact with a system in order to analyze and comprehend its actions. Artificial intelligence (AI) has embraced the learning approach to guide unsupervised machine learning using incentives and punishments. RL enables the agent to adapt its actions in response to environmental input. One way that people learn is via repeated interactions. Very few Reinforcement Learning algorithms, even when given exact instructions to maximize reward, will converge to the global optimum. The majority of the learning is accomplished autonomously by this automated learning technique, which allows for little human involvement. A kind of behavior modification known as "reinforcement learning" involves the use of positive reinforcement and negative punishment to shape behavior.

Reasoning Layers (RL) falls somewhere in the middle, neither quite in the unsupervised learning camp nor quite in the supervised one. It is a unique method of learning in which the agent receives positive reinforcement for good conduct and negative reinforcement for bad behavior. The agent is designed to find the best answer by maximizing its reward in the long run. There are two stages to this method as well: training and prediction. Q-Learning is a specialized method of learning that makes use of a table to depict "State-Action" pairs together with predefined rules. The table is continuously updated with rewards and penalties as the agent goes through numerous rounds of comprehending the system's behavior via ongoing interaction, progressing towards the intended objective. The following sections provide more detailed explanations of the Markov decision process. There are several difficulties with RL, despite its great benefits. Problems include the fact that it uses a lot of memory to keep the values of each state, which may become an issue when the problem is complicated and there are many states. Decision makers use value approximation methods, such Neural Networks or Decision Trees, to address such issues. Researchers strive to lessen the influence of these inaccurate value estimates by exploring various ways and putting greater emphasis on solution quality, but there are numerous drawbacks associated with employing them as well.

When the decision-maker has some influence over the outcome but the outcome is also somewhat random and each change in state is the result of an action, (MDP) is used to simulate the decision-making process. In general, there are several Markov model options to choose from. Every possible state S in the discrete Markov model

has the capacity to carry out an action A with a probability P and a corresponding reward R. Reinforcement learning becomes problematic when both the likelihood and the benefits are uncertain. In this study, we use Q Learning—specifically, it's most fundamental implementation, the Q table—to resolve one such elementary issue. Solving the Markov decision problem (MDP) is a common application of RL. It is possible to formalize almost all RL issues as MDP. The next section provides a detailed discussion of the MDP learning and prediction model.

The application of MDP and RL in the cloud optimization problem allows for continuous monitoring of the behavior states of cloud resources and the implementation of corrective actions utilizing state-action pairs. Achieving goal-related goals requires meticulously examining the system's behavior due to the inherent unpredictability of cloud resources. The closer the function gets to the goal, the more rewards and penalties it receives. Training uses RL's reward and punishment processes to teach the system how to behave, and prediction mode is when it uses this knowledge to guess what to do in a particular situation to maximize value.

Experimental Analysis

Using the AHP ranking algorithm, the goal of this research stage—optimization inside the selected cloud—is to optimize the virtual machines (VMs) locally, once they have been rated by the SVM-PSO algorithm using the local ranking process. The virtual machines (VMs) that were found using "SVM" based categorization are now being considered for internal optimization in preparation for container deployment. Within the cloud, there are two steps for doing local optimization at the virtual machine level.

- Docker image size reduction approach should be applied to all regions and clouds in order to optimize the size of container images in virtual machines.

- Make a decision on the state-action combination that will maximize throughput by assessing the utilization state and then deciding to migrate to other virtual machines or clouds for optimal performance. This will ensure that the application or load is distributed equally across all virtual machines, leading to optimal system performance.

In this research, a cloud application system agent is used to keep tabs on virtual

machines (VMs) and react to changes so that VM performance is optimized. Nothing will be done to enhance environmental management as a result of this. One way to represent this state-action combination is via a Markov model. A reinforcement learning challenge using a punishment and reward strategy may be associated with an agent's understanding of such a dynamic and changing environment.

A technique called Q-Learning for learning by reward. It can calculate the actions' utility even in the absence of an environment model. With the aid of the action-value pair and the anticipated benefit of the present action, it does this. As it moves through its environment, the agent learns to maximize its reward by doing the action that is now most advantageous. Using the current action as a starting point, execute a random action to find the maximum value of Q if the current state is not final (i.e., optimal usage and performance). Starting with the current state and progressing to the next state in an iterative fashion, the system eventually achieves its goal and that cycle repeats itself.

Environment setup and analysis

Workload deployment according to the available resources now, optimization research includes doing tasks on the selected cloud. Continuing to enhance the existing environment should be the primary goal once the ranking process-based deployment of additional workloads has been completed. To ensure optimal performance in the cloud, it is desirable to have all virtual machines (VMs) returned to their normal condition following optimization, without any instances of under- or over utilized VMs. Using the Q-Learning table, this experiment takes into account and designs five distinct states and five distinct actions.

States description:

- Virtual machines are said to be in a "Over-utilized" condition if their CPU and memory utilization exceed 70%.
- Virtual machines are in a "Under- utilized" condition if their CPU and memory utilization is below 40%.
- Virtual machines are regarded to be in a "Normal" condition when their CPU and

memory use is over 40% and below 70%, respectively.

- The "Unutilized" state is reached when the virtual machine's usage is "0 %" or very near to it.

- Virtual machines are in a "Fully- utilized" condition when their CPU and memory usage is around 100%.

Action Description

- To get virtual machines (VMs) back to a "Normal state" when they're not "Over utilized," you need to "Move-Out" their containers.

- To get virtual machines (VMs) out of the "Under-utilized" condition and into the "Normal" state, you must "Move-in" containers into the VMs.

- "De-provision" virtual machines from the system if they are in the "Un-utilized" status.

- The appropriate step to perform when virtual machines are in the "Fully-Utilized" state is to "Provision" more virtual machines into the system.

- "No-Action" is the appropriate system action to perform when virtual machines are in the "Normal" state.

Based on the previously developed state-action model, Table 4.11 specifies that the system's initial Q-table should have all "0" values.

TABLE 4. 11 Q-table with initial state using MDP-RL

State-Action	Under-utilized	Over-utilized	Un- utilized	Fully-utilized	Normal
Move-in	0	0	0	0	0
Move-out	0	0	0	0	0
Provision-VM	0	0	0	0	0
De-provision-	0	0	0	0	0
No-Action	0	0	0	0	0

Step 1: The present status of the system dictates the course of action: either transfer containers in or move them out to restore normalcy to all the virtual machines,

depending on whether they are under- or over-utilized.

Step 2: Ultimately, we want everyone to be back to normal, so we're trying to figure out the "Optimal Policy"—the fastest way to get there. In order to achieve the goal of "All VMs are in normal state," one might use various rules and follow different paths in the Q-Learning table. A policy is considered "optimal" if it follows the shortest route.

Step 3: A "move-out" or "move-in" choice is considered a reward if it moves the system closer to the objective, and a punishment if it moves it farther away. A "+1" indicates a prize, whereas a "-1" indicates a penalty. In the case of "No-Action," it is integer zero. You can see the State-Action flow for VM using MDL-RL in Figure 4.13.

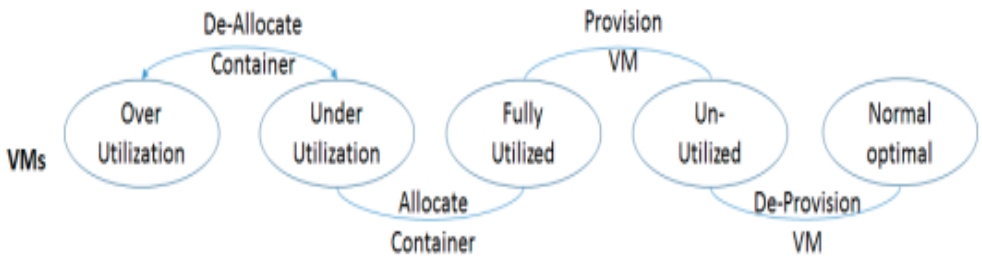


Figure 4. 13 VM State – Action flow for MDP-RL

Step 4: Following this explanation of the pseudo code and the Boltzmann equation, the appropriate action will be done in light of the present situation. To get the numbers in Q Table, we use the following

$$Q[s, a] = Q[s, a] + alpha*(R + gamma*Max[Q(s', A)] - Q[s, a])$$

Let learning rate (alpha) as 0.5 and discount factor (gamma) as 0.9, Where

$$Q(state, action) = R(state, action) + Gamma * Max[Q(next\ state, all\ actions)]$$

Max[Q(s', A)] gives a maximum value of Q for all possible actions in the next state.

States = { +1,0,-1}, Actions { +1,0,-1,-5,+5}, Rewards { 0.+1}, Punishment (0, -1 }

&Goal = 100.

Step 5: The system will provide additional virtual machines when all of them reach the "over-utilized" stage. If there are too many virtual machines that are "un-utilized," the system will remove them.

Step 6: An iterative process including rewards and punishments is carried out based on the present condition, with the goal of reaching a value of 100. Assuming all virtual machines (VMs) are operating at full capacity, this indicates that optimization is no longer necessary and the system has returned to its baseline condition.

Q Learning pseudo code

```
Define epsilon, alpha, gamma
Initiate Q-table with Zeros
Observe initial state 's'
Repeat:
    Select an actiion 'a'
        With probability e select random action 'a'
        Else select action a = argmax(Q(s,a'))
    Carry out action 'a'
    Observe next state s' and reward 'r'
    Calculate Q-target = r + gamma*max[ Q(s',A) ]
    Calculate Q-delta = Q-target – Q (s,a)
    Add alpha*Q-delta to Q(s,a)
    s = s'
until terminated
```

Results and discussion

Here, It is presumed that the system reached a stable state and accomplished the aim of 100 after 278 rounds of consistently updating the rewards and penalties and using the Q-Learning table, at which point all virtual machines were in a "Normal" condition and no action was needed. After it meets the objective, the revised Q-Table value is shown in Table 4.12.

TABLE 4. 12 Updated Q-Table using MDP RL

State-Action	Under-utilized	Over-utilized	Un-utilized	Fully-utilized	Normal
Move-in	53	42	49	54	45
Move-out	21	64	78	56	22
Provision-VM	36	85	12	82	94
De-provision-	17	21	89	94	69
No-Action	39	34	69	79	100

Here we have two distinct objectives, one of which is to attain 94 and the other 100. For subsequent decision-making, the optimum route is the one that maximizes reward while still accomplishing the end objective.

In the future, one may utilize this knowledge to make predictions, such as deciding what to do next if a virtual machine's state-action is already on the best route to achieve its goal. Based on the system's present condition, this MDP process helps it make a corrective choice to attain its optimum state, which in turn makes the environment more optimal.

Automating this decision-making process is more relevant and, in most cases, produces favorable outcomes since the correctness of the system's decision-making is evaluated in opposition to keeping tabs on the VMs' use status via the use of tools. In about 97% of cases, it fits the decision criteria.

4.3.2 RESOURCE OPTIMIZATION USING PREDICTIVE ANALYTICS

Analysts and researchers are making use of predictive analytics, a new machine learning approach, to forecast a system's future performance by analyzing its historical and present patterns of behavior. Many different statistical models and neural network approaches may be found in the field of predictive analytics. Managing a system effectively also involves learning the system's present behavior over time and utilizing it to forecast it’s near future, in addition to prediction. So, for better decision making, it's best to use both historical system behavior data and

present system state data for prediction. Predictions using the present system state and the (MDP) were covered extensively in the preceding section. This section provides an in-depth analysis of many prediction methods that rely on statistical analysis and neural networks.

Predicting accurately the future resource requirements based on the system's previous behavior and deploying the proper set of resources instead of consuming more resources than necessary is one of the primary optimizations addressed by prediction.

In this section, we will go over a few of the most common methods of prediction, and then we will apply chosen algorithms to forecast our virtual machine needs.

Survey of Prediction Algorithms available in Machine learning

Regression Models

Regression analysis is a sort of predictive modeling that looks at the relationship between a set of independent variables (predictors) and a set of dependent (target) variables (objective). It then fits a model between the two sets of variables to help with future state prediction. For each prediction issue, one may examine one of several available regression approaches, such as linear, logarithmic, exponential, etc. There are a variety of common models available for making predictions, and some examples include logistic regressions, multiple regressions, linear regressions, and others. By using mathematical processes and drawing on past data, the best-fit line or curve may be fitted to the given datasets. The accuracy of the predictions and an error term relative to the input data are common components of the output of most regression methods. The primary objective of every regression model is to find the best fit model in order to minimize error and improve prediction accuracy. When dealing with tiny, non-circular data sets, regression prediction tends to work well. Since regression analysis produces inaccurate predictions when dealing with big data sets, researchers have been exploring other prediction models.

Time series and forecasting – ARIMA

Time series prediction is founded on the premise that long-term data collection may be used to inform mathematical models about the future. This is a list of the most

often used models for analyzing trends and patterns. The time series model's ability to forecast is highly dependent on the degree of correlation between the data. These models are better able to capture information about the source of the issue when the time component is included because they assume that all data points occur at equal intervals and do not take contextual information into account when making predictions. Predicting future observations using models fitted to existing data is the essence of forecasting, of the problems with time series. Most time series have one important feature: as the points in the series go closer together in time, the correlation between them gets stronger but the forecast accuracy decreases as they go more away. Although other time series models exist, ARIMA is by far the most often used (Autoregressive Moving Average).

A wide variety of common temporal patterns may be captured by time series data using the ARIMA model class. ARIMA models the current data to forecast future behavior using data that is both recent and distinct in the past. If you want to build an ARIMA model, you need to make sure your data is completely static, meaning it doesn't trend significantly higher or lower. What this means is that for the process to be stable, the data distribution has to be tightly clustered around the overall mean.

Classification

There are two main types of classification algorithms: supervised and unsupervised. Which of these two approaches is utilized to classify an amalgamation of vectors into well-known classes is dependent on the algorithms being considered. Without training, unsupervised training uses decision-making criteria to classify input data into one of many categories. In contrast, supervised learning uses a data set with pre-defined classes to teach the system to identify all instances of those classes via iterative training. As part of the testing process, unknown data is assigned a class during the classification or prediction phase. Predictions using this classifier type are also possible, with varying degrees of success depending on the specific classifier under consideration.

Neural network

Prediction and categorization are two applications of neural networks. Neural networks learn in a way that is analogous to the human brain, with neuronal strength

determined by weights, as opposed to the conventional statistical approach. Through iterative adjustments to the weights, the error is minimized until saturation is reached. When the mistake is close to zero or very small, it is deemed learned. A neural network learns in a supervised fashion when its weights are changed according to the forecast of which category it will correctly recognize.

As iterations go, Regular adjustments are made to the weights until the inaccuracy is reduced to a minimum. There is an abundance of neural network models for classification and prediction; a few examples include the Hopfield model, multilayer perceptrons, and back propagation. Here, the accuracy of the predictions is dependent on the network's activation and training levels.

For the purpose of predicting virtual machine requirements given task arrival time and task size, this study employs a regression-based statistical method known as ARIMA and a neural network model known as LSTM. The next sections delve into the specifics of these algorithms and the outcomes.

Capacity planning in multi-cloud

It may be challenging to achieve cost benefits in a multi-cloud environment when trying to efficiently manage and optimize cloud resources via capacity planning. The objective is to install and use the right amount of virtual machines in order to optimize resource usage. Predicting the exact amount of virtual machines required to run the resource-constrained tasks is critical for reaching an ideal environment. Based on workload projection and capacity planning, it is necessary to have good (VM) count prediction and management. With several clouds involved, resource and task management becomes more complicated in a multi-cloud environment. Therefore, trustworthy techniques of forecasting are essential for effective capacity planning. In a subsequent section, we will discuss the (LSTM) network, which is used for prediction, however this study assessed several neural network models.

Experiments and Results on VM Capacity Prediction

Predicting the capacity of virtual machines (VMs) required to successfully handle the average task/workload coming daily at various times is an optimization stage job in the framework. In this case, as previously said, ARIMA and Neural Network The

LSTM method is being evaluated for the purpose of predicting the virtual machine capacity required to handle upcoming tasks and workloads. This aids in determining the optimal quantity of virtual machines (VMs) to deploy on that specific day.

ARIMA is useful for projecting short-term requirements using historical data, but it ignores context and operates on the assumption that tasks arrive at regular intervals. Long short-term memory (LSTM) neural networks take context into account, forget some prior data, and let new data in for improved predictions. By combining historical and real-time system data, we can automate provisioning and de-provisioning according to work schedules and better prepare for future capacity demands.

Prediction based on ARIMA

An important aspect of maximizing these resources is predicting the capacity of virtual machines (VMs) and the number of VMs needed to meet the workload in demand. When the number of existing (VMs) is insufficient to meet increased workloads, more VMs must be spun off. When there are too many virtual machines or not enough being utilized, it's vital to reorganize the workload and release the surplus ones. In order to decide how many (VMs) to release or add, it is important to know the demand for VMs over time.

Next, we may use a time series approach to forecasting and prediction to understand this demand. ARIMA is limited to predicting the next immediate task based on past and current data, and it presupposes that tasks arrive at regular intervals. Task length, task intervals, and other contextual variables were disregarded. So, to find a different way to use Neural Networks and increase the accuracy of the predictions, we investigated LSTM neural networks. The section that follows will address this.

Prediction based on LSTM

When dealing with time series sequential data, one form of neural network model that has been used to tackle difficulties is the recurrent neural network (RNN). Applications of RNNs include neural machine translation, speech recognition, and solving problems using sequential time series data. Because of the models' extensive dependencies, the computed gradients are transmitted back and forth several times,

which causes the problem of gradient vanishing and explosion to occur in deep RNN models. In response to these concerns, LSTM RNNs were created. This paradigm uses the "gated RNN" approach to solve long-term reliance problems; it has several successful applications, including speech recognition, machine translation, and image captioning. The capacity of this model to remember and maintain data over extended periods of time is its main selling point, setting it apart from more conventional RNN models. (RNN) data processing is shown in Figure 4.14.

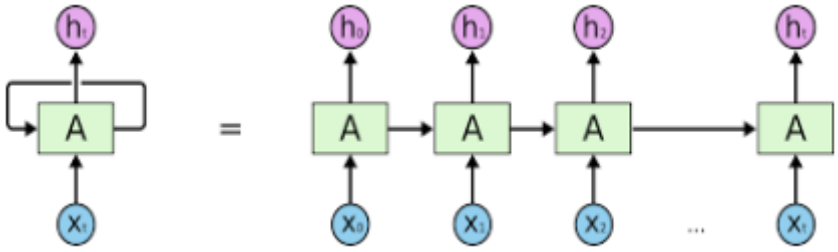


Figure 4. 14 RNN- Recursive data processing

Parts of a long short-term memory (LSTM) include a memory cell ("c"), an input gate ("I"), an output gate ("o"), and a forget gate ("f"). The function of a memory cell is to retrieve and retain information (states) from earlier time series. The data stream that enters the memory cell is controlled by the input gate. "Remind or remember" what data has already been stored in a memory cell is the function of an output gate. The output gate typically controls the bits of data used to calculate the output, often called the hidden state. For each value of X input, the LSTM approach employs a sigmoid layer—also called the forget layer—to calculate the amount of information that has to be erased from the previous cell state. The current module's cell state, C, is used as input to the next repeating module.

The output value between 0 and 1, C(t-1), that the forget layer outputs for each cell state indicates the amount of data to be wiped from the current state. When that is complete, the LSTM network decides how much new data to add to the cell state. The sigmoid and tanh layers make up this. Multiplying and adding the outputs of these two layers to the cell state computed by the forget layer improves its accuracy. In Figure 4.15, we can see the LSTM framework in action, interacting with different components. Figure 4.16 shows the LSTM model's information flow diagram.

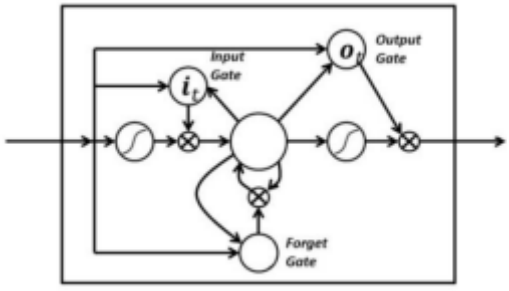


Figure 4. 15 LSTM framework with gate functions

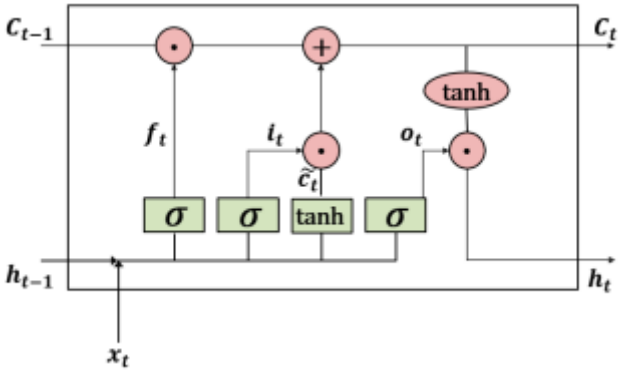


Figure 4. 16 LSTM with logical flow

Finally, every cell has to generate some kind of value. For this goal, an extra sigmoid and tanh layer is used, which generates a cell state that has been demonstrated to be effective. Due to their ability to retain information about an activity's current state, LSTM networks are well-suited for time series forecasting. The following equation may be used to determine how to compute the different values of the current states by using the value of the previous state.

$$f_t = Sigmoid\,(W_f * [\, h_{t-1} * x_t \,] + b_f)$$
$$i_t = Sigmoid\,(W_i * [\, h_{t-1} * x_t \,] + b_i)$$
$$\hat{C_t} = tanh(W_c * [\, h_{t-1} * x_t \,] + b_c)$$
$$C_t = tanh(f_c * \; C_{t-1} + \; i_t \,] + \hat{C_t})$$
$$o_t = Sigmoid(W_o * [\, h_{t-1} * x_t \,] + b_o)$$
$$h = o_t * tanh * C_t$$

In a broader sense, LSTM is applicable to regression together with categorization. In this research, it is used as a regression model to predict when VMs would be required

to manage the expected workloads. The quantity of virtual machines (VMs) and the duration of jobs that came on each day are inputs into this prediction model. Based on this data, the network can predict how many virtual machines will be required down the road. Since LSTM considers context while generating predictions, it beats ARIMA.

Experimental Results

The research tracked the amount of virtual machines (VMs) utilized and the number of tasks coming on a certain day for a duration of 30 days. We employ ARIMA and LSTM with the data we acquire. The following table 4.13 details the number of virtual machines (VMs) required to execute a job throughout the previous 30 days.

TABLE 4. 13 Days VS Number of VMs

Days	VM Count	Days	VM count
1	19	16	21
2	22	17	19
3	24	18	21
4	19	19	22
5	21	20	22
6	18	21	22
7	19	22	22
8	22	23	21
9	23	24	19
10	21	25	21
11	20	26	21
12	20	27	21
13	21	28	23
14	21	29	21
15	22	30	20

By projecting the amount of virtual machines (VMs) needed on the 31st, 32nd, and 33rd days, we can improve environment management. Getting as many virtual machines as possible that are close to reality is the key to getting the best possible outcome. Table 4.13 shows the number of virtual machines (VMs) required for 30 days to fulfill the stated set of planned activities.

Table 4.14 shows the actual and predicted numbers of virtual machines (VMs) deployed from days 31–35, as well as the numbers predicted using ARIMA and LSTM.

TABLE 4. 14 Prediction using ARIMA and LSTM algorithms

Days	Actual VMS	Prediction ARIMA	Prediction LSTM
31	21	20	20.6
32	20	22	21.7
33	22	23	21.8
34	19	22	20.3
35	23	24	22.1

Predictions made by the LSTM model are much superior than those of the ARIMA model because it learns using a recursive neural network and takes contextual information on the task arrival time delay into account. Figure 4.17 displays the outcomes of comparing the prediction performance of LSTM and ARIMA with that of the actual virtual machines (VMs) deployed from day 31 to day 35.

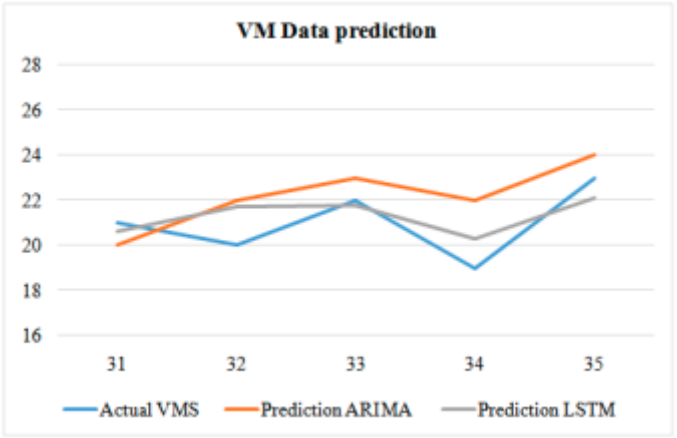


Figure 4. 17 Prediction Comparison across Actual and ARIMA, LSTM

4.3.3 KNOWLEDGE MANAGEMENT

The administration of knowledge is an essential part of every application. Researchers drew on either prior experience with a comparable system or information gleaned from the present system via long-term observation of its operation. Key components of the present research trend include building a knowledge database and utilizing the collected information to make vital decisions using machine learning algorithms. Crucial components include

- A guide on creating a database for managing knowledge.
- How to learn new things and put that information to use in making predictions.

- The best way to assess the findings in light of how to use the information to make the results more accurate.

Numerous algorithms, based on heuristics and determinism, are at your disposal. The first stage involves teaching the system, also known as learning, so that it can learn from experiences with known results. The learning errors become less frequent after a certain amount of repetition. Looking at past data with the intention of predicting future outcomes is what predictive analytics is all about. When it comes to learning and prediction, other sets of algorithms, such the Markov Decision Process (MDP), are used. These algorithms focus on current behavior rather than past conduct. Either approach may be used to choose a course of action for correction, depending on the specifics of the case.

An important aspect of this research is decision-making using data collected from recurrent system interactions. Decisions about resource optimization are informed by the acquired information and are situational appropriate. Below are the main points covered in each area of study?

- determining the necessary capacity to handle the task/workload capacity that is coming for deployment and the active task/workload operating in virtual machines or containers on a regular basis. In this case, capacity management makes advantage of the prediction algorithms. Therefore, in order to forecast the future capacity requirements, the system's historical data is used.

- Using Q-Learning techniques, MDP-RL algorithms make decisions based on system interactions using their state-action pair, and they predict actions based on current states, such as moving containers (workloads) in or out, which relies on the optimal policy derived to achieve the goals.

- Helps build the knowledge base by applying PSO algorithm for VM ranking decisions and AHP for cloud migration workload rankings. The migration judgments made by these algorithms are backed by their prior decisions in comparable scenarios. By combining information about the present with that about the past, optimal judgments may be made for the future.

That being said, our multi-cloud architecture took into account the need of creating

the knowledge base using information acquired from previous scenarios in order to make future judgments.

Optimizing for the cloud is crucial for efficient resource management and creating a perfect environment with high performance and low cost. Management, optimization, and oversight of the cloud environment are so crucial. This study applies optimization on many levels, first by determining the optimal number of virtual machines (VMs) to handle the anticipated workload loads, and then by distributing containers (applications) across those VMs in such a way that they are all equally loaded and perform better. Extensive experiments are conducted to optimize the process of migrating workloads across clouds using containers and virtual machines on the chosen cloud. Machine learning algorithms such as ARIMA, LSTM, and MDP-RL are carefully chosen because they provide optimum use of cloud resources while still producing adequate results. When tested in a multi-cloud setting, the aforementioned ranking procedures significantly improved resource management and operational efficiency.

CHAPTER 5
EXPERIMENTAL RESULTS AND DISCUSSION

Using a five-stage architecture, this study tests and validates it using various situations and settings before comparing its performance to that of competing multi-cloud management systems. Considerations for assessment include RightScalr, clickr, and a handful of other multi-cloud management technologies currently on the market. Additionally, for managing a multi-cloud environment, there are a few of published research publications that touch on comparable topics. Nevertheless, prior studies and similar technologies have failed to provide enough proof to help with ranking the clouds and using knowledge management to make better decisions. The study also makes use of the robust framework to boost computing capacity using technologies like Spark and Kafka together, which aids in decision making and processing speed. To prove and conclude that this methodology is useful in managing multi-cloud environments, this section discusses a few of these cases and their outcomes.

5.1 EXPERIMENTS CONDUCTED WITH DIFFERENT SCENARIOS

Four clouds—Open Stack private cloud, (AWS), Microsoft Azure, and Google—are included in this study's investigation, as mentioned before. Kubernetes, an orchestration platform for cross-cloud service orchestration, and cloud forms, a Red Hat-developed tool for integration and monitoring, link the four clouds. Docker containers take care of every step of the development process, from building to deploying and managing releases. Whether it's scheduling, service discovery, managing resources, or service management, Kubernetes takes care of it all when it comes to operational administration and security. The monitoring tool, which consists of cloud forms, is set up in every participating cloud. This ensures that the resources dispersed across all clouds are always being monitored. As new research parameters are defined, the influx DB database will back them up indefinitely. Each cloud's virtual machines and containers have their data gathered every ten minutes. Three days, each lasting nine to ten hours, are needed to examine the data. In order to guarantee that the research-related framework idea consistently functions in diverse contexts, this study will test and confirm it.

The primary goals of this framework validation and analysis are as follows.

- Managing resource demands (to provide new VMs or release existing ones) via virtual machine consolidation based on resource use and projection.
- Moving containers from one virtual machine (VM) to another, either inside the same cloud or to various clouds, with the help of business-driven cloud selection.
- Task and workload scheduling in accordance with the virtual machine (VM) associated with a chosen cloud.
- Streamlining cloud-based tasks via the use of learning, knowledge management, and categorization algorithms.
- Using a framework and tools to improve calculation performance for decision making.

To verify the aforementioned goals and assess the outcomes, three separate scenarios were developed and are detailed in the section that follows.

5.1.1 TYPES OF VM CONSIDERED IN EACH CLOUD

It was determined to specify each virtual machine's settings (VMs) and containers (Dockers) together with jobs (workloads) so that there would be uniformity in measuring the parameters and process control over the VMs and containers. Table 5.1 details the various experimental sizes and parameter settings.

Table 5. 1 VM parameters considered for experimentation

Container (Docker)	Type 1	Type 2	Type 3
Memory (GB)	8	4	2
Core	2	1	0.5
Task Length (MI)	1000 - 20000		
Total Number of Tasks	20-100		
VM Type	Large	Medium	Small
Core	8	4	2
CPU (MIPS)	2000	1000	500
Memory (GB)	32	16	8
Disk (GB)	850	410	160
Bandwidth (bits)	1000	800	500
PE	2	1	1

For the sake of this research, virtual machines (VMs) and containers are built in each cloud with a size that is near to these parameters. In a multi-cloud setup, the criteria serve as broad recommendations for locating virtual machines (VMs) with comparable attributes in each of the participating clouds. In order to confirm that the method is effective and flexible enough to be used in the future by the industry, In order to ensure that the framework's stages and results were consistent, the experiments were run in two different circumstances.

Scenario 1: All public cloud with Single region

Here, we have a multi-cloud setup where AWS, Azure, and Google Cloud all share a single region. This model was developed for the purpose of experimenting with three different public clouds, as shown in Figure 5.1.

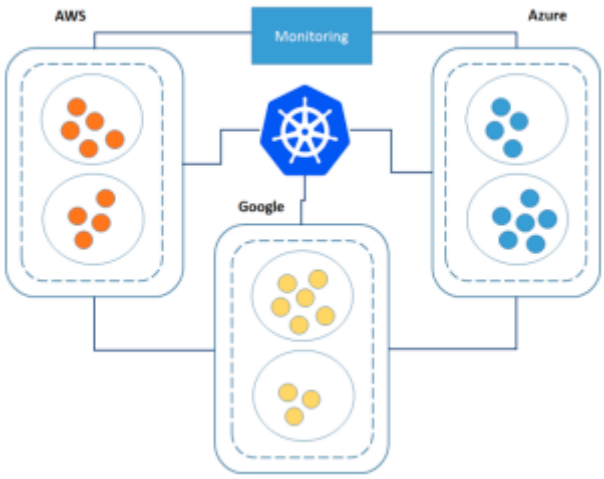


Figure 5. 1 Multi-cloud scenario – All public cloud

Scenario 2: Public cloud and Private cloud

This setup makes use of four different types of clouds: three public ones (Google, Azure, and AWS) and one private one (Open stack). This time around, we used the same set of tests to prove that our multi-cloud management solution worked. After analyzing and discussing the pertinent facts at each level, the following are the results that were taken into account.

Verifying that the same set of methods also works in a larger cloud setting will demonstrate the procedures' generalizability. Figuring out the four-cloud situation, we

see that it consists of three public clouds and one private cloud.

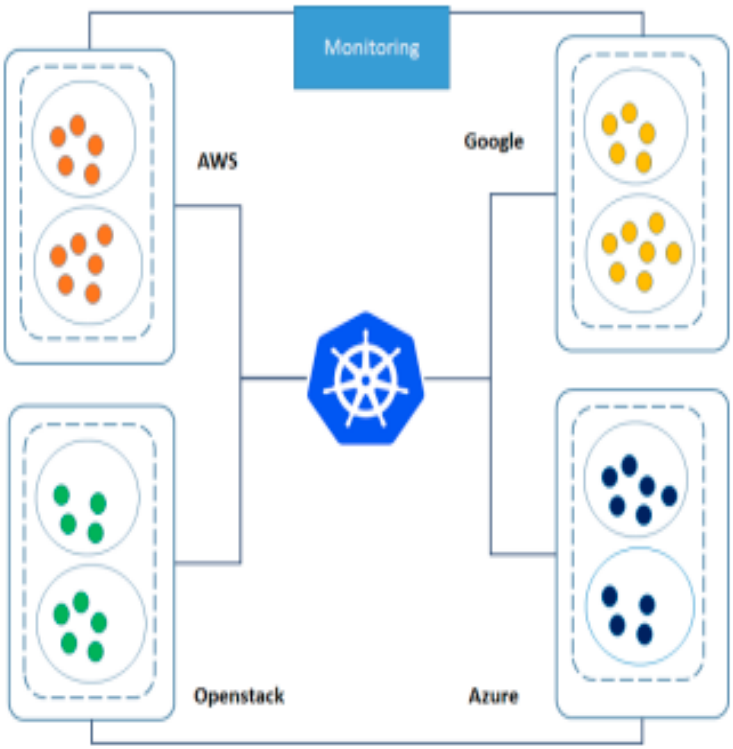


Figure 5. 2 Multi-cloud scenario - Three public and a private cloud

For each step in the two cases described above, we ran experiments and analyzed the data to make sure the framework we built consistently produced the desired outcomes, making it an effective tool for managing multi-cloud environments.

Stage 1 - Global ranking (selecting right cloud): After rating the clouds using the AHP algorithm, the one with the best set of criteria was chosen for future deployment. Here, we take a look at AWS, Azure, and Google Cloud, all of which are public clouds. As business metrics, the following are prioritized: cost (per virtual machine per month average), utilization (%), network latency (mSec), reaction time (mSec), and availability (in %).

Scenario 1: Three clouds – all public Clouds.

In order to rate the multi-cloud environment, AHP takes the following factors into account. The goal here is to use business criteria for analysis and decision making to choose the best cloud to deploy new jobs and workloads. You can see the values of the AHP-related business parameters in the table.

Table 5. 2 Business parameters for AHP in three cloud scenario

Cloud	Cost $	Availability %	Net.Lat (mS)	Res.Time	Utilization %
Aws	28.2	98.7	278	168	83
Azure	29.4	97.6	257	216	89
Google	27.9	95.2	289	188	78

The ranking shown in Table 5.4 is the result of running the AHP algorithm with these criteria.

The Priority Index Criteria are shown in table 5.3.

Table 5. 3 Business priority Index for three cloud scenario

	Cost	Net.Lat	Availability	Res.Time	Utilization
Cost	1	7	3	2	9
Net.Lat	1.143	1	1	1	8
Availability	0.333	1	1	3	7
Res Time	0.5	1	0.333	1	5
Utilization	0.111	0.125	0.143	0.2	1

Table 5.4 displays the final ranking of the three clouds as determined by the AHP algorithm.

Table 5. 4 Ranking identified through AHP for 3 cloud scenario

	V1
Gcp	0.6716804
Aws	0.2277760
Azure	0.10055435

Google Cloud Platform (GCP) is definitely the top choice for more research according to the ranking method, which considered the selected attributes that are significant to the venture.

Scenario 2 – Four clouds: Three public and one private cloud.

Here, we analyzed using the same AHP method and criterion set, but with the business priorities set differently. The ranking procedure is explained by the following outcomes.

Important metrics calculated for every cloud, such as monthly virtual machine cost, availability, network latency, response time, and utilization. The business requirements that were determined for four different cloud scenarios are shown in Table 5.5.

Table 5. 5 Business Parameters value for four cloud scenario

Cloud	Cost $	Availability %	Net.Lat(mS)	Res.Time (Sec)	Utilization %
Aws	28.2	98.7	278	168	83
Azure	29.4	97.6	257	216	89
Google	27.9	95.2	289	188	78
Open Stack	31.4	99.3	117	102	83

Table 5.6 shows the business importance for each criterion.

Table 5. 6 Business priority index for four cloud scenario

	Cost	Net.Lat	Availability	Res.Time	Utilization
Cost	1	5	3	9	8
Net.Lat	0.2	1	1	2	6
Availability	0.333	1	1	3	9
Res Time	0.111	0.5	0.333	1	7
Utilization	0.125	0.167	0.111	0.143	1

Table 5.7 displays the final ranking of clouds as determined by the AHP algorithm.

Table 5. 7 Final ranking using AHP Cloud for four cloud scenario

	V1
Gcp	0.43914459
Aws	0.37578273
Azure	0.12530775
Open stack	0.0596493

After running the chosen cloud through the AHP rating process with all business factors taken into account, when considering future deployments, Google Cloud Platform (GCP) stands out as the obvious winner.

Based on the given set of business needs, AHP makes it quite obvious that Google Cloud is the best choice for doing more analysis and deploying workloads in both scenarios.

Stage 2 – Local ranking (VM within cloud): This analytical phase seeks to determine which virtual machines (VMs) are most suited for the deployment of new workloads. The PSO-SVM approach uses a number of parameters to choose the candidate virtual machines, including task execution time, task execution cost, and VM power.

Detailed below are the results that indicate which virtual machines are most suited to launch the jobs that are currently in the deployment pipeline. To determine the VM rankings, a PSO technique was used, which incorporates three objective functions: power, time, and cost. Initially, the SVM classifier could be used to assign new workloads to the group of VMs that were not in use very often.

Scenario 1: Three clouds – all public clouds.

Here, we'll use support vector machine (SVM) categorization based on CPU and memory to identify the top five virtual machines (VMs) from the group of "under-utilized" VMs.

Table 5.8 shows the top five virtual machines' prices, speeds, and IOPS and bandwidth.

Table 5. 8 VM parameter with attribute values for local ranking

VM	Price	SPEED	Iops	Bandwidth Cost
VM1	0.148	2000	250	0.024
VM2	0.32	3350	210	0.019
VM3	0.25	3000	230	0.02
VM4	0.2	2500	300	0.029
VM5	0.29	3250	370	0.02

Next, we need to figure out how to assign these five VMS to the ten tasks that are now waiting to be deployed, measured in MIPS. For a better understanding of the work deployment in the top 5 VMs, please refer to the accompanying table5.9 and graphic.

Table 5. 9 Ask and its length to be deployed in VMS (in MIPS)

Task	Length
t1	3200
t2	1952
t3	6293
t4	3750
t5	5576
t6	4165
t7	6840
t8	4513
t9	4000
t10	3500

Task scheduling and deployment are shown in Figure 5.3, which is based on the algorithm's identification of the VM's rank.

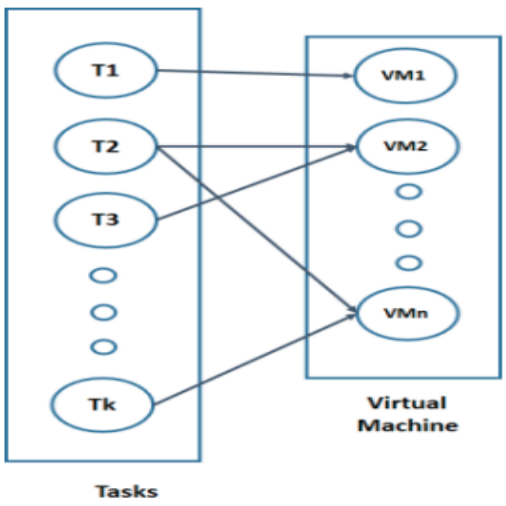


Figure 5. 3 Scheduling of Tasks with VMs

Prioritization of VMs using PSO techniques' objective functions.

Table 5. 10 VM ranking using PSO objective functions

Task / VM	VM1	VM2	VM3	VM4	VM5
t1	4	1	2	5	3
t2	3	1	2	5	4
t3	4	3	2	5	1
t4	4	1	2	5	3
t5	4	3	2	5	1
t6	4	3	1	5	2
t7	4	3	2	5	1
t8	4	3	2	5	1
t9	4	2	1	5	3
t10	4	1	2	5	3

According to the table above, Using PSO multi objective, you should optimize the objective function by averaging its weights and then rank them from lowest to highest. This will allow us to determine which virtual machines are most suited for each work. As a result, the highest-ranked virtual machine (VM) is chosen for scheduling and subsequent execution of each job. Rather of waiting for the first virtual machine to finish executing task1, the second assignment is set to be released to the next available virtual machine if the same VM is granted high priority for both tasks. Virtual machines (VMs) hosted by Google Cloud are used for all of these analyses since, according to the AHP algorithm, they were rated first for this specific scenario.

Scenario 2 – Four clouds: Three public and one private.

Here, GCP was used as the rated cloud to conduct the ranking of ten virtual machines (VMs) each with twenty jobs. The SVM classifier used the following 10 virtual machines (VMs) as inputs, as shown in Table 5.11.

Table 5. 11 VM parameters with attribute value for four cloud scenario

VM	Price	SPEED	Iops	Bandwidth Cost
VM1	0.32	3350	210	0.019
VM2	0.148	2000	250	0.024
VM3	0.33	3450	218	0.021
VM4	0.31	3300	215	0.0198
VM5	0.3	3280	300	0.023
VM6	0.28	3200	350	0.0198
VM7	0.36	3400	220	0.022
VM8	0.28	2980	320	0.025
VM9	0.297	3500	350	0.0205
VM10	0.29	3250	370	0.02

The top 20 jobs that are now awaiting deployment are shown in table 5.12 in MIPS.

Table 5. 12 Tasks length in MIPS for four cloud scenario

Task	Length (MI)	Task	Length (MI)
T1	3200	T11	4521
T2	1952	T12	3010
T3	6293	T13	4100
T4	3750	T14	2200
T5	5576	T15	2025
T6	4165	T16	3350
T7	6840	T17	6500
T8	4513	T18	5800
T9	4020	T19	1630
T10	5980	T20	3914

Table 5.13 shows the results of assigning the 20 jobs to virtual machines (VMs) using a rating mechanism. The scheduling procedure takes this rating into account when

selecting the appropriate virtual machine (VM) to deploy the job under consideration. The scheduler's task waiting deployment rankings for virtual machines found using the PSO technique are shown in Table 5.13.

Table 5. 13 VM Ranking of 10 VMs for 20 tasks in four cloud scenario

Task / **VM**	**VM 1**	**VM 2**	**VM 3**	**VM 4**	**VM 5**	**VM 6**	**VM 7**	**VM 8**	**VM 9**	**VM11 0**
t1	2	9	3	1	8	4	5	10	7	6
t2	1	6	3	2	7	5	4	10	9	8
t3	6	10	8	5	4	1	9	7	3	2
t4	2	9	4	1	8	3	7	10	6	5
t5	6	10	7	5	4	1	8	9	3	2
t6	4	9	6	2	8	1	7	10	5	3
t7	7	10	8	6	4	2	9	5	3	1
t8	5	9	6	4	7	1	8	10	3	2
t9	3	9	6	1	8	2	7	10	5	4
t10	6	10	7	5	4	1	9	8	3	2
t11	5	8	6	4	7	1	8	10	3	2
t12	2	8	3	1	8	5	4	10	7	6
t13	3	8	6	2	8	1	7	10	5	4
t14	1	6	3	2	9	5	4	10	8	7
t15	1	6	3	2	8	5	4	10	9	7
t16	2	9	3	1	8	4	5	10	7	6

t17	7	10	8	5	4	2	9	6	3	1
t18	6	10	7	5	4	1	9	8	3	2
t19	1	5	3	2	7	6	4	10	9	8
t20	2	9	5	1	8	3	7	10	6	4

The chosen essential virtual machines are prepared to launch tasks. Each work is assigned to the virtual machine (VM) with the highest ranking. The next best rated virtual machine is utilized for deployment if the one ranked 1 is currently in use by another job.

In each case, the available virtual machines (VMs) are ranked according to their performance using a PSO-based multi objective function, and the tasks are then assigned accordingly. This is useful for selecting virtual machines (VMs) for a certain task's deployment in the chosen cloud according to objective criteria. The next optimization step would include moving current workloads to different virtual machines (VMs) after Workloads that had been queued up were all eventually deployed.

Stage 3: Container consolidation and optimization– MDP-RL methods

Our goal is to ensure that all of the virtual machines (VMs) are functioning correctly once we assign new workloads to them by redistributing and balancing their workloads.

Using the model-free Q-Table and Q-Learning method, we find the optimal strategy to reach this state (all VMs in Normal state). After a few hundred iterations, we have the last Q-Table representing the current status of VMs with two separate policies.

When deciding whether to go in, out, or take no action at all, the best course of action is the one that has the shortest route to the goal. The settings are causing Q-Learning to converge after around 250 iterations.

The optimum outcome for the given situation utilizing the MDP-RL approach is explained in the following Q-Table.

Scenario 1: Three Clouds – All public clouds

Here, all the virtual machines (VMs) in a certain cloud (like Google Cloud) are moved from an over-utilized to an under-utilized condition using SVM classification. The concept-based state-action pair is completed using the Q-table. Every single cell in the Q-table was set to the "0" state at the beginning. By engaging in continuous dialogue with the system, it is able to learn from its successes (e.g., the choice to remove VMs from an overloaded state) and its failures (e.g., the decision to add VMs to the overloaded state) via the use of reinforcement learning (RL) methods. The algorithm learns and updates the Q-Table every time with the right rewards and penalties. The process repeats until all virtual machines are in the "Normal" or "optimized" condition. Optimal policy is the course of action that will bring about the desired outcome. If you want to go where you're going, you need to make choices that put you on the best possible policy route.

After more than three hundred and thirty iterations, the ultimate Q-Table value is shown in table5.14, which also provides two ideal policies for more investigation. In this case, the system learns by constant interaction, and getting to the ideal state is done by making the right decisions with the knowledge we have towards our goal.

It is standard practice to initialize the Q-Table to "0" for all values before running the analysis.

Both the incentive and the punishment are set to "+2" and "-1" respectively. For each iteration, the Q-value is updated using the following formula, which takes into account the previous value.

$$Q[s, a] = Q[s, a] + alpha*(R + gamma*Max[Q(s', A)] - Q[s, a])$$

Due to the intrinsic randomness of the system, the Q-Value is updated continually using the RL technique via ongoing interaction.

Finding the optimal strategy to reach 100 is the ultimate goal after much iteration.

Consistent reinforcement and correction from its users teaches the system new skills.

Table 5.14 displays the final Q-Table value after the program has completed its aim.

Table 5. 14 Final Q-Table values after reaching the goal for three cloud

State-Action	Under-utilized	Over-utilized	Un-utilized	Fully-utilized	Normal
Move-in	38	47	42	62	69
Move-out	42	58	65	72	89
Provision-VM	34	49	76	79	59
De-provision-	24	53	79	87	77
No-Action	19	34	84	94	100

Two policies have been reached; the best policy is the one that maximizes reward and accomplishes the "goal of 100" first. When all virtual machines (VMs) have reached saturation and are in a "Normal" state, the system is stable and completely "Optimized," and the decision-making process for "moving-out" or "moving-in" containers from VMs follows this ideal strategy.

Scenario 2: Four Clouds – 3 public and 1 private clouds

We repeated the process in Scenerio-2, this time experimenting with the effects of a private cloud on our analysis. In the startup process, the Q-Table is initialized to all "0" values. The ultimate Q-Value approaches saturation and the end aim in table 5.15 after a few hundred cycles. We identify the optimum policies that meet the aim. As previously said, the objective was to get 100 in order to bring the system into the "No-Action - Normal" state, which is deemed normal under these circumstances. Table 5.15 displays the final Q-Table value after hundreds of repetitions for each of the four cloud situations, indicating when the objective is attained.

Table 5. 15 Final Q-Table values after reaching the goal for four cloud scenario

State-Action	Under-utilized	Over-utilized	Un-utilized	Fully-utilized	Normal
Move-in	29	38	42	37	59
Move-out	38	42	53	72	89
Provision-VM	42	51	69	79	94
De-provision-	18	43	37	87	77
No-Action	46	24	44	94	100

The MDP-RL method is useful in both cases because it provides insight into the system's behavior and allows for continuous involvement, with the system being regularly rewarded or penalized for progress. It is common to have to go through

hundreds of iterations since the system won't converge. Repetition of the exercise with modified learning settings is required to reach the target (alpha, beta, and others).

Even if all of the policies aiming at the target achieve 100%, there may be those that fall short. The ideal policy is the one that maximizes gain and reaches 100. Even if the system doesn't converge and hit 100 every time, it will still attain saturation after hundreds of iterations if the Q-Value doesn't change. This is not always required or even feasible, but it does happen. It is possible to choose the course of action that will have the most impact on policy by making predictions based on the knowledge gained.

It is possible for MDP-RL to fail to converge due to its computing demands. However, convergence is achievable with careful parameter selection. The convergence of a system depends on its aims and objectives, and a well-designed system with incentives and penalties will provide superior results.

Stage-4: Prediction of VMs based on task arrival – VM count optimization.

At this point, It's important to check whether the virtual machines are running at full or partial capacity after moving certain jobs from the queue and redistributing them using container migration.

- In order to optimize the VM count, it is necessary to eliminate any virtual machines that are still in the empty state if the system is operating at over capacity.
- New virtual machines (VMs) need to be supplied in order to handle the newly waiting tasks and workloads if the system is operating below capacity, which indicates that there are more workloads in the queue that need to be deployed.

For this reason, we validate the simulated situation's required virtual machine capacity using predictive analytics that combine the Arima statistical technique with the LSTM approach. Then, we deploy the virtual machines according to our predictions, taking into account task arrival and operating capacity (CPU, memory, and utilization).

Scenario 1: Three Clouds – All public clouds

The scenario involves using ARIMA and LSTM algorithms for VM prediction. For a

duration of 30 days, the amount of virtual machines (VMs) required according to the workload is monitored daily. Both models are used to forecast the virtual machine needs for the following five days; LSTM outperforms ARIMA in terms of prediction accuracy from Day 31 to Day 35 as anticipated. When making predictions, the job lengths for each day are taken into account as well. For three different cloud scenarios, the LSTM algorithm's VM prediction is shown in Figure 5.4.

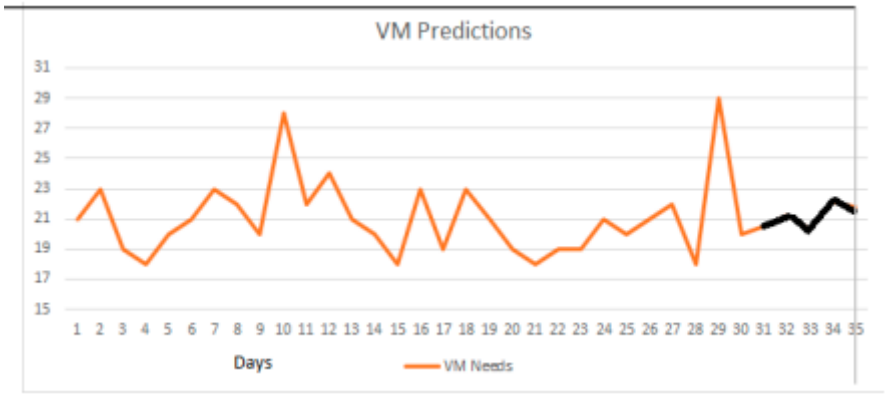


Figure 5. 4 Prediction of VM with LSTM algorithm for 3 clouds

Scenario 2: Four Clouds – Three public and one private cloud

Here, we use ARIMA and LSTM to make the same VM prediction. For 30 days straight, we measure the work volume to determine the number of virtual machines required. The results are shown on the following page. For the following five days (Day 31–Day 35), the VM needs are anticipated using both models. The results of the VM prediction using the LSTM algorithm for four different cloud scenarios are shown in Figure 5.5.

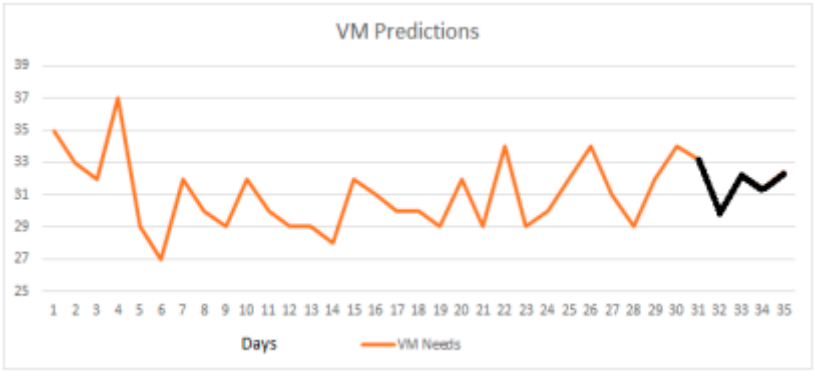


Figure 5. 5 Prediction of VM with LSTM algorithm for four clouds.

Figure 5.6 shows the results of comparing the LSTM and ARIMA algorithms' prediction accuracy; the results show that, for all three cloud scenarios, the LSTM algorithm's incorporation of contextual information improves its prediction accuracy.

For four cloud scenarios, Figure 5.7 shows that LSTM outperforms ARIMA in terms of prediction accuracy. This is due to the fact that LSTM incorporates contextual information into its learning process.

Table 5. 16 VM capacity prediction

Days	Scene-1 VM needs ARIMA	LSTM	Scene-2 VM needs ARIMA	LSTM
1	21		35	
2	23		33	
3	19		32	
4	18		37	
5	20		29	
6	21		27	
7	23		32	
8	22		30	
9	20		29	
10	28		32	
11	22		30	
12	24		29	
13	21		29	
14	20		28	
15	18		32	
16	23		31	
17	19		30	
18	23		30	
19	21		29	
20	19		32	
21	18		29	
22	19		34	
23	19		29	
24	21		30	
25	20		32	
26	21		34	
27	22		31	
28	18		29	
29	19		32	
30	29		34	
31	21.4	20.5	32.4	33.2
32	22.1	21.2	33.2	29.9
33	20.8	20.3	34.2	32.3

34	22.1	22.1	29.5	30.4
35	21.8	21.8	33.4	32.4

Table 5. 17 Task capacity prediction

Days	Scene-1	Scene-2
	Tasks len	Tasks len
1	1082	2082
2	1258	3248
3	1598	5593
4	1832	4832
5	1398	6398
6	1654	4654
7	1476	5476
8	4378	4878
9	3874	3894
10	3846	3346
11	2834	3832
12	1832	4842
13	3723	3923
14	2386	3386
15	1482	3482
16	1962	4942
17	4276	5276
18	2956	6952
19	3232	4232
20	1865	4835
21	2976	3936

22	4365	4761
23	3876	3376
24	2763	3793
25	2643	4649
26	2897	3897
27	3198	2158
28	2864	4869
29	4532	4738
30	2398	4392
31	3243	5782
32	3874	4972
33	4126	4286
34	2984	4676
35	3462	5128

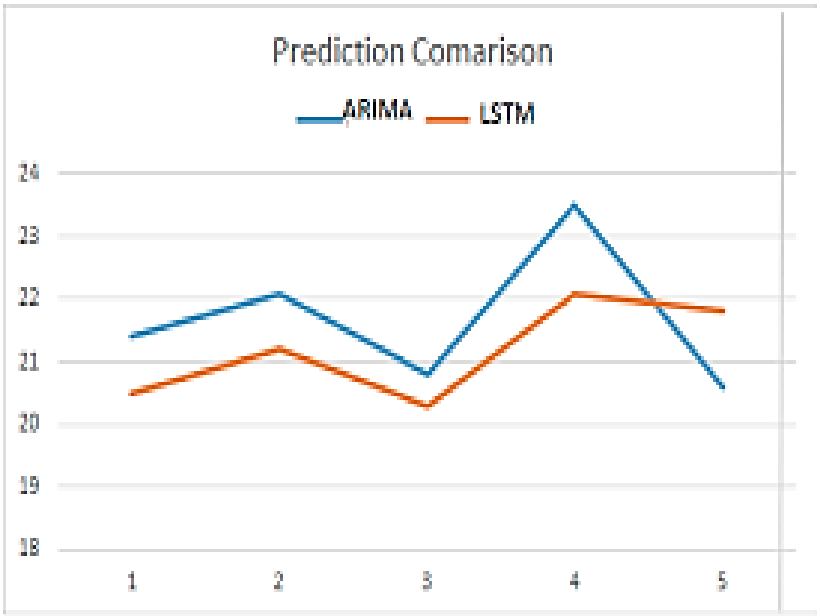


Figure 5. 6 Prediction comparison between LSTM and ARIMA for 3 cloud scenario

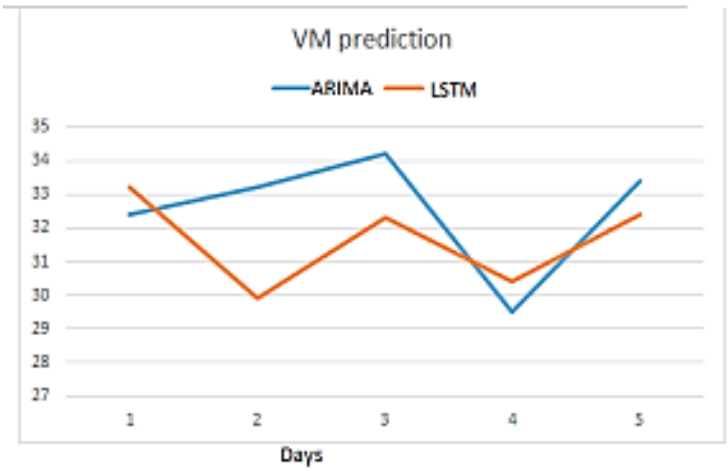


Figure 5. 7 Prediction comparison between LSTM and ARIMA for Four cloud scenario

Because Google Cloud was chosen for study based on AHP rating, the VM required for Google Cloud is predicted in both scenarios. Given that LSTM takes contextual information like task duration and time gap into account, its prediction accuracy is somewhat higher than ARIMA. All tasks are anticipated to be completed at regular intervals in the ARIMA model. When compared with the actual computations made that day based on capacity, the extra knowledge of the time gap for LSTM when considering forget states led to superior predictions. When comparing LSTM to ARIMA, the former has a lower prediction error.

Stage 5: Knowledge management based on previous history

Algorithms like MDP-RL, PSO-SVM, Prediction with Neural Network, etc.—used for optimization and validation—use long-term system observations to find things like task arrival time, peak and non-peak hours, time delay between task arrival, average number of virtual machines (VMs) needed, average VM failure rate, VM performance, and so on. Participation in the system on an ongoing basis accumulates data that may be used to validate results and decisions.

Prescriptive Knowledge

- The quantity of virtual machines (VMs) required according to workloads (seen and predicted daily)
- The daily pattern of CPU and memory utilization in each cloud

- The ranking of clouds for certain deployment types
- Consistency — Cloud-specific policies and procedures for every area.
- The availability and cost of resources for every cloud

Descriptive Knowledge

- The results of PSO-SVM and other machine learning methods.
- The way the system behaves when presented with a state, as learned using MDP-RL.
- Context-aware neural network capacity prediction for virtual machines.
- Organizing and allocating workloads according to each cloud's specific requirements in terms of reaction time, availability, SLA, network latency, etc.

The decision-making knowledge management structure, shown in figure 5.7, is based on historical data. All the prior information in relation to the various factors stated in Figure 5.8 is used by this knowledge management structure.

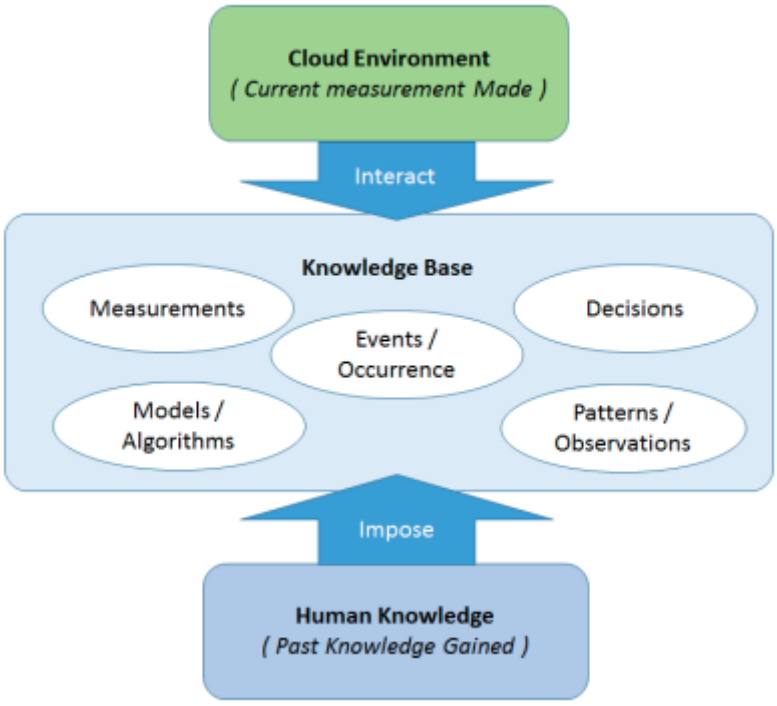


Figure 5. 8 Knowledge management framework

A foundation of knowledge has been established at this stage of the framework via user observations and decisions, in addition to interactions with cloud environments.

This database has been properly archived and indexed in order to facilitate validation and cross-verification. The information repository stores aggregated data that is both descriptive and prescriptive in nature, gathered over time. Machine learning algorithms correctly store and index all data, including decisions, measurements, patterns seen over time, etc. The judgments made based on the existing status of the system may be further substantiated and their correctness improved by consulting this knowledge base.

5.2 VALIDATION OF RESULTS, ANALYSIS AND INFERENCES

Aside from the general framework scenario validation, there are a few of experiments that provide light on the efficacy of specific machine learning methods used for analysis, such as SVM classification, ARIMA and LSTM prediction, and MDP-RL optimization. What follows is a discussion of a few of the results and conclusions.

5.2.1 CLASSIFICATION ACCURACY

We looked at the train and test data sets, three use types, and the accuracy of SVM-based VM classification using the confusion matrix, among other things. Here, 100 virtual machines (VMs) are grouped into three cases: over-utilized, typical, and under-utilized. The SVM classification mapping that comes from the "R" framework is shown in Figure 5.9. In Figure 5.10, The SVM classifier's cluster mean is visible to us.

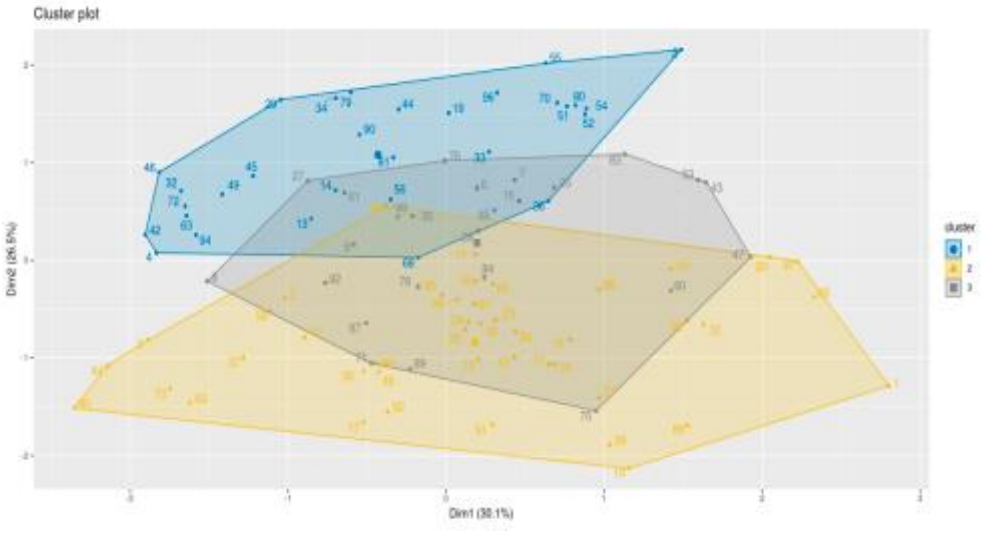


Figure 5. 9 SVM classification for VMs - thematic map

	CPU	Memory	Utilization	Power
Cluster 1	49.79494	1192.741	48.07718	104.8349
Cluster 2	52.64460	1856.902	46.46752	106.6957
Cluster 3	46.44009	1527.151	45.68213	107.9334

Figure 5. 10 Cluster mean arrived through SVM classifier

Train 75 % and Test 25%

Our data set consists of 75 training points and 25 testing points, for a total of 100 points. The classification findings shown in Figure 5.11 are based on the confusion matrix. Classification accuracy levels are 92%. Using the virtual machine's (VM) statistics on CPU, memory, usage, and power, the support vector machine (SVM) classifier is trained and classified. We chose these data points because they are typical VM attributes and may effectively categorize VMs of any kind in any cloud environment. The goal is to find virtual machines that aren't getting enough use by taking into account not only their utilization level but also other metrics like power, CPU, and memory. The optimization and deployment processes will benefit from this. The SVM algorithm, when run via the "R" analysis tool, produced the confusion matrix, as shown in Figure 5.11.

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1  4  1  0
         2  0 13  1
         3  0  0  6

Overall Statistics

               Accuracy : 0.92
                 95% CI : (0.7397, 0.9902)
    No Information Rate : 0.56
    P-Value [Acc > NIR] : 0.0001043

                  Kappa : 0.8638
 Mcnemar's Test P-Value : NA
```

Figure 5. 11 Confusion matrix and classification accuracy through SVM classifier

Using additional training samples clearly improves classification accuracy, in order to conduct further analyses and enhancements using the MDP-RL methodology.

5.2.2 PREDICTION ACCURACY AND EFFECTIVENESS

Optimizing the administration of a multi-cloud system includes capacity management, among other critical components. Therefore, it is essential to predict the capacity of virtual machines by taking into account the workloads that need to be handled and the amount of workloads that are anticipated to be delivered. An accurate forecast will aid in capacity planning, provisioning, and readiness to handle workload demands. Finding the best prediction model that is both accurate and realistic is, thus, crucial. To evaluate the forecast accuracy and address the needed VM capacity, this forecast is comprised on the statistical ARIMA method as well as the Neural Network - LSTM techniques. In order to test this on the daily needs of the VMs over time, several time series and forecasting techniques were used. One such approach was LSTM, which uses contextual information such as task waiting times and time lag to create much more accurate forecasts.

5.2.3 END-TO-END MANAGEMENT OF MULTI-CLOUD SCENARIO

Results from experiments comparing a three-cloud setup with three public clouds versus a four-cloud setup with three public clouds and one private cloud are shown in table 5.18. We provide the results and efficacy of each level of the framework so you may compare them.

Table 5. 18 Comparison of results with two scenarios

Algorithm outcome	Scenario 1 : 3 cloud – all public	Scenario 2: 4 cloud – One Private, Three Public
Ranking of cloud using AHP	Google cloud	Google cloud
Ranking of VMs using PSO-SVM	Top 6 VMs are selected	Top 9 VMs are selected
Scheduling task to VM	9 task are scheduled in 5 VMs	15 tasks are scheduled in 9 VMs
Capacity Prediction using ARIMA/LSTM for VMs	LSTM prediction is higher	LSTM prediction is higher
MDP-RL for container migration and optimization	Took more than 300 iterations to converge	Took more than 500 iterations to converge
Use of Knowledge management Repository	Previous Knowledge was useful	Previous Knowledge was useful

5.3 LARGE DATA MANAGEMENT WITH SUPPORTING TOOLS

Cloud monitoring and data collecting are commonplace in this study. With massive amounts of data being acquired at regular intervals and the need to analyze and make decisions more often, a system that can compute quickly and speed up the process is important. The superior design of SPARK (an in-memory database) for querying and processing sets it apart from all conventional systems, as several engineers and researchers have shown and put into practice. To handle numerous activities at once with available resources, this research effort uses Spark in combination with Kafka, a cluster-based message management system, to handle several requests in parallel.

5.3.1 SPARK AND KAFKA FRAMEWORK

Spark allows for high throughput of real-time data streams, scalability, and fault tolerance via its application programming interface (API). Many different sources, like Kafka, may feed data into conventional data storage, which can then process the data using sophisticated algorithms like MDP, PSO, SVM, etc. One of Spark's primary data structures, Resilient Distributed Datasets (RDD) use logical divisions to organize datasets. Distributed and calculated on various nodes in the cluster, these objects are immutable.

Kafka is the backbone of real-time data streaming and a communications and integration platform for Spark streaming. Data analysis employing complex algorithms is commonplace in Spark Streaming. When finished, the results may be shared in yet another Kafka topic, saved in HDFS, databases, or shown on dashboards. In this case, the database is just used for (MySQL) analysis.

5.3.2 EXPERIMENTS AND RESULTS

In order to expedite analysis and decision making, this study investigated how Kafka and Spark may be used for asynchronous I/O processing, in-memory data processing, and data streaming. Machine learning and neural networks do this by generating decisions based on complicated calculations and massive amounts of data, both historical and in real-time. Figure 5.12 illustrates the end-to-end data flow for both the training data set and real-time data analysis, and it integrates components like Kafka without Spark. We compare the results with a typical setup on virtual machines and

with Spark and Kafka working together.

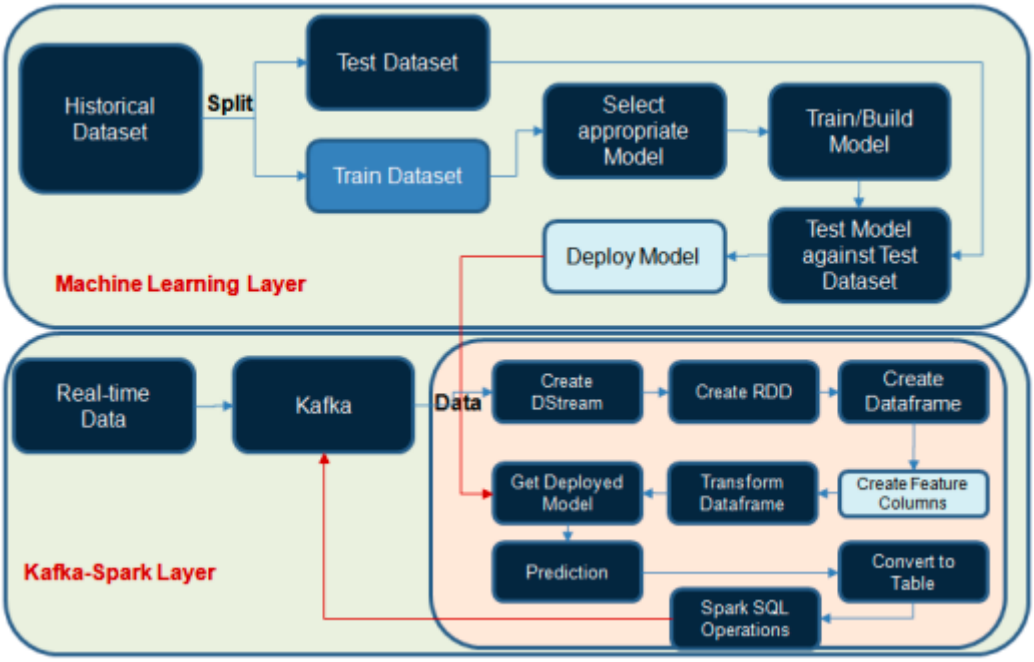


Figure 5. 12 Framework for Kafka-spark computation

Data volume taken into account for experiments as a whole. Here, data is gathered from all four clouds every 30 seconds throughout the course of 12 hours from various cloud settings utilizing monitoring tools.

Here we can see the total amount of rows of data that were gathered. With the Kafka-Spark system, the quantity of rows utilized for calculation is shown in Table 5.19.

Table 5. 19 Records used for each cloud for computation

Records	Numbers (Rows)
Number of Records for AWS VMs	93,854
Number of Records for Azure VMs	1,13,593
Number of Records for Google VMs	87,345
Number of Records for Open Stack VMs	1,19,239
Number of Container related datas (AWS)	2,45,653
Number of Container related datas (Azure)	3,45,642
Number of Container related datas (Google)	2,87,325
Number of Container related datas (Open Stack)	1,98,678
Total Database - GB Size	6.3 GB data

See Table 5.20 for a visual representation of the data processing speeds compared to the algorithms taken into account at various stages. Results show that as compared to the conventional method, the Spark-Kafka combo significantly accelerates data processing, with even complicated algorithms, such as LSTM and MDP, producing much faster output. You can see the difference between the old method and the new Kafka-spark framework in terms of computing speed in Table 5.20.

Table 5. 20 Comparison of computational performance

Algorithms	**Traditional (8Core, 4 VCPU) in (Sec)**	**Kafka - Spark Cluster in (Sec)**
Parameter selection using PSO	2.349	0.0334
Classification using SVM	1.86	0.0278
AHP Ranking	2.59	0.0831
MDP-RL Processing	6.98	1.035
LSTM Prediction	4.98	0.953

In a multi-cloud environment where the system is inherently unpredictable and ever-changing, the use of Kafka-Spark as a framework is clearly enhancing computational speed and facilitating rapid decision-making.

This study's multi-cloud architecture was tested in several environments, verified against a real-world situation, and shown to be successful in handling the majority of operational concerns.

Additionally, when compared to competing products, our effort outperforms them in terms of ranking and optimization. The goal of establishing an optimum environment for ease of administration and cost efficiency in multi-cloud was met by the tests done in diverse scenarios.

The main features that set this framework apart are its ability to be easily transported across multiple cloud environments utilizing containers, its ability to use prior knowledge in decision-making, and its unique components that increase computation speed.

In order to solve the real-time challenge with better methods in the future, this study will be ongoing to enhance and fine-tune the framework.

CHAPTER 6
CONCLUSION, RECOMMENDATION AND FUTURE SCOPE

6.1 CONCLUSION

To take advantage of the advantages offered by various cloud providers, multi-cloud solutions have become the standard in today's IT infrastructure environment. These approaches provide more adaptability, robustness, and freedom from vendor lock-in, but they also bring new, difficult difficulties that must be carefully managed. By looking at these problems from the perspective of end-to-end operations management, we can learn a lot about how to handle multi-cloud strategies well and find out which approaches are necessary for overall management.

Integrating different systems and services is a complicated process, which adds another layer of difficulty to operating a multi-cloud setup. Various cloud platforms have distinct application programming interfaces (APIs), data formats, and security standards that organizations need to negotiate. Due to the complexity, a solid management structure is required to coordinate and integrate cloud resources without any hitches. Methods like unified management consoles and cloud management platforms (CMPs) are vital in fixing this problem. Companies may simplify their operations, enforce regulations, and get insight into all of their cloud environments with the help of CMPs, which give centralized management. Businesses may save time and effort managing different cloud services and be confident that everything is running smoothly and legally by using these platforms.

Ensuring uniform security and compliance across many cloud environments is another major obstacle. Inadequate management of the security protocols and compliance standards of individual cloud providers may lead to security holes. There must be a unified security architecture put in place for all cloud environments. To achieve this goal, methods like unified threat detection and response capabilities offered by multi-cloud security systems must be implemented. To top it all off, businesses need to set up uniform security policies and procedures to make sure all their cloud services are up to code. This way, companies can keep their data and apps protected from security attacks while still staying in compliance.

Careful attention is also required for cost control, another key component of multi-cloud operations. When dealing with several providers that have different price systems, the unexpected expenses might arise from the pay-as-you-go approach of cloud services. Budgeting and other cost optimization tools in the cloud are two of the most important components of effective cost management. By revealing spending habits and use trends, cloud cost management systems help businesses find and fix inefficiencies. In addition, expenditure should be in line with corporate objectives and should not be frivolous; this may be achieved via the use of cost allocation and budgeting procedures. Companies may get the most out of their cloud investments and save money by using these methods.

Optimizing performance in a multi-cloud setting calls for a sophisticated strategy for dividing up resources and tasks. Optimizing resource consumption across different cloud providers' platforms may be a challenge due to their differing performance characteristics. When it comes to fixing performance problems, methods like auto-scaling and workload relocation really shine. Depending on performance requirements and budget constraints, workload migration entails systematically migrating data and applications across different cloud providers. In contrast, auto-scaling dynamically changes resource allocation based on demand in the here and now. Using these methods, businesses may improve speed, decrease latency, and guarantee that their apps work well in any cloud setting.

There are further obstacles in multi-cloud settings related to data management and interoperability. Consistent, available, and accessible data is essential when managing data across several cloud platforms. Tools for data integration and data synchronization make it easy for data to move freely across different cloud services. With data integration technologies, you can combine several data sources into one cohesive picture, and data synchronization keeps all platforms up-to-date. Data integrity and sound decision-making are both supported by these methods.

Last but not least, a comprehensive approach to application and service lifecycle management in a multi-cloud setting requires monitoring and oversight. If you want your cloud resources to be utilized efficiently and in line with your organization's goals, you need a governance structure that has rules, processes, and oversight mechanisms. Performance tracking, anomaly detection, and compliance assurance all

rely on continuous monitoring and reporting. Organizations may keep a tight rein on their multi-cloud systems and be ready for everything by instituting strong monitoring and governance procedures.

The significance of a holistic strategy that incorporates several approaches is highlighted when looking at the difficulties of multi-cloud management from the perspective of end-to-end operations management. Organizations may successfully traverse the complexity of multi-cloud environments by using cloud management platforms, multi-cloud security solutions, tools to optimize costs, approaches to improve performance, data management strategies, and strong governance frameworks. Improved IT operations in terms of agility, resilience, and cost-effectiveness are the end result of this comprehensive strategy, which takes into account both the inherent problems and the benefits of multi-cloud initiatives. Success in the ever-changing cloud environment will depend on enterprises' ability to continuously adapt and develop these tactics as their multi-cloud strategies change.

6.2 RECOMMENDATION OF THE STUDY

The following suggestions for better multi-cloud management practices are derived from an exhaustive study of the difficulties associated with multi-cloud environments and methods for overseeing end-to-end operations:

1. **Adopt a Unified Cloud Management Platform (CMP):** Organizations should consider investing in a unified CMP to simplify operations and guarantee smooth integration across various cloud environments. For effective management of cloud resources, this platform should provide centralized control, visibility, and automation capabilities. Organizations may streamline processes, codify rules uniformly, and enhance governance by bringing all management functions onto one dashboard.

2. **Implement Comprehensive Multi-Cloud Security Solutions:** Because every cloud provider has its own unique set of security policies and regulations, it is essential to implement multi-cloud security solutions. The goal of these solutions is to standardize security rules across all cloud environments and offer unified threat detection and response capabilities. Solutions with sophisticated security

features, such vulnerability assessment, identity and access management, and encryption, should be prioritized by organizations.

3. **Utilize Cloud Cost Optimization Tools:** Organizations may tackle the issue of unexpected prices by using cloud cost optimization technologies that provide comprehensive insight into use and spending. These resources should aid in planning and forecasting while also shedding light on spending habits and revealing potential places to save money. Setting up notifications for overspending and employing reserved instances are two cost management methods that may be used to assist control expenses efficiently.

4. **Leverage Auto-Scaling and Workload Migration Techniques:** Implementing auto-scaling and workload transfer solutions may help enterprises boost performance and optimize resource utilization. To achieve performance and cost goals, it is recommended to employ auto-scaling to dynamically modify resources according to real-time demand and workload migration to transfer applications and data across cloud providers. By using these strategies, you may maximize productivity while minimizing waste.

5. **Enhance Data Management and Integration:** In order to keep data consistent and accessible, it is vital to manage it effectively across several cloud platforms. Businesses should integrate solutions for data synchronization and integration that allow data to move freely across different cloud environments. To further guarantee data integrity and facilitate effective decision-making, data governance rules and procedures should be established.

6. **Establish Robust Governance and Monitoring Frameworks:** In order to keep multi-cloud environments under control, enterprises need create and execute thorough governance frameworks including rules, processes, and oversight tools. To keep tabs on performance, spot irregularities, and guarantee conformity with organizational goals and regulatory mandates, governance methods should include continuous monitoring and reporting.

Organizations may improve their IT operations' adaptability, robustness, and cost effectiveness by using these suggestions, which will help them tackle the difficulties of multi-cloud management and make the most of different cloud environments.

6.3 SUGGESTIONS OF THE STUDY

The following recommendations for improving the comprehension and administration of multi-cloud settings are derived from the findings of the analysis of multi-cloud difficulties and management approaches:

1. **Conduct In-Depth Case Studies:** Investigating comprehensive case studies of companies that have effectively used multi-cloud strategies may help one better grasp the practical aspects of multi-cloud management. In order to find the best practices, typical mistakes, and successful remedies, these case studies should center on different sectors and cloud setups.

2. **Develop Advanced Management Frameworks:** In order to better manage many clouds, future studies should concentrate on creating sophisticated frameworks that include new technologies like AI and ML. Automation, predictive analytics, and intelligent decision-making may all be improved with the help of these frameworks, leading to more efficient and successful cloud operations in general.

3. **Explore Hybrid Cloud Approaches:** Additional insights may be gained by investigating hybrid cloud systems that blend on-premises and cloud resources, in addition to multi-cloud strategies that entail leveraging services from different cloud providers. Companies thinking about combining cloud and on-premises resources may benefit greatly from studies that examine the effects of hybrid models on performance, cost, and security.

4. **Investigate Multi-Cloud Compliance Challenges:** Keeping up with regulations that apply to various cloud providers may be a challenging task. In order to guarantee constant conformity to industry norms and laws, future research should investigate the unique compliance issues of multi-cloud settings and provide solutions.

5. **Assess the Impact of Cloud Provider Interdependencies:** Studying the interconnections between various cloud providers and how they impact multi-cloud management as a whole should be the primary goal of researchers. It will be helpful to study the potential effects of problems with one provider on other providers and come up with plans to lessen such effects.

6. **Evaluate Cost Management Strategies:** More study is needed to determine which cost management tactics and technologies work best, since cloud providers' pricing schemes might vary widely. Research may look at the effects on total cloud spending and financial performance of various methods of budgeting, cost distribution, and optimization.

7. **Study User Experience and Interface Design:** Cloud management solutions' efficacy and uptake are greatly influenced by how easy they are to use. If you want to know how to make CMPs and multi-cloud management solutions more user-friendly and efficient, look into user experience research and interface design.

8. **Examine Long-Term Sustainability:** Managing and adapting to multi-cloud strategies is a constant process. How can businesses adapt their strategies to stay up-to-date with technology changes and evolving business needs? Future research should look at the viability of multi-cloud methods over the long run.

Future research may help businesses better navigate complicated cloud settings by addressing these ideas and contributing to a more thorough knowledge of multi-cloud management difficulties and solutions.

6.4 FUTURE SCOPE OF THE STUDY

Research into multi-cloud management is an ever-changing field, because to the ever-changing cloud computing world. The research on multi-cloud issues has a wide future potential since it covers a lot of ground that might be ground-breaking. Important points to consider for further investigation are:

1. **Integration of Emerging Technologies:**

    - **Artificial Intelligence and Machine Learning:** Look at ways that multi-cloud settings might benefit from AI and ML to improve automation, analytics, and decision-making. Investigate AI-powered solutions for forecasting performance trends, identifying outliers, and improving resource allocation.

- **Edge Computing and Serverless Architectures:** Analyze how server less and edge computing affect multi-cloud plans. Look at how these technologies may be incorporated into current multi-cloud setups to make them more efficient, scalable, and fast.

2. **Advanced Security Solutions:**

   - **Zero Trust Architecture:** Research the use of Zero Trust security concepts in setups with multiple clouds. Examine several cloud platforms to see how Zero Trust concepts might be used to improve security and compliance.

   - **Blockchain for Security and Compliance:** Look at how block chain technology may work to make multi-cloud setups more secure and compliant. Look at the potential applications of block chain technology for safe data transfers, audit trails, and regulation enforcement.

3. **Cost Optimization Innovations:**

   - **Dynamic Pricing Models:** Learn about dynamic pricing models and how they may help with managing costs in setups with many clouds. Determine the best ways to spend money by analyzing real-time price data.

   - **Cost Forecasting and Budgeting Tools:** Create and evaluate state-of-the-art budgeting and cost forecasting solutions. To get better, more meaningful cost information, look into how these tools can connect with cloud management solutions.

4. **Enhanced Data Management Techniques:**

   - **Data Fabric and Integration:** Examine how data fabric technologies may help bring together disparate cloud-based data management systems. Research the ways in which data fabric solutions streamline data management, synchronization, and integration.

   - **Data Privacy and Sovereignty:** Learn about the problems with and potential solutions to data sovereignty and privacy in multi-cloud setups.

Investigate ways in which businesses may manage data across various clouds in a way that complies with global data protection requirements.

To aid enterprises in navigating the complexity of multiple cloud environments and maximizing the potential of their multi-cloud strategies, future research should focus on these areas to improve multi-cloud management.

REFERENCES

CHAPTER-1

[1] R. Pareek, "Cloud Computing," *Journal of Global Research in Computer Science*, vol. 2, no. 7, pp. 1-7, 2011.

[2] Z. Tavbulatova, K. Zhigalov, S. Kuznetsova, and A. Patrusova, "Types of cloud deployment," *Journal of Physics: Conference Series*, vol. 1582, no. 1, pp. 1-6, 2020. doi: 10.1088/1742-6596/1582/1/012085.

[3] R. Nazir, Z. Ahmed, Z. Ahmad, N. Shaikh, A. Laghari, and K. Kumar, "Cloud Computing Applications: A Review," *EAI Endorsed Transactions on Cloud Systems*, vol. 6, no. 17, pp. 1-11, 2020. doi: 10.4108/eai.22-5-2020.164667.

[4] N. Shenoy, K. Kumar, and A. Rao, "An Audit on Cloud Architectures Addressing Data Privacy and Security Concerns," vol. 29, no. 6, pp. 6373-6382, 2020.

[5] P. Pawar, M. Naik, M. Choudhari, M. Tonge, and M. Adhav, "Analysis of Cloud Storage," *International Journal of Research and Analytical Reviews*, vol. 9, no. 2, pp. 392-402, 2022.

[6] M. G. Avram, "Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective," *Procedia Technology*, vol. 12, pp. 529-534, 2014. doi: 10.1016/j.protcy.2013.12.525.

[7] H. Imran, U. Latif, A. Ikram, M. Ehsan, A. Ikram, W. Khan, and S. Wazir, "Multi-Cloud: A Comprehensive Review," *IEEE*, vol. 1, no. 1, pp. 1-5, 2021. doi: 10.36227/techrxiv.13642850.

[8] M. AlZain, E. Pardede, B. Soh, and J. Thom, "Cloud Computing Security: From Single to Multi-clouds," *Hawaii International Conference on System Sciences*, vol. 4, no. 2, pp. 5490-5499, 2012. doi: 10.1109/HICSS.2012.153.

[9] P. Somasundaram and I. Aeme, "Enhancing Security in Multi-Cloud Environments Through Federated Access Control," *International Journal of Computer Engineering & Technology*, vol. 14, no. 2, pp. 90-96, 2023. doi: 10.17605/OSF.IO/7BMPA.

CHAPTER-2

[1] N. Bharath Kumar, "Challenges and Solutions for Integrating AI with Multi-Cloud Architectures," *International Journal of Multidisciplinary Innovation and Research Methodology*, vol. 1, no. 1, pp. 1-10, 2024.

[2] M. González-Rodríguez, L. Otero-Cerdeira, E. González-Rufino, and F. Rodríguez-Martínez, "Study and Evaluation of CPU Scheduling Algorithms," *Heliyon*, vol. 10, no. 9, pp. 29-59, 2024, doi: 10.1016/j.heliyon.2024.e29959.

[3] H. Fadhil, "Optimizing Task Scheduling and Resource Allocation in Computing Environments using Metaheuristic Methods," *International Journal of Computing Research*, vol. 15, no. 1, pp. 157-179, 2024.

[4] K. Merseedi and S. Zeebaree, "Cloud Architectures for Distributed Multi-Cloud Computing: A Review of Hybrid and Federated Cloud Environment," *Indonesian Journal of Computer Science*, vol. 13, no. 2, pp. 1644-1673, 2024.

[5] T. Kaur, "Multi-Cloud Networking: Navigating the Complexities of Advanced Cloud Networks," *International Journal of Advanced Research in Engineering & Technology*, vol. 15, no. 3, pp. 256-265, 2024.

[6] K. Voruganti, "Orchestrating Multi-Cloud Environments for Enhanced Flexibility and Resilience," *Journal of Technology and Systems*, vol. 6, no. 2, pp. 9-25, 2024, doi: 10.47941/jts.1810.

[7] R. Kotha, "Multi-Cloud Strategies for Enhanced Resilience and Flexibility," *Journal of Artificial Intelligence & Cloud Computing*, vol. 2, no. 4, pp. 1-5, 2023, doi: 10.47363/JAICC/2023 (2) E133.

[8] T. Salehnia, A. Seyfollahi, S. Raziani, A. Noori, A. Ghaffari, A. Alsoud, and L. Abualigah, "An optimal task scheduling method in IoT-Fog-Cloud network using multi-objective moth-flame algorithm," *Multimedia Tools and Applications*, vol. 83, no. 12, pp. 1-22, 2023, doi: 10.1007/s11042-023-16971-w.

[9] K. Senjab, S. Abbas, N. Ahmed, and A. ur R. Khan, "A survey of Kubernetes scheduling algorithms," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1-18, 2023.

[10] N. Nivedhaa and I. Pub, "Evaluating Devops Tools and Technologies for Effective Cloud Management," *International Journal of Cloud Computing (IJCC)*, vol. 1, no. 1, pp. 20-32, 2023.

[11] T. Alyas, T. Ghazal, B. Alfurhood, G. Issa, O. Ali, and Q. Niazi, "Optimizing Resource Allocation Framework for Multi-Cloud Environment," *Computers, Materials & Continua*, vol. 75, no. 2, pp. 4119-4136, 2023, doi: 10.32604/cmc.2023.033916.

[12] P. Somasundaram and I. Pub, "Enhancing Security in Multi-Cloud Environments Through Federated Access Control," *International Journal of Computer Engineering & Technology*, vol. 14, no. 2, pp. 90-96, 2023, doi: 10.17605/OSF.IO/7BMPA.

[13] S. Balasubramanian, A. Fawad, M. Zahoor, E. Ellahi, S. S. Yerasuri, and B. Muniandi, "Efficient Workload Allocation and Scheduling Strategies for AI-Intensive Tasks in Cloud Infrastructures," *Power System Technology*, vol. 47, no. 4, pp. 82-102, 2023.

[14] J. Alonso, L. Orue-Echevarria Arrieta, V. Casola, A. Torre, M. Huarte, E. Osaba, and J. L. Lobo, "Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1-20, 2023.

[15] P. Somasundaram, "Navigating Regulatory Compliance in Multi-Cloud Environments: Challenges and Technological Solutions," *International Journal of Core Engineering & Management*, vol. 7, no. 2, pp. 1-17, 2022.

[16] T. Zheng, J. Wan, J. Zhang, and C. Jiang, "Deep Reinforcement Learning-Based Workload Scheduling for Edge Computing," *Journal of Cloud Computing*, vol. 11, no. 3, pp. 30-50, 2022.

[17] W. Cheng, H. Feng, and G. Liang, "Design of IT Infrastructure Multi-cloud Management Platform Based on Hybrid Cloud," *Wireless Communications and Mobile Computing*, vol. 2022, no. 8, pp. 1-12, 2022, doi: 10.1155/2022/9227948.

[18] R. Kaur, Prof Verma, Kavita, N. Jhanjhi, and M. Talib, "A Comprehensive Survey on Load and Resources Management Techniques in the Homogeneous and Heterogeneous Cloud Environment," *Journal of Physics: Conference Series*, vol. 79, no. 1, pp. 12-36, 2021.

[19] H. Mezni, S. Mokhtar, S. Aridhi, and F. Charrada, "Towards big services: A synergy between service computing and parallel programming," *Computing*, vol. 103, no. 6, pp. 60-70, 2021.

[20] G. Chatzithanasis, E. Filiopoulou, C. Michalakelis, and M. Nikolaidou, "Exploring Cost-Efficient Bundling in a Multi-Cloud Environment," *Simulation Modelling Practice and Theory*, vol. 111, no. 6, pp. 30-50, 2021.

[21] E. El-Sharawy, "A Review on the CPU Scheduling Algorithms: Comparative Study," *International Journal of Network Security*, vol. 21, no. 1, pp. 19-26, 2021, doi: 10.22937/IJCSNS.2021.21.1.4.

[22] F. Ebadifard and S. M. Babamir, "Autonomic task scheduling algorithm for dynamic workloads through a load balancing technique for the cloud-computing environment," *Cluster Computing*, vol. 24, no. 2, pp. 1-27, 2021, doi: 10.1007/s10586-020-03177-0.

[23] N. Arora and R. Banyal, "Hybrid scheduling algorithms in cloud computing: A review," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 880-895, 2021, doi: 10.11591/ijece.v12i1.pp880-895.

[24] V. Bucur and L.-C. Miclea, "Multi-Cloud Resource Management Techniques for Cyber-Physical Systems," *Sensors*, vol. 21, no. 24, pp. 83-64, 2021, doi: 10.3390/s21248364.

[25] A. K. Ayyadapu, "Securing Multi-Cloud Environments with AI and Machine Learning Techniques," *Chelonian Conservation and Biology*, vol. 16, no. 2, pp. 1-12, 2021.

[26] N. Mohammad, "Enhancing Security and Privacy in Multi-Cloud Environments: A Comprehensive Study on Encryption Techniques and Access Control Mechanisms," *International Journal of Computer Engineering & Technology*, vol. 12, no. 2, pp. 51-63, 2021.

[27] L. Heilig, E. Lalla-Ruiz, and S. Voss, "Modeling and Solving Cloud Service Purchasing in Multi-Cloud Environments," *Expert Systems with Applications*, vol. 147, no. 3, p. 113-165, 2020.

[28] A. Priyadarshini, "Task Scheduling Algorithms in Multi-Cloud Environment," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, pp. 4530-4533, 2020.

[29] A. Aldailamy, J. Y. Maipan-uku, and A. Muhammed, "Heuristic Task Scheduling Algorithms for Optimal Resource Utilization in Grid Computing," *International Journal of Advanced Research in Engineering & Technology (IJARET)*, vol. 11, no. 12, pp. 1099-1107, 2020, doi: 10.34218/IJARET.11.12.2020.107.

[30] V. N. S. S. Chimakurthi, "The Challenge of Achieving Zero Trust Remote Access in Multi-Cloud Environment," *ABC Journal of Advanced Research*, vol. 9, no. 2, pp. 89-102, 2020, doi: 10.18034/abcjar.v9i2.608.

[31] O. Tomarchio, D. Calcaterra, and G. Di Modica, "Cloud resource orchestration in the multi-cloud landscape: A systematic review of existing frameworks," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 2-24, 2020, doi: 10.1186/s13677-020-00194-7.

[32] K. Kritikos, C. Zeginis, J. Iranzo, R. Gonzalez, D. Seybold, F. Griesinger, and J. Domaschka, "Multi-cloud provisioning of business processes," *Journal of Cloud Computing*, vol. 8, no. 1, pp. 50-60, 2019.

[33] P. Suresh, "Enhanced Resource Allocation and Workload Management using Reinforcement Learning Method for Cloud Environment," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 1-10, 2019, doi: 10.35940/ijrte.D8983.118419.

[34] B. Di Martino, A. Esposito, and E. Damiani, "Towards AI-Powered Multiple Cloud Management," *IEEE Internet Computing*, vol. 23, no. 3, pp. 64-71, 2019.

[35] J. Vijayashree, J. Jayashree, and M. Dr., "Multi-Cloud Performance and Workflow Management," *Journal of Emerging Technologies and Innovative Research*, vol. 6, no. 5, pp. 1-10, 2019.

[36] A. Celesti, A. Galletta, M. Fazio, and M. Villari, "Towards Hybrid Multi-Cloud Storage Systems: Understanding How to Perform Data Transfer," *Big Data Research*, vol. 16, no. 1, pp. 80-90, 2019.

[37] R. Cao, Z. Tang, C. Liu, and B. Veeravalli, "A Scalable Multi-cloud Storage Architecture for Cloud-Supported Medical Internet of Things," *IEEE Internet of Things Journal*, vol. 45, no. 2, pp. 1-1, 2019.

[38] S. Mandati, "The Influence of Multi-Cloud Strategy," *South Asian Journal of Engineering and Technology*, vol. 9, no. 1, pp. 1-4, 2019.

[39] A. Markus and J. D. D. Dombi, "Multi-Cloud Management Strategies for Simulating IoT Applications," *Acta Cybernetica*, vol. 24, no. 1, pp. 83-103, 2019.

[40] R. Chaudhary, G. Aujla, N. Kumar, and J. Rodrigues, "Optimized Big Data Management across Multi-Cloud Data Centers: Software-Defined-Network-Based Analysis," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 118-126, 2018.

[41] N. Ferry, F. Chauvel, H. Song, A. Rossini, M. Lushpenko, and A. Solberg, "CloudMF: Model-Driven Management of Multi-Cloud Applications," *ACM Transactions on Internet Technology*, vol. 18, no. 2, pp. 1-24, 2018, doi: 10.1145/3125621.

[42] M. Guo, Q. Guan, and W. Ke, "Optimal Scheduling of VMs in Queueing Cloud Computing Systems with a Heterogeneous Workload," *IEEE Access*, vol. 23, no. 3, pp. 1-1, 2018, doi: 10.1109/ACCESS.2018.2801319.

[43] B. Jambunathan and K. Dr., "Design of DevOps Solution for Managing Multi-Cloud Distributed Environment," *International Journal of Engineering & Technology*, vol. 7, no. 3.3, pp. 6-37, 2018, doi: 10.14419/ijet.v7i2.33.14854.

[44] A. Babu and M. Latha, "Efficient Ideal Algorithm for Task Scheduling in Cloud Computing," *International Journal of Engineering & Technology*, vol. 7, no. 3-12, pp. 5-50, 2018.

[45] K. Alexander, C. Lee, E. Kim, and S. Helal, "Enabling End-to-End Orchestration of Multi-Cloud Applications," *IEEE Access*, vol. 45, no. 3, pp. 1-1, 2017, doi: 10.1109/ACCESS.2017.2738658.

[46] H. Mezni and M. Sellami, "Multi-cloud Service Composition using Formal Concept Analysis," *Journal of Systems and Software*, vol. 134, no. 2, pp. 40-50, 2017.

[47] F. Lahmar and H. Mezni, "Multi-cloud service composition: A survey of current approaches and issues," *Journal of Software: Evolution and Process*, vol. 30, no. 8, pp. 1-30, 2017.

[48] V. N. S. S. Chimakurthi, "Risks of Multi-Cloud Environment: Micro Services Based Architecture and Potential Challenges," *ABC Research Alert*, vol. 5, no. 3, pp. 33-44, 2017.

[49] N. Mimura Gonzalez, T. Carvalho, and C. Miers, "Cloud resource management: Towards efficient execution of large-scale scientific applications and workflows on complex infrastructures," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 1-10, 2017.

[50] L. Heilig, E. Lalla-Ruiz, and S. Voss, "A Cloud Brokerage Approach for Solving the Resource Management Problem in Multi-Cloud Environments," *Computers & Industrial Engineering*, vol. 95, no. 3, pp. 3-40, 2016.

[51] K. Kritikos, T. Kirkham, B. Kryza, and P. Massonet, "Towards a security-enhanced PaaS platform for multi-cloud applications," *Future Generation Computer Systems*, vol. 67, no. 1, pp. 4-40, 2016.

[52] J. Moura and D. Hutchison, "Review and Analysis of Networking Challenges in Cloud Computing," *Journal of Network and Computer Applications*, vol. 60, no. 1, pp. 1-10, 2016.

[53] A. Ferrer, D. G. Pérez, and R. González, "Multi-cloud Platform-as-a-service Model, Functionalities and Approaches," *Procedia Computer Science*, vol. 97, no. 1, pp. 63-72, 2016.

[54] S. S. Gill and I. Chana, "A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges," *Journal of Grid Computing*, vol. 14, no. 2, pp. 80-90, 2016.

[55] F. Faniyi and R. Bahsoon, "A Systematic Review of Service Level Management in the Cloud," *ACM Computing Surveys*, vol. 48, no. 3, pp. 1-27, 2015, doi: 10.1145/2843890.

[56] J. Debnath and G. Serpen, "Real-Time Optimal Scheduling of a Group of Elevators in a Multi-Story Robotic Fully-Automated Parking Structure," *Procedia Computer Science*, vol. 61, no. 1, pp. 1-19, 2015, doi: 10.1016/j.procs.2015.09.202.

[57] V. Munteanu, C. Sandru, and D. Petcu, "Multi-cloud resource management: cloud service interfacing," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 3, no. 1, pp. 3-30, 2014.

[58] J. Jofre, C. Velayos, G. Landi, M. Giertych, A. Hume, G. Francis, and A. Oton, "Federation of the BonFIRE multi-cloud infrastructure with networking facilities," *Computer Networks*, vol. 61, no. 1, pp. 184–196, 2014, doi: 10.1016/j.bjp.2013.11.012.

[59] N. Grozev and R. Buyya, "Multi-Cloud Provisioning and Load Distribution for Three-Tier Applications," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 9, no. 3, pp. 1-21, 2014, doi: 10.1145/2662112.

[60] A. Agarwal and S. Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment," *International Journal of Computer Trends and Technology*, vol. 9, no. 7, pp. 7-70, 2014.

[61] J. Škrinárová, "Implementation and evaluation of scheduling algorithm based on PSO HC for elastic cluster criteria," *Open Computer Science*, vol. 4, no. 3, pp. 4-40, 2014, doi: 10.2478/s13537-014-0216-3.

[62] M. Singhal, S. Chandrasekhar, T. Ge, R. Sandhu, R. Krishnan, G.-J. Ahn, and E. Bertino, "Collaboration in Multi-cloud Computing Environments: Framework and Security Issues," *Computer*, vol. 46, no. 2, pp. 76-84, 2013, doi: 10.1109/MC.2013.46.

[63] M. AlZain, B. Soh, and E. Pardede, "A Survey on Data Security Issues in Cloud Computing: From Single to Multi-Clouds," *Journal of Software*, vol. 8, no. 5, pp. 90-100, 2013.

[64] A. A. Innocent, "Cloud Infrastructure Service Management - A Review," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 2, pp. 287-292, 2012.

[65] S. García-Gómez, M. Jimenez, Y. Taher, C. Momm, F. Junker, J. Bíró, A. Menychtas, V. Andrikopoulos, and S. Strauch, "Challenges for the Comprehensive Management of Cloud Services in a PaaS Framework," *Scalable Computing*, vol. 13, no. 3, pp. 40-50, 2012.

BIBLIOGRAPHY

Articles

[1] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in Cloud Computing: State of the Art and Research Challenges," *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 430-447, 2018. Available: 10.1109/TSC.2017.2711009.

[2] K. M. Sim and W. H. Sun, "Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions," *IEEE Trans. Syst. Man Cybern., Part A: Syst. Humans*, vol. 33, no. 4, pp. 560–572, 2017.

[3] M. N. Acharya and C. A. Patel, "Load Balancing in Cloud Computing Using Ant Colony Optimization," *Int. J. Comput. Eng. Technol.*, vol. 8, no. 6, pp. 54–59, Nov.-Dec. 2017.

[4] X. Wu, R. Jiang, and B. Bhargava, "On the Security of Data Access Control for Multi-Authority Cloud Storage Systems," *IEEE Trans. Serv. Comput.*, vol. 10, no. 2, pp. 258–272, 2017.

[5] P. Barham et al., "Xen and the Art of Virtualization," *ACM SIGOPS Operating Syst. Rev.*, vol. 37, no. 5, pp. 164–177, 2016.

[6] F. Paraiso, P. Merle, and L. Seinturier, "soCloud: A Service-Oriented Component-Based PaaS for Managing Portability, Provisioning, Elasticity, and High Availability Across Multiple Clouds," *Computing*, vol. 98, no. 5, pp. 539-565, 2016.

[7] B. Asgari, M. G. Arani, and S. Jabbehdari, "An Efficient Approach for Resource Auto-Scaling in Cloud Environments," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 5, pp. 2415–2424, Oct. 2016.

[8] P. Ray, "A Survey of IoT Cloud Platforms," *Future Comput. Informatics J.*, vol. 1, no. 1-2, pp. 35-46, 2016.

[9] R. Dawadi, S. Shakya, and R. Paudyal, "CoMMoN: The Real-Time Container and Migration Monitoring as a Service in the Cloud," *J. Inst. Eng.*, vol. 12, no. 1, pp. 51–62, 2016.

[10] M. Rathore, S. Rai, and N. Saluja, "Load Balancing of Virtual Machine Using Honey Bee Galvanizing Algorithm in Cloud," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 4, pp. 4128–4132, 2015.

[11] M. Su, L. Zhang, and Y. Wu, "Systematic Data Placement Optimization in Multi-Cloud Storage for Complex Requirements," *IEEE Trans. Comput.*, vol. 64, no. xx, pp. 1–14, xxx 2015.

[12] S. Nachiyappan and S. Justus, "Cloud Testing Tools and Its Challenges: A Comparative Study," *Procedia Comput. Sci.*, vol. 50, pp. 482-489, 2015.

[13] Soni, G. Vishwakarma, and Y. K. Jain, "A Bee Colony Based Multi-Objective Load Balancing Technique for Cloud Computing Environment," *Int. J. Comput. Appl.*, vol. 114, no. 4, pp. 0975–8887, Mar. 2015.

[14] Z. M. Jiang and A. E. Hassan, "A Survey on Load Testing of Large-Scale Software Systems," *IEEE Trans. Software Eng.*, vol. 41, pp. 1091-1118, 2015.

[15] R. Gao and J. Wu, "Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization," *Future Internet*, vol. 7, pp. 465–483, 2014.

[16] M. Ranjeet, S. Vivek, A. Jibi, and Rajni, "Analysis and Comparison of Symmetric Key Cryptographic Algorithms Based on Various File Features," *Int. J. Netw. Security Its Appl.*, vol. 6, no. 4, pp. 43–53, Jul. 2014.

[17] E. Pacini, C. Mateos, and C. García Garino, "Dynamic Scheduling Based on Particle Swarm Optimization for Cloud-Based Scientific Experiments," *CLEI Electron. J.*, vol. 17, no. 1, pp. 3–3, 2014.

[18] D. S. Papailiopoulos and A. G. Dimakis, "Locally Repairable Codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5843–5855, May 2014.

[19] C. Henry, H. Chen, H. Yuchong, and P. Patrick, "NCCloud: A Network-Coding-Based," *IEEE Trans. Comput.*, vol. 63, no. 1, pp. 31–46, Jan. 2014.

[20] S. B. and R. R. Rajalaxmi, "Artificial Bee Colony Based Feature Selection for Effective Cardiovascular Disease Diagnosis," *Int. J. Sci. Eng. Res.*, vol. 5, no. 5, May 2014.

[21] J. O. G. Garcia and K. M. Sim, "A Family of Heuristics for Agent-Based Elastic Cloud Bag-of-Tasks Concurrent Scheduling," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1682–1699, 2013.

[22] M. Singhal et al., "Collaboration in Multi-Cloud Computing Environment: Framework and Security Issues," *IEEE Comput. Soc.*, vol. 46, no. 2, pp. 76–84, Feb. 2013.

[23] M. Sun and W. Ren, "Improvement on Dynamic Migration Technology of Virtual Machine Based on Xen," in *Int. Forum on Strategic Technol.*, vol. 2, pp. 124–127, 2013.

[24] M. Velasquez and P. Hester, "An Analysis of Multi-Criteria Decision Making Methods," *Int. J. Oper. Res.*, vol. 10, no. 2, pp. 56–66, May 2013.

[25] R. M. Rahman and F. Afroz, "Comparison of Various Classification Techniques Using Different Data Mining Tools for Diabetes Diagnosis," *J. Softw. Eng. Appl.*, vol. 6, pp. 85–97, 2013.

[26] Y. Laili et al., "A Ranking Chaos Algorithm for Dual Scheduling of Cloud Service and Computing Resource in Private Cloud," *Comput. Ind.*, vol. 64, no. 4, pp. 448–463, 2013.

[27] M. Khorshed and A. Ali, "A Survey on Gaps, Threat Remediation Challenges and Some Thoughts for Proactive Attack Detection in Cloud Computing," *Future Gener. Comput. Syst.*, vol. 28, no. 6, pp. 833–851, Jun. 2012.

[28] S. Kumar and R. Ryan, "Cloud Computing - Research Issues, Challenges, Architecture, Platforms and Applications: A Survey," *Int. J. Future Comput. Commun.*, vol. 1, no. 4, pp. 356–360, Dec. 2012.

[29] S. Palanisamy and K. S., "Classifier Ensemble Design Using Artificial Bee Colony Based Feature Selection," *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 3, pp. 2, May 2012.

[30] T. Chen, "Comparative Analysis of SAW and TOPSIS Based on Interval-Valued Fuzzy Sets: Discussions on Score Functions and Weight Constraints," *Expert Syst. Appl.*, vol. 39, no. 2, pp. 1848–1861, Jan. 2012.

[31] T. Roblitz, "Towards Implementing Virtual Data Infrastructures – A Case Study with iRods," *Comput. Sci.*, vol. 13, no. 3, pp. 21–33, Sept. 2012.

[32] V. Vinothina, R. Sridaran, and P. Ganapathi, "A Survey on Resource Allocation Strategies in Cloud Computing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 6, pp. 97–104, 2012.

[33] Z. Yan, H. Hongxin, A. Gail-Joon, S. Stephen, and Y. Yau, "Efficient Audit Service Outsourcing for Data Integrity in Clouds," *Syst. Softw.*, vol. 85, no. 5, pp. 1083–1095, May 2012.

[34] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Softw.: Pract. Exp.*, vol. 41, no. 1, pp. 23–50, 2011.

[35] D. Warneke and O. Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 6, pp. 985–997, Jun. 2011.

[36] H. Jin, W. Gao, S. Wu, X. Shi, X. Wu, and F. Zhou, "Optimizing the Live Migration of Virtual Machines by CPU Scheduling," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1088–1096, 2011.

[37] S. Subhashini and V. Kavitha, "A Survey on Security Issues in Service Delivery Models of Cloud Computing," *J. Netw. Comput. Appl.*, vol. 38, no. 1, pp. 1–11, Jan. 2011.

[38] Y. Gao, G. Zhang, J. Lu, and H.-M. Wee, "Particle Swarm Optimization for Bi-Level Pricing Problems in Supply Chains," *J. Glob. Optim.*, vol. 51, no. 2, pp. 245–254, 2011.

[39] M. Hajjat, X. Sun, Y. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, and M. Tawarmalani, "Cloudward Bound," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, p. 243, 2010.

[40] C. Wang, Q. Wang, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *Proc. 29th IEEE Conf. Inf. Commun. (INFOCOM'10)*, San Diego, CA, Mar. 15-19, 2010, pp. 525–533.

[41] M. Swarnamugi, P. Dhavachelvan, and G. Sureshkumar M. Sathya, "Evaluation of QoS Based Web-Service Selection Techniques for Service Composition," *Int. J. Softw. Eng.*, vol. 1, no. 5, pp. 73–90, Feb. 2010.

[42] M. Vukolić, "The Byzantine Empire in the Intercloud," *ACM SIGACT News*, vol. 41, no. 3, p. 105, 2010.

[43] Rochwerger, D. Breitgand, E. Levy, and A. Galis, "The Reservoir Model and Architecture for Open Federated Cloud Computing," *IBM J. Res. Dev.*, vol. 53, no. 4, pp. 4:1–4:11, Apr. 2010.

[44] F. Angus, C. Huang, W. Lan, and J. Stephen, "An Optimal QoS-Based Web Service Selection Scheme," *Inf. Sci.*, vol. 179, no. 19, pp. 3309–3322, Sept. 2009.

[45] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype and Reality for Delivering Computing as the 5th Utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009.

[46] C. Makris, Y. Panagis, E. Sakkopoulos, and V. Diamadopoulou, "Techniques to Support Web Service Selection and Consumption with QoS Characteristics," *J. Netw. Comput. Appl.*, vol. 31, no. 2, pp. 108–130, Apr. 2008.

[47] H. Deng and C. Yeh, "Simulation-Based Evaluation of De-Fuzzification Based Approaches to Fuzzy Multi-Attribute Decision Making," *IEEE Trans. Syst. Man, Cybern., Part A: Syst. Humans*, vol. 36, no. 5, pp. 968–977, Sept. 2006.

[48] M. Reyes-Sierra and C. C. Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 287–308, 2006.

[49] K. E. Parsopoulos and M. N. Vrahatis, "Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization," *Nat. Comput.*, vol. 1, no. 2–3, pp. 235–306, 2002.

[50] Y. Dogan and F. Ozguner, "Matching and Scheduling Algorithms for Minimizing Execution Time and Failure Probability of Applications in Heterogeneous Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 3, pp. 308–323, Mar. 2002.

[51] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality Reduction Using Genetic Algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, no. 2, pp. 164–171, Jul. 2000.

[52] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

[53] C. Christer and R. Fuller, "Interdependence in Fuzzy Multiple Objective Programming," *Fuzzy Sets Syst.*, vol. 65, no. 1, pp. 19–29, May 1994.

Books

[1] E. Brown, *Advanced Multi-Cloud Management Techniques*, Elsevier, 2022.

[2] Thomas, *Multi-Cloud Deployment: Techniques and Best Practices*, O'Reilly Media, 2022.

[3] O. Martinez, *Techniques for Effective Multi-Cloud Management*, Morgan Kaufmann, 2022.

[4] B. Johnson, *Managing Multi-Cloud Environments: A Practical Guide*, Springer, 2021.

[5] G. Patel, *Multi-Cloud Architecture and Management*, Routledge, 2021.

[6] M. Harris, *Managing Complex Multi-Cloud Ecosystems*, Springer, 2021.

[7] Smith, *Multi-Cloud Management: Strategies and Techniques*, Wiley, 2020.

[8] L. Miller, *Multi-Cloud Operations and Integration*, Academic Press, 2020.

[9] H. Wilson, *Challenges in Multi-Cloud Management: A Comprehensive Overview*, Springer, 2020.

[10] Lee, *Cloud Computing and Multi-Cloud Management*, Cambridge University Press, 2019.

[11] J. Roberts, *Multi-Cloud Security and Compliance*, Wiley, 2019.

[12] N. Clark, *End-to-End Cloud Management: From Single to Multi-Cloud*, Elsevier, 2019.

[13] Kumar, *Comprehensive Multi-Cloud Strategies*, CRC Press, 2018.

[14] K. Green, *Optimizing Multi-Cloud Environments*, CRC Press, 2018.

[15] F. Davis, *End-to-End Operations in Multi-Cloud Systems*, Morgan Kaufmann, 2017.

LIST OF PUBLICATIONS

LIST OF INTERNATIONAL/NATIONAL CONFERENCE

1. Malege Sandeep, Analysing Multicloud Managing For Business Challenges, International Conference on ADVANCES IN Science, Engineering, Management And Humanities (ICASEMH – 2023), Gandhibagh, Nagpur, Maharashtra, India, 26th February, 2023

2. Malege Sandeep, Impact Of Multicloud Management On Optimal Workload Management, National Conference on Recent Advances in Science, Engineering, Humanities, and Management (NCRASETHM - 2024), Banquet, Noida, India, 28th January, 2024

LIST OF JOURNAL PUBLICATIONS

1. Malege Sandeep & Dr. Rohita Yamaganti Published the paper titled " Performance Evaluation of Sequential Assignment and Scheduling Algorithms in Cloudsim " in International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169 Volume: 11 Issue: 11, December 2023

2. Malege Sandeep & Dr. Rohita Yamaganti Published the paper titled " Analysis of Resource Allocation Performance Using MDP and Genetic Algorithm " in International Journal of Enhanced Research in Science, Technology & Engineering, ISSN: 2319-7463, Vol. 13 Issue 10, October-2024

REPRINT OF THE RESEARCH PAPER & CONFERENCE CERTIFICATES

IARF Conferences

International Conference on Advances in Science, Engineering, Management and Humanities (ICASEMH- 2023)

ICASEMH/2023/C0223 228

CERTIFICATE

This is to Certify that Malege Sandeep, Research Scholar, Ph.D in Computer Science & Engineering, P.K. University, Shivpuri, M.P.

has presented a paper on Analysing Multicloud Managing for Business Challenges.

at **International Conference on Advances in Science, Engineering, Management and Humanities (ICASEMH - 2023)** organized by the **International Academic Research Forum (IARF)** at Hotel Siddhartha Inn, Gandhibagh, Nagpur, Maharashtra, India. On 26th February, 2023

Dr. Rayess Ahmad Dar
Organizing Secretary

IARF CONFERENCES IARF

Dr. Shabnam Arora
Convener

INTERNATIONAL CONFERENCE ON ADVANCES IN SCIENCE, ENGINEERING, MANAGEMENT AND HUMANITIES (ICASEMH – 2023)

26TH FEBRUARY, 2023

CERTIFICATE NO : ICASEMH /2023/C0223228

ANALYSING MULTICLOUD MANAGING FOR BUSINESS CHALLENGES

MALEGE SANDEEP

Research Scholar, Ph. D. in Computer Science & Engineering
P. K. University, Shivpuri, M.P., India

ABSTRACT

Multicloud management has become an essential strategy for businesses aiming to address growing complexities in cloud computing. In a multicloud environment, organizations utilize services from multiple cloud providers, which allows for greater flexibility, risk mitigation, and cost optimization. However, managing multicloud solutions presents significant challenges, especially in terms of integration, data security, and cost control. One of the key challenges is ensuring interoperability between different cloud platforms, which often have distinct architectures and standards. This can complicate data migration, system communication, and workflow integration. Moreover, businesses must tackle security concerns, as multiple clouds increase the attack surface, making it harder to implement consistent security policies and compliance measures across platforms. Another challenge is the rising operational costs, as businesses may face unexpected charges from inefficient resource allocation or lack of comprehensive cost management tools across providers. To address these challenges, businesses must invest in robust multicloud management platforms that enable seamless integration, security management, and cost monitoring across multiple providers. In study, while multicloud environments offer numerous benefits such as reduced vendor lock-in and enhanced resilience, effective multicloud management requires comprehensive tools and strategies to overcome these operational and security hurdles.

IJESTI CONFERENCES

NCRASETHM/2024/C0124139

National Conference on Recent Advances in Science, Engineering, Humanities and Management (NCRASETHM-2024)

CERTIFICATE

This is to certify that Malege Sandeep,

Research Scholar, Ph.D in Computer Science and Engineering,

P.K University, Shivpuri, M.P.

has presented a paper in **NCRASETHM-2024** held on **28th** January 2024 at Hotel Krishna Residency & Banquet, Noida, India.

Title of the paper Impact of Multicloud Management on Optimal Workload Management.

Dr. Syed Imtiyaz
convenor

IJESTI Conferences

Dr. N. V. Krishna Prasad
Conference Chair

National Conference on Recent Advances in Science, Engineering, Humanities, and Management (NCRASETHM - 2024)
28th January, 2024, Banquet, Noida, India.

CERTIFICATE NO : NCRASETHM /2024/C0124139

IMPACT OF MULTICLOUD MANAGEMENT ON OPTIMAL WORKLOAD MANAGEMENT

MALEGE SANDEEP

Research Scholar, Ph. D. in Computer Science & Engineering
P. K. University, Shivpuri, M.P., India

ABSTRACT

The impact of multicloud management on optimal workload management is profound, as businesses increasingly adopt multiple cloud platforms to distribute and manage their workloads more efficiently. Multicloud strategies allow organizations to leverage the strengths of different cloud providers, ensuring that specific workloads are executed in the most suitable environment. This flexibility enhances performance, improves resource utilization, and reduces downtime, leading to better overall workload optimization. One of the key advantages of multicloud management is its ability to dynamically distribute workloads based on performance, cost, and availability. By managing workloads across different clouds, businesses can avoid overburdening a single platform, thus preventing bottlenecks and ensuring more efficient processing. This also allows for scaling on demand, enabling companies to handle fluctuating workloads without compromising performance. However, managing workloads in a multicloud environment presents its challenges. Without proper management tools, it can be difficult to monitor performance and balance resources across platforms. Multicloud management solutions help organizations address this by providing centralized control, automation, and real-time analytics to ensure that workloads are distributed optimally. Overall, effective multicloud management enhances workload efficiency by enabling businesses to take advantage of multiple cloud environments, ensuring that resources are allocated effectively while minimizing costs and maximizing performance.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: 25 July 2023 Revised: 12 September 2023 Accepted: 30 November 2023

Performance Evaluation of Sequential Assignment and Scheduling Algorithms in Cloudsim

[1]Malege Sandeep

[1]Research Scholar, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P.

[2]Dr. Rohita Yamaganti

[2]Associate Professor, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P.

Abstract

The CloudSim simulator is used for task scheduling, with jobs assigned to available resources according to the sequential and scheduling algorithms. The results demonstrate a significant improvement in both execution cost and completion time when using the scheduling algorithm, as compared to sequential assignment. Specifically, the scheduling algorithm achieves reduced execution costs and improved completion times as the number of cloudlets increases. The performance metrics highlight the effectiveness of the scheduling algorithm in optimizing task execution in a cloud computing environment.

Keywords: Sequential, Scheduling, CloudSim, Task, Resource

I.INTRODUCTION

When it comes to cloud computing, efficient resource allocation and task execution is absolutely dependent on how well sequential assignment and scheduling algorithms function. A modern computing foundation, cloud computing allows businesses to extend their operations without the expense and hassle of maintaining large hardware infrastructures by providing scalable and flexible computing resources. Nevertheless, whether handling large-scale applications or multiple users, effective scheduling and assignment procedures are essential for making the most of these resources. An open-source framework called CloudSim has recently come to light as a powerful tool for testing algorithms that operate on the cloud. It models and simulates various aspects of cloud computing. Data centers, virtual machines (VMs), and cloudlets—representing user tasks or applications—are all part of the cloud, and this platform is adaptable and extensible enough to accommodate them. With CloudSim, researchers may test various algorithms in a controlled environment before implementing them in real-world systems. It simulates the cloud's resource allocation and scheduling process.

Algorithms for task assignment and scheduling primarily aim to find the best way to distribute resources among jobs and guarantee their optimal execution. Time to completion of tasks, resource usage, load balancing, and overall system throughput are all aspects of performance that are directly impacted by these algorithms. In particular, algorithms for sequential assignment and scheduling are those that distribute resources and schedule jobs in a predefined order. Algorithms like these tend to shine in situations when jobs are interdependent or have clear priority.

Due to their ease of use and efficacy in resource management, sequential scheduling algorithms like First-Come-First-Served (FCFS), Shortest Job First (SJF), and Round Robin (RR) have been extensively researched. To keep up with the ever-evolving nature of cloud settings, it is essential to test how well these conventional scheduling algorithms handle situations where workloads, resources, and network conditions are subject to change. It is necessary to thoroughly examine the advantages and disadvantages of sequential scheduling algorithms due to the increasing severity of problems such resource congestion, task dependencies, and the requirement for load balancing. Several performance criteria, including execution time, makespan, resource utilization, energy consumption, and fairness, are used to evaluate sequential assignment and scheduling algorithms in CloudSim. An important metric in task scheduling is makespan, which is the sum of all the time needed to finish a set of tasks. The goal of most algorithms designed to minimize makespan is to speed up overall completion times by distributing work efficiently across available resources. To counteract resource idleness and the associated needless overheads in cloud environments, it is equally critical to achieve high resource utilization while simultaneously balancing loads. Adaptability to changing conditions, such as resource availability or task arrival rates, is another way to measure scheduling algorithms' performance.

Additionally, it is necessary to assess sequential scheduling algorithms' capacity to manage diverse and intricate cloud

infrastructures. Many virtual machines with diverse amounts of RAM, storage space, and computing power are commonplace in cloud environments, among other resources with diverse capabilities. Scheduling algorithms have a problem due to this heterogeneity because they must maximize system efficiency by allocating resources according to the unique needs of each activity. Consider the consequences of assigning a computationally heavy task to a virtual machine (VM) that lacks sufficient processing capacity. This could lead to longer execution times and worse overall system performance. In order to comprehend the scalability and practicality of sequential scheduling algorithms for use in real-world cloud systems, it is crucial to evaluate their performance in diverse cloud settings. One of the most important metrics for measuring cloud computing performance is energy usage. A major amount of the operational expenses of large-scale data centers are related to energy prices. Reduced power usage without sacrificing performance should be the goal of any effective scheduling method. While some sequential algorithms aim to reduce energy expenses by consolidating work onto fewer virtual machines (VMs), others may lead to underutilized resources or wasteful energy use. Because cloud providers are committed to sustainable computing and want to lessen their impact on the environment without sacrificing speed, evaluating scheduling algorithms' energy consumption is crucial.

By simulating a wide range of cloud environments—complete with varied numbers of virtual machines (VMs), cloudlet workloads, and network conditions—CloudSim gives researchers a controlled environment in which to test and compare scheduling tactics. Because it is possible to test algorithms in a simulated environment under a variety of conditions that would be costly or otherwise impractical to replicate in a real cloud setting, it is well suited for use in performance evaluations. Researchers can find the top methods for computationally heavy workloads or tasks with high data transfer requirements by using CloudSim.

II.REVIEW OF LITERATURE

Gollapalli, Mohammed et al., (2023) A continual commitment to customer satisfaction is of the utmost importance, especially with the exponential increase in the number of people using cloud services around the world. So, to improve the cloud computing environment's performance, job scheduling is crucial. It is possible to characterize the majority of the published research in this area by saying that it aims to maximize resource usage, reduce cost, and increase performance. Taking a number of factors into account, this study lays the groundwork for understanding more recent efforts to improve and optimize the current cloud computing task scheduling algorithm. Additionally, this study found the best performing algorithm in the cloud environment by comparing three task scheduling algorithms: Max-Min, First Come First Serve (FCFS), and Round Robin (RR). The performance metrics used were the cost of virtual machine resources, average time, and makespan. The CloudSim simulation tool was used to conduct the experimental evaluations. Based on makespan and average waiting time, Max-Min outperformed the other algorithms in Space and Time-shared policies.

Kaur, Rajbhupinder et al., (2021) With the idea of commercially applying technology to public consumers, cloud computing has become a leading technology. The capacity to dynamically assign resources on-demand, providing noticeable scalability, performance, low maintenance requirements, and cost-effectiveness, is what makes cloud computing so well-known. With the help of virtualization, users are able to share the resources. When job scheduling is done well, cloud computing can yield high performance. When thinking about how to dynamically allocate resources to maximize performance while minimizing makespan, task scheduling should be your first priority. Given its direct impact on other performance monitoring parameters such as power consumption, optimal cost, scalability, and availability, task scheduling is seen an important aspect. This study investigates the makespan in cloud computing using four well-known task scheduling algorithms: first-come-first-serve (FCFS), round-robin (RR), shortest-job-first (SJF), and particle swarm intelligence (PSO). The functioning of the job scheduling algorithms under consideration has been studied using VMs (virtual machines) and data centers with unique IDs. The makespan was calculated by recording the execution time needed to successfully complete each stage. The procedure for carrying out the task scheduling using the right algorithm, pseudocode, and flowchart is detailed in the study article. The outcomes have been derived from four different task scheduling algorithms and compared using makespan as the metric of choice.

Stan, Roxana-Gabriela et al., (2021) T His research lays the groundwork for a suite of techniques that may be used to assess how well a task scheduling scheme works in environments with different types of computers. We run tests on big workloads using simulations and formally state a scheduling paradigm for hybrid edge-cloud computing ecosystems. In addition to the traditional cloud datacenters, we also take into account edge datacenters that use battery-operated devices such as Raspberry Pi edge computers and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: 25 July 2023 Revised: 12 September 2023 Accepted: 30 November 2023

smartphones. We specify the computational resources' practical capabilities. After a timetable is established, the resource capacities determine whether or not the different needs of the tasks can be met. Using the Round-Robin, Min-Min, Max-Min, and Shortest Job First scheduling schemes, we construct an assessment and scheduling framework and measure common scheduling metrics like makespan, throughput, mean waiting time, and mean turnaround time. In contrast to cloud-only mediums, our research and findings reveal that state-of-the-art independent task scheduling algorithms perform worse in heterogeneous edge-cloud environments, with more task failures and less optimum datacenter resource usage. Specifically, for any scheduling system, over 25% of jobs fail to perform when dealing with big collections of tasks because of low battery or restricted memory.

Chowdhury, Nobo et al., (2018) An IT paradigm, cloud computing has seen extensive use in the delivery of a wide range of services over the Internet. It makes it simpler to get resources and premium services. In order for cloud services to be provided to consumers efficiently, their operational procedure needs to be planned. Allocating different computer resources to applications and achieving optimal system throughput are the two main objectives of task scheduling. The unpredictability of the scenario grows in relation to the magnitude of the task, making it highly likely to be solved efficiently. To shed light on this matter within the realm of cloud computing scheduling, a plethora of intellectual approaches are suggested. This study compares and contrasts various cloud-based scheduling algorithms with respect to the parameters that are important to each.

Rajguru, Abhijit & Apte, S.. (2013) Because it is the most crucial component of the computer system, the central processing unit (CPU) needs to be used effectively. The scheduling problem takes on critical importance as the demand for processing power grows. Research in computer engineering is most prominently focused on the critical and difficult topic of work scheduling and load balancing. Allocating processes to processors in a way that minimizes total execution time and optimizes processor utilization is the definition of task scheduling. Redistributing processing demands across available processors is known as load balancing, and it improves system performance. Using a variety of qualitative metrics, this article compares and contrasts the efficiency of many task scheduling methods. According to the results, there are benefits and drawbacks to using task scheduling algorithms. By analyzing current job scheduling algorithms, this research aims to contribute to the future creation of novel scheduling algorithms.

III.MATERIAL AND METHODS

To model a communication and heterogeneous resource environment, one can use the CloudSim toolkit. To confirm the findings, the CloudSim simulator is employed. The scheduling technique and CloudSim's default Sequential assignment are used for the trials. For the sake of generalization, the jobs arrive in a uniformly random distribution. The scheduler then uses these methods to send these jobs to the available resources. To compare the two algorithms, we change all of the parameters in the same way.

The datacenter configuration that was created looks like this: There are two hosts and one processing element.The virtual machines utilized for this experiment are set up as shown in the figure below.

Virtual Machines	Virtual Machine 1	Virtual Machine 2
RAM (MB)	5024	5024
Processing Power (MIPS)	22000	11000
Processing Element (CPU)	1	1

Figure 1: Configuration of VMs

IV.RESULTS AND DISCUSSION

Optimizing Performance in Terms of Execution Cost

By picking the right resource, cost-based tasks decrease the execution cost of individual activities using a greedy approach. Table 1 shows that compared to sequential allocation, the scheduling technique significantly improves the cost of task execution. The quantity of cloudlets has a direct correlation to the improvement in cost.

Table 1 Comparison of Execution Cost

No. Of Cloudlets	Scheduling Algorithm	Sequential Assignment
25	565.88	735.72
50	1131.83	1471.41
75	1697.70	2207.06
100	2263.57	2942.75

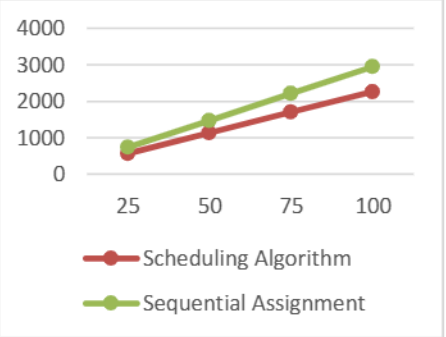


Figure 2: Analysis of Execution Cost

Scheduling Algorithm and Sequential Assignment are two scheduling algorithms. Table 1 analyzes their execution costs across different numbers of cloudlets. Both approaches have an increasing execution cost with an increasing number of cloudlets. Nevertheless, when contrasted with Sequential Assignment, the Scheduling Algorithm always reveals cheaper costs. The Scheduling Algorithm, for instance, would set you back 565.88 at 25 cloudlets, whereas Sequential Assignment would set you back 735.72.

Performance Optimization with Focus on completion time

The number of cloudlets is increased in each trial of the experiment. The results clearly show that scheduling algorithms outperform sequential approaches when it comes to job completion time.

Table 2 Comparison of Completion Time

No. Of Cloudlets	Scheduling Algorithm	Sequential Approach
25	53.01	58.95
50	164.68	181.65
75	334.50	399.88
100	584.72	654.05
125	910.08	998.03
150	1298.46	1439.77

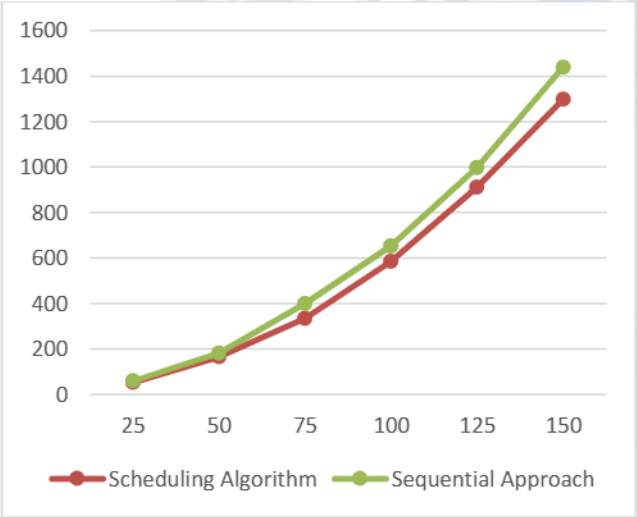


Figure 3: Analysis of Completion Time

According to the results in Table 2, the execution time for both approaches grows in direct proportion to the number of cloudlets. In contrast to the Sequential Approach, the Scheduling Algorithm reliably produces faster completion times. As an example, the Sequential Approach takes 58.95 units of time and the Scheduling Algorithm takes 53.01 units at 25 cloudlets.

V.CONCLUSION

The research shows that compared to the conventional sequential assignment method, the suggested scheduling algorithm is far more efficient in terms of execution time and cost. The results demonstrate that the suggested approach is effective in optimizing cloud resource use by simulating it in a heterogeneous resource environment using CloudSim. The suggested algorithm's ability to steadily decrease execution costs and improve completion times as the number of cloudlets grows verifies its promise for improving cloud computing efficiency. Advanced scheduling approaches are a viable option for managing resources and tasks in the cloud, as the results show how important they are for lowering operational expenses and satisfying deadline requirements.

REFERENCES: -

[1] Gollapalli, M., Al-Amoudi, A., Aldossary, A., Alqarni, A., Alwarthan, S., Almunsour, Y., Abdulqader, M., Mohammad, R., and Chaabani, S., "Modeling Algorithms for Task Scheduling in Cloud Computing Using CloudSim," *Mathematical Modelling of Engineering Problems*, vol. 9, no. 5, pp. 1201-1209, 2023.

[2] Kaur, R., Laxmi, V., and Jindal, B., "Performance evaluation of task scheduling algorithms in virtual cloud environment to minimize makespan," *International Journal of Information Technology*, vol. 14, no. 6, pp. 753-755, 2021.

[3] Stan, R.-G., Bajenaru, L., Negru, C., and Pop, F., "Evaluation of Task Scheduling Algorithms in Heterogeneous Computing Environments," *Sensors*, vol. 21, no. 17, pp. 1-16, 2021.

[4] Mahmood, I., Zeebaree, S., Shukur, H., Jacksi, K., Yasin, H., Radie, A., and Najat, Z., "Task Scheduling Algorithms in Cloud Computing: A Review," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 4, pp. 1041-1053, 2021.

[5] Hussain, A., Aleem, M., Azhar, M., Iqbal, M., and Islam, M. A., "Investigation of Cloud Scheduling Algorithms for Resource Utilization using CloudSim," *Computing and Informatics*, vol. 38, no. 3, pp. 525-554, 2019.

[6] Chowdhury, N., Uddin, K., Afrin, S., Adhikary, A., and Rabbi, F., "Performance Evaluation of Various Scheduling Algorithm Based on Cloud Computing System," *Asian Journal of Research in Computer Science*, vol. 2, no. 1, pp. 1-6, 2018.

[7] Malik, B., Amir, M., Mazhar, B., Ali, S., Jalil, R., and Khalid, J., "Comparison of Task Scheduling Algorithms in Cloud Environment," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, pp. 384-390, 2018.

[8] Singh, P., and Walia, N., "A Review: Cloud Computing using Various Task Scheduling Algorithms," *International Journal of Computer Applications*, vol. 142, no. 7, pp. 30-32, 2016.

[9] Indukuri, R., Varma, P., and Sundari, R., "Performance Evaluation of Two Stage Scheduling Algorithm in Cloud Computing," *British Journal of Mathematics & Computer Science*, vol. 6, no. 3, pp. 247-256, 2015.

[10] Atiewi, S., Yussof, S., and Rusli, M., "A Comparative Analysis of Task Scheduling Algorithms of Virtual Machines in Cloud Environment," *Journal of Computer Science*, vol. 11, no. 6, pp. 804-812, 2015.

[11] Himani, H., and Kaur, K., "Deadline Scheduling Algorithms in Cloud Computing: A Review," *International Journal of Computer Applications*, vol. 96, no. 24, pp. 19-21, 2014.

[12] Rajguru, A., and Apte, S., "A Performance Analysis of Task Scheduling Algorithms using Qualitative Parameters," *International Journal of Computer Applications*, vol. 74, no. 19, pp. 33-38, 2013.

[13] Sun, H., Chen, S.-p., Jin, C., and Guo, K., "Research and Simulation of Task Scheduling Algorithm in Cloud Computing," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 11, pp. 6664-6672, 2013.

[14] He, Z., Zhang, X., Zhang, H., and Xu, Z., "Study on New Task Scheduling Strategy in Cloud Computing Environment Based on the Simulator CloudSim," *Advanced Materials Research*, vol. 651, pp. 829-834, 2013.

[15] P. Salot, "A survey of various scheduling algorithm in cloud computing environment," *International Journal of Research in Engineering and Technology*, vol. 2, no. 2, pp. 131-135, 2013.

International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 13 Issue 10, October-2024, Impact Factor: 8.375

Analysis of Resource Allocation Performance Using MDP and Genetic Algorithm

Malege Sandeep[1], Dr. Rohita Yamaganti[2]

[1]Research Scholar, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P
[2]Associate Professor, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P

ABSTRACT

Optimal distribution of resources within the context of known limits and unknown uncertainties is the goal of resource allocation techniques, which pose a significant problem in many fields. To represent dynamic decision-making processes under uncertainty, the MDP framework offers a robust probabilistic model, and the GA searches for optimum or near-optimal solutions by using evolutionary principles. This study presents an approach to dynamically allocating cloud resources for NFV components using Markov Decision Process (MDP) applied to an NP-hard issue. In addition, Bayesian learning method is applied to monitor the historical resource usage in order to predict future resource reliability. Our suggested technique outperforms comparable approaches, according to experimental data..

Keywords: Network Functions Virtualization, Resource Allocation, Markov Decision Process, Bayesian Learning

INTRODUCTION

Numerous fields rely on efficient resource allocation, including manufacturing, healthcare, transportation, cloud computing, and telecommunications. Achieving efficient resource allocation is becoming more challenging as systems get more sophisticated and the need for optimum use increases. Markov decision processes (MDPs) and genetic algorithms (GAs) are two methods that have been extensively used to address resource allocation issues. When it comes to improving speed, saving cost, and making sure resources are distributed fairly, these solutions provide complimentary options. Allocating resources is the practice of systematically distributing limited resources across competing activities in order to accomplish goals. Limitations on time, money, or other resources are common in the actual world that affect this procedure. For example, in cloud computing, it is necessary to dynamically assign resources like CPU, memory, and storage in order to maximize processing efficiency and minimize energy usage. Allocating resources, such as delivery routes or cars, in logistics is also about minimizing operating costs and meeting tight deadlines. The presence of several interacting variables in these situations makes them intrinsically complicated due to the frequent involvement of stochastic settings.

When it comes to modeling decision-making in stochastic contexts, MDP is a strong approach. Finding the best policies for sequential choice issues might be a bit of a challenge, but MDP offers a systematic method based on dynamic programming and probability theory. When the system's state changes over time as a result of the selected actions and probabilistic transitions, it becomes very valuable. The four main components of MDP are states, actions, transition probabilities, and rewards. Using an MDP model of the resource allocation issue, decision-makers may assess the costs and benefits of both short-term and long-term benefits. To decrease delay and adapt to changing traffic circumstances, MDP may be used in network routing, for instance, to find the ideal pathways for data packets. As the number of states and action spaces increases, however, MDP's scalability becomes an increasingly pressing issue. The so-called "curse of dimensionality," in which computing demands grow unmanageable, is a common outcome of large-scale resource allocation challenges. This constraint is often filled by using approximation techniques like value iteration and reinforcement learning. Even with all these improvements, MDP may not be the best choice for situations with a lot of uncertainty or lack of structure since it relies on exact model parameters and transition probabilities.

However, Genetic Algorithms (GA) provide an alternative that is based on heuristics and is motivated by evolutionary and natural selection concepts. When dealing with optimization issues involving big and complicated search spaces, GAs shine where more conventional mathematical approaches fail. Optimal or near-optimal solutions may be found by GAs by repeatedly evolving a population of candidate solutions, which allows them to investigate a broad range of options. To guarantee search adaptability and robustness, GA's key operators—including mutation, selection, and

crossover—imitate biological processes. Among the many uses of GAs in resource allocation, job scheduling, load balancing, and portfolio optimization are just a few examples. By optimizing the deployment of virtual machines in the cloud, GA may minimize energy usage without compromising service quality, for example. For issues involving partial or ambiguous data, GAs are a good alternative to MDP as they don't need explicit modeling of transition probabilities. When applied to contemporary high-performance computing systems, GAs' parallel nature also enables quicker convergence.

REVIEW OF LITERATURE

Peng, Zhihao et al., (2022) Because it assigns jobs to the most suitable resources for execution—be it a grid, a cloud, or a network of peers—task scheduling is a crucial part of every distributed system. High temporal complexity, slower program execution durations, and non-simultaneous processing of input tasks are some of the drawbacks of common scheduling techniques. On heterogeneous distributed computing systems, exploration-based scheduling algorithms take a long time to execute because they need several techniques to prioritize work. Consequently, such systems have bottlenecks when it comes to job prioritizing. It is reasonable to use quicker algorithms to prioritize jobs based on their execution time. One evolutionary strategy to solving complicated problems rapidly is the genetic algorithm (GA). In order to schedule tasks on cloud computing with different priority queues, this study suggests a parallel GA that uses a MapReduce architecture. The primary objective of this research is to find the best way to use MapReduce architecture for cloud computing job scheduling in order to decrease overall execution time. There are two steps to the proposed method's task scheduling process. Initially, the GA was used with heuristic approaches to distribute tasks to processors. Afterwards, the GA was utilized with the MapReduce framework to distribute jobs to processors. Our tests take into account diverse resources with distinct task execution capabilities and the possibility of executing a job on several resources with different durations. Particle swarm optimization, intelligent water drops, moth-flame optimization, and the whale optimization algorithm are among the techniques that the suggested approach surpasses.

Khamayseh, Yaser et al., (2018) On the one hand, wireless networks have limited resources and experience several restrictions, making resource distribution a highly difficult process. In contrast, most resource allocation issues are NP-hard and need a large number of calculations. As a result, practical and efficient answers are badly needed. Fairly distributing the available resources of the network to all active users is the focus of resource allocation concerns. While defining fairness is challenging, this study takes into account the users' and the network operator's (service provider) perspectives on the matter. There are several applications for bio-inspired algorithms, which provide straightforward and efficient answers to complex issues. An efficient method for allocating resources in wireless networks for multimedia is presented in this article by use of a Genetic Algorithm. We use simulation to test how well the suggested approach works. The simulation results demonstrate that the suggested approach outperformed the others.

Ezugwu, Absalom et al., (2016) Managing grid resources effectively to execute the many tasks sent to the grid is crucial to the grid computing system's operational effectiveness. This study delves into a different approach to effectively search, match, and allocate distributed grid resources to tasks in a manner that satisfies the resource need of every grid user job. An approach to resource selection is proposed that utilizes populations based on multisets and is based on the notion of genetic algorithms (GA). Additionally, the article introduces a hybrid GA-based scheduling system that effectively finds the optimal resources for user workloads in a conventional grid computing setting. To improve the GA components' search capabilities in a wide problem space, the suggested resource allocation technique incorporates additional mechanisms, such as populations based on multiset and adaptive matching. The significance of operator improvement on conventional GA is shown via the presentation of empirical data. The suggested addition of a second operator fine-tuning is fast, accurate, and able to handle large work arrival rates, according to the early performance findings.

Porta, Juan (2013) Planning for future land uses is the focus of this research, which employs genetic algorithms. Expert opinion and up-to-date state and federal regulations inform the selection of applicable constraints and optimization targets. This strategy may readily integrate other factors. The shape-regularity of the land use patches that are produced and the appropriateness of the land itself are two optimization criteria that are used. When allocating land, we utilize the existing plots as a baseline. The method may take a long time to execute since there might be a lot of plots that are impacted. Thus, many parallel paradigms, including multi-core parallelism, cluster parallelism, and hybrids of the two, are the subject of the work's implementation and analysis. The results of these experiments demonstrate that genetic algorithms are well-suited to solving land use planning issues.

Goel, Tushar et al., (2003) This research focuses on the GA's potential utility in real-world industrial settings with constrained budgets for simulations. In particular, we look for a way to maximize limited multi-objective optimization while minimizing the overall simulation budget, which includes both the population size and the number of generations. We do a research to find ways to make things better, but we only let 1,000 simulations run. Time savings of a considerable magnitude were achieved by taking advantage of parallelization via the use of concurrent simulations for every GA generation on an HP quad-core cluster. In addition, we looked at how to distribute computing resources efficiently to get the most speed boost. We used a finite element model with 58,000 components to simulate the

crashworthiness of an automobile, in addition to two analytical cases. With a cap of 1,000 simulations, we tested out different population sizes and generations. Monte-Carlo and space filling sampling techniques were used to compare the optimization performance. It was discovered that the GA could find numerous workable solutions with trade-offs. To get excellent trade-off solutions, it is shown that a high number of generations is desirable. There were noticeable gains in efficiency with the vehicle's layout. Additionally, this example demonstrates that a large number of viable solutions may be obtained in the first few generations using about 200 simulations, even for problems with a tiny feasible area.

Dai, Y. et al., (2003) One of the most pressing problems in software testing is determining how to best use the available testing resources to provide the highest level of dependability. Models created under the premise of simple series or parallel modules are often used in the many papers on this problem. The optimization issue becomes more difficult to solve as system configurations get more complicated. When dealing with complicated software systems structures and various goals, this study introduces an evolutionary approach for testing-resource allocation challenges. When we allocate testing resources, we think about system dependability and testing costs simultaneously. Putting the strategy into action is a breeze. We present some numerical examples to demonstrate how the technique may be used.

EXPERIMENTAL SETUP

We have devised a battery of tests to evaluate the efficiency. The RHEL 6.5 operating system was used in all trials on Dell PowerEdge Servers R720 and R810. Our network offers a speed of 100 Mbps. In order to test the NFV components, we used WorkflowSim. Researchers may use WorkflowSim, a process simulator, to test out different methods for optimizing workflows. For this reason, we simulated varying VM and NFV component sizes using WorkflowSim.

Maximizing the predicted total reward over a limited horizon is one of the criteria we use to evaluate MDP. We calculate the best finite-horizon policy given an MDP and a horizon H. Horizon H= K and H=1 were the two cases we tested. Whereas H=1 simply takes into account one state, H=K signifies that we examine k-1 states till the end. We use the abbreviations MDPk for K-horizon and MDP1 for 1-horizon MDP here. A popular scheduling method is the Genetic Algorithm (GA), which we have compared to our suggested algorithms, MDPk and MDP1.

Our experiment's GA configuration looks like this: Random seed, one-third mutation rate, one-point cross-over recombination, one-wheel selection as parents, twenty-odd generations, one-hundredth of a population size, and uniform mutation method.

Two measures, execution cost and time restriction, are used to assess the allocation strategies. The first one shows the time cost divided by the allocation, while the second one shows the cost to complete the NFV components.

We devised studies to investigate how well our suggested MDP performed under varied settings, such as with variable numbers of components, virtual machines, and due dates. We divided the whole time cost into two parts so that we could examine it more thoroughly. The time cost of initializing all conceivable states of MDP is recorded by one step that is supplied as an initialization stage. The second stage is a solution stage that keeps track of how much time it took to determine the best allocation plan.

RESULTS AND DISCUSSION

Here are the experimental findings with different numbers of NFV component jobs and a fixed number of virtual machines (VMs), as shown in Figures 1 and 2. Figure 2 shows the overall running time of the solution stage and Figure 1 shows the entire running time of the approach. By comparing the solution stage running time cost of other approaches and taking execution time into account, we find that our MDPk method performs better. Nevertheless, the approach calculates all potential allocation solutions after traversing all possible states in the startup step. It is evident from comparing Fig. 1 with Fig. 2 that the initialization step takes much more time than the solution stage. One important consideration in dynamic resource allocation is the time cost of solutions. Given that evaluating a single node takes a constant amount of time, O(1), the execution time for MDPk is O(t(v+1)), and for MDP1 it is O(t × v), where t is the number of NFV component tasks and v is the number of virtual machines..

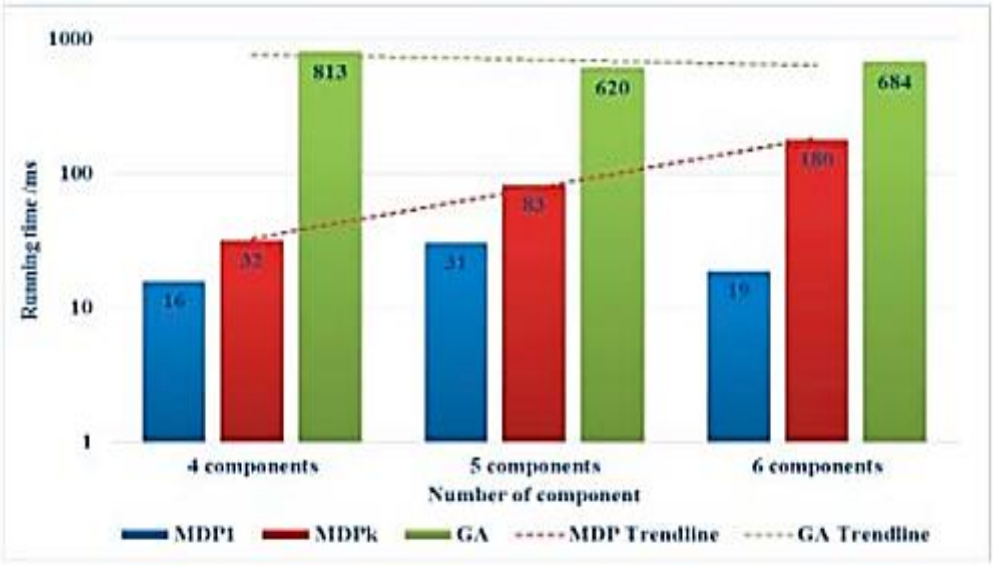


Figure 1: Total running time with different number of NFV Component

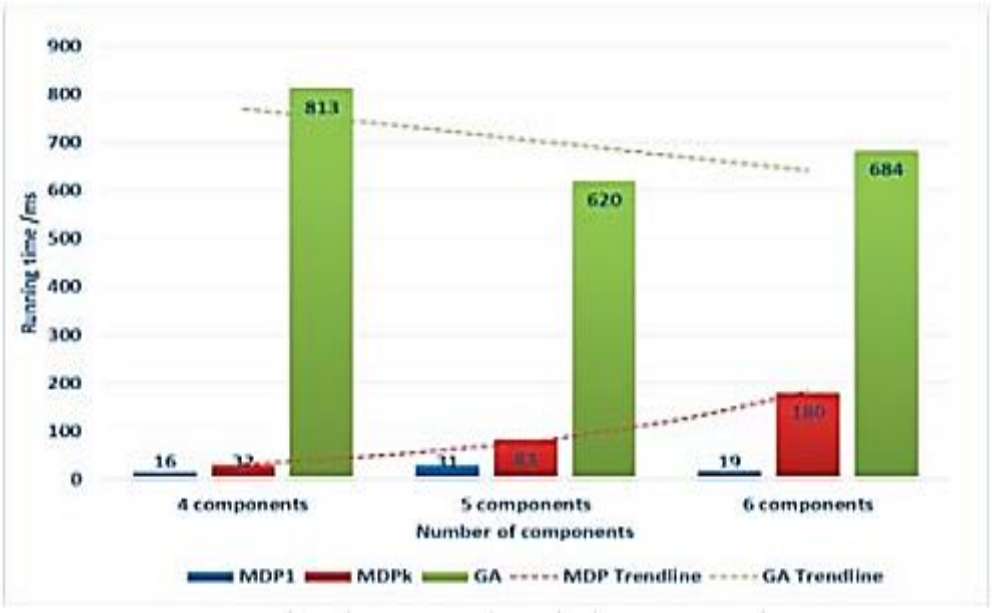


Figure 2: Solution stage running time with different number of NFV Component

Experimental findings for recording method running time while establishing a constant number of NFV component jobs and modifying the numbers of virtual machines (VMs) are shown in Figure 3. Although GA has a higher running time than other approaches, it is impossible to predict how long it will take since the GA program continues to run until either stopped or interrupted at a certain point in time. Meanwhile, GA isn't able to provide the best possible outcome on a global scale. Whenever the random seed is not consistent, the outcomes will vary. When all parameters, including the number of people and the number of generations, crossover rate, and mutation rate, remain constant, GA may provide a better allocation plan at a lower time cost, particularly for complex NFV processes and VM circumstances.

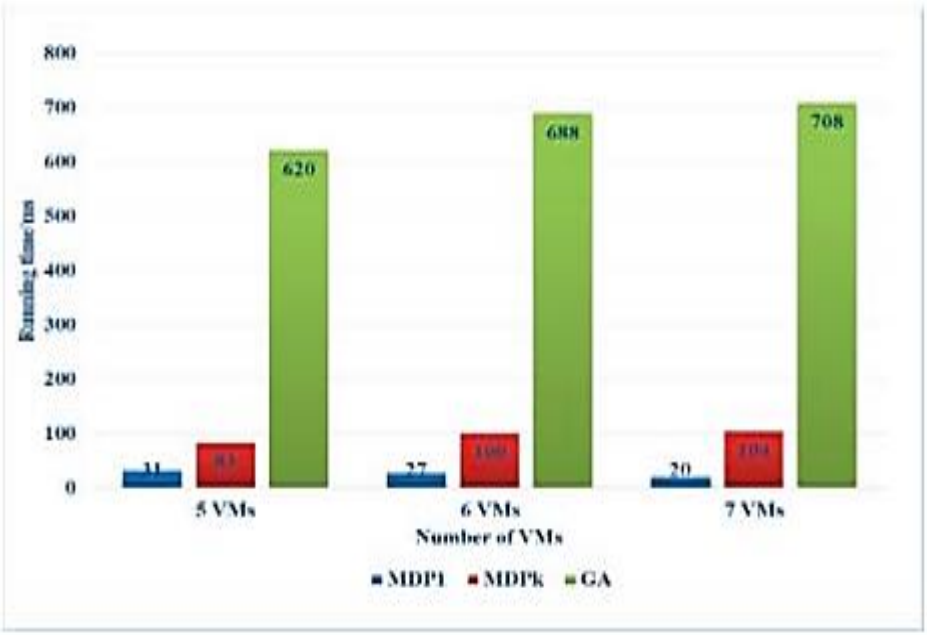


Figure 3: Running time with different number of VMs

With a considerable rise in the number of components, MDPk and MDP1 take up more time than GA on the total running time trend line, with the majority of that time being spent during startup. When compared to GA, the time required for the solution stage of MDPk and MDP1 is much lower. One explanation is that compared to GA, MDP is more likely to do floating point calculations. GA lacks expected value computation, in contrast to MDP.

A constant number of virtual machines (VMs) and a variable number of NFV component jobs are used to illustrate our experimental findings (Fig. 4). Figure 5 displays the outcomes of our experiments when the number of VMs is varied but the number of NFV component jobs remains constant. When changing NFV component tasks or virtual machines (VMs), MDPk might achieve the global optimum. The GA result is comparable to the MDPk result. In theory, the K-horizon MDP can discover the optimal solution in any feasible state. Some allocation plans will be excluded when the deadline component is taken into account, however, since the deadline partition is backward-designed while the technique to determine the total cost is forward-looking.

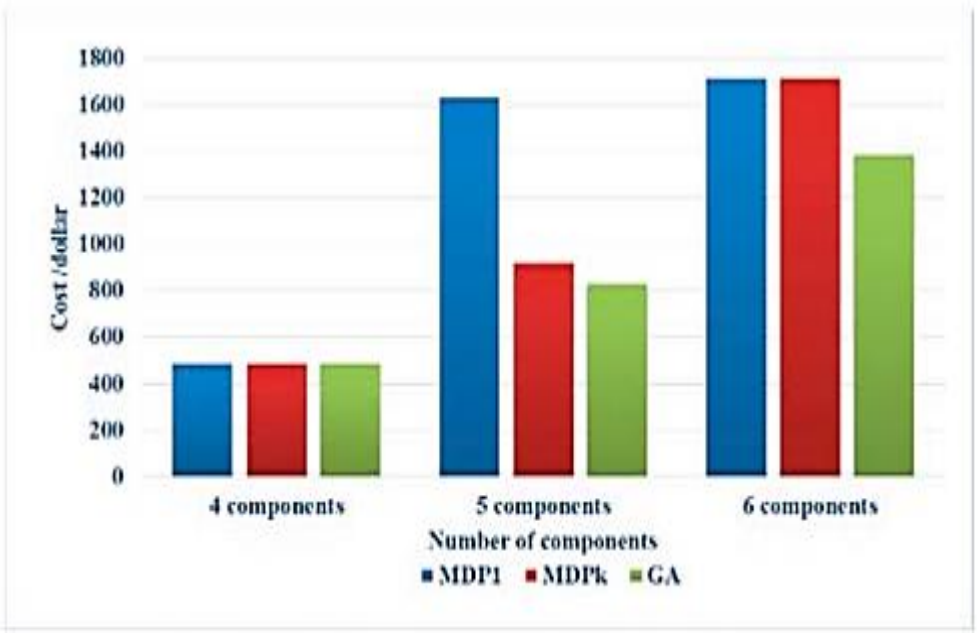


Figure 4: Cost with different number of NFV Component

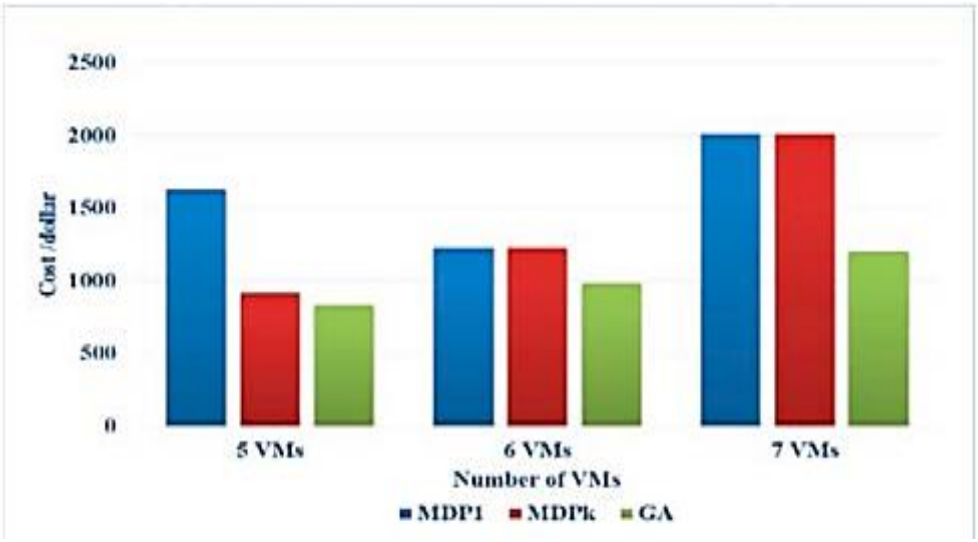


Figure 5: Cost with different number of VMS

To further investigate the potential impact of the deadline on the outcome, we devised an experiment with several due dates for recording the total cost. Figure 6 displays the outcomes. Taking the deadline into account, the GA approach may sometimes provide a more optimal allocation plan. Because it skips over one step and picks the best node in that stage, MDP1 performs the poorest. What this means is that it just reflects local optimization in the near term.

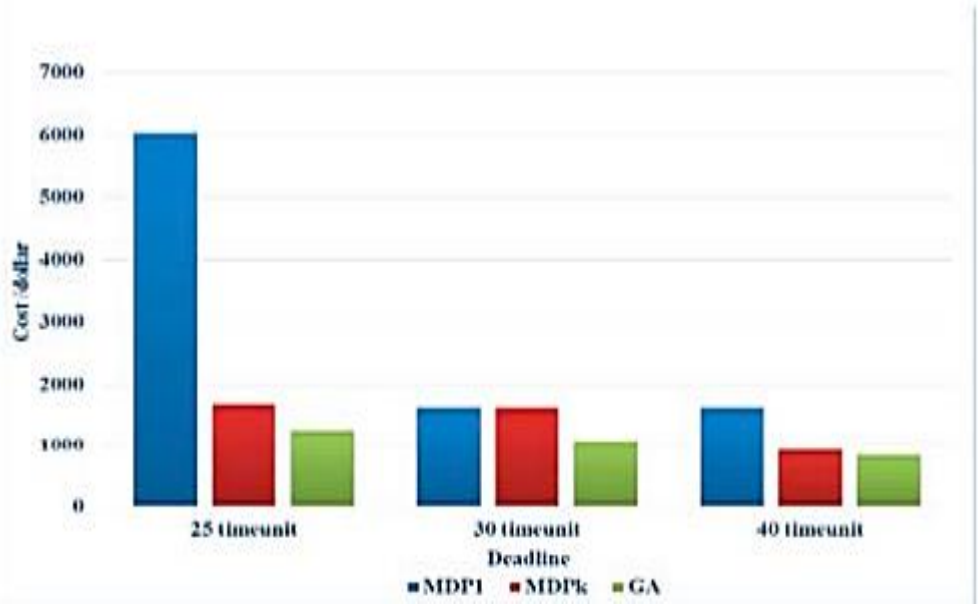


Figure 6: Cost with different deadlines

For this reason, MDP1 should not be used first in cases when time is of the essence. Because MDP1 can't predict the whole cost and the cost looks to be cheaper for certain selections at some phases, it can't strike a balance between the short and long term. As the deadline draws nearer, MDP1 must decide whether to invest more resources to save time and succeed or fail.

CONCLUSION

When comparing MDPk, MDP1, and Genetic Algorithm (GA), it becomes clear that MDPk achieves global optimization and has the fastest solution stage running time. This is particularly true when dealing with different NFV component jobs, virtual machines, and deadlines. The exhaustive state traversal in MDPk's initialization step increases the time cost, but the solution stage is more efficient and reliable than GA, which has inconsistent results and can't ensure global optimality. Although it would take a relatively long period, our MDPk may achieve the global ideal. When considering the time cost, GA is the best way for solving the problem. But GA won't always come up with the optimal answer. When faced with a tight timeline, GA can still identify a better answer. Only when dealing with somewhat unique challenges may MDP1 save time

REFERENCES

[1]. H. Jafari Kaleybar, M. Davoodi, M. Brenna, and D. Zaninelli, "Applications of Genetic Algorithm and Its Variants in Rail Vehicle Systems: A Bibliometric Analysis and Comprehensive Review," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2023, doi: 10.1109/ACCESS.2023.3292790.

[2]. Z. Peng, P. Pirozmand, M. Motevalli, and A. Esmaeili, "Genetic Algorithm-Based Task Scheduling in Cloud Computing Using MapReduce Framework," *Mathematical Problems in Engineering*, vol. 2022, no. 4, pp. 1–11, 2022, doi: 10.1155/2022/4290382.

[3]. Z. Peng, P. Pirozmand, M. Motevalli, and A. Esmaeili, "Genetic Algorithm-Based Task Scheduling in Cloud Computing Using MapReduce Framework," *Mathematical Problems in Engineering*, vol. 2022, no. 4, pp. 1–11, 2022, doi: 10.1155/2022/4290382.

[4]. S. Wang, Y. Wu, and R. Li, "An Improved Genetic Algorithm for Location Allocation Problem with Grey Theory in Public Health Emergencies," *International Journal of Environmental Research and Public Health*, vol. 19, no. 15, p. 9752, 2022, doi: 10.3390/ijerph19159752.

[5]. S. Jomthanachai, W. Rattanamanee, R. Sinthavalai, and W.-P. Wong, "The application of genetic algorithm and data analytics for total resource management at the firm level," *Resources, Conservation and Recycling*, vol. 161, p. 104985, 2020, doi: 10.1016/j.resconrec.2020.104985.

[6]. R. P. M. and D. M., "Efficient task allocation approach using genetic algorithm for cloud environment," *Cluster Computing*, vol. 22, no. 21, 2019, doi: 10.1007/s10586-019-02909-1.

[7]. Y. Khamayseh and R. Al-qudah, "Resource Allocation using Genetic Algorithm in Multimedia Wireless Networks," *International Journal of Communication Networks and Information Security*, vol. 10, no. 2, pp. 419–424, 2018, doi: 10.17762/ijcnis.v10i2.3280.

[8]. W. Rankothge, F. Le, A. Russo, and J. Lobo, "Optimizing Resource Allocation for Virtualized Network Functions in a Cloud Center Using Genetic Algorithms," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 1–1, 2017, doi: 10.1109/TNSM.2017.2686979.

[9]. A. Ezugwu, N. Okoroafor, S. Buhari, M. Frincu, and S. Junaidu, "Grid Resource Allocation with Genetic Algorithm Using Population Based on Multisets," *Journal of Intelligent Systems*, vol. 26, no. 1, pp. 169–184, 2016, doi: 10.1515/jisys-2015-0089.

[10]. M. Zarra-Nezhad, R. Yousefi Hajiabad, and M. Fakhernia, "Evaluation of Particle Swarm Algorithm and Genetic Algorithms Performance at Optimal Portfolio Selection," *Asian Economic and Financial Review*, vol. 5, pp. 88–101, 2015.

[11]. J. Porta, F. Parapar, F. Rivera, I. Santé, and I. Crecente, "High performance genetic algorithm for land use planning," *Computers, Environment and Urban Systems*, vol. 37, no. 1, pp. 45–58, 2013, doi: 10.1016/j.compenvurbsys.2012.05.003.

[12]. A. Haruna, N. Abba, N. Zakaria, and N. Haron, "Grid Resource Allocation: A Review," *Research Journal of Information Technology*, vol. 4, no. 2, pp. 38–55, 2012.

[13]. J. Gu, J. Hu, T. Zhao, and G. Sun, "A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment," *Journal of Computers*, vol. 7, no. 1, pp. 42–52, 2012, doi: 10.4304/jcp.7.1.42-52.

[14]. T. Goel, N. Stander, and Y.-Y. Lin, "Efficient resource allocation for genetic algorithm based multi-objective optimization with 1,000 simulations," *Structural and Multi-Disciplinary Optimization*, vol. 41, no. 3, pp. 421–432, 2010, doi: 10.1007/s00158-009-0426-9.

[15]. Y. Dai, M. Xie, K. Poh, and B. Yang, "Optimal testing-resource allocation with genetic algorithm for modular software systems," *Journal of Systems and Software*, vol. 66, no. 1, pp. 47–55, 2003, doi: 10.1016/S0164-1212(02)00062-6.

[16]. M. Louati, S. Benabdallah, F. Lebdi, and D. Milutin, "Application of a Genetic Algorithm for the Optimization of a Complex Reservoir System in Tunisia," *Water Resources Management*, vol. 25, no. 10, pp. 2387–2404, 2011, doi: 10.1007/s11269-011-9814-1.

[17]. J. Alcaraz and C. Maroto, "A Robust Genetic Algorithm for Resource Allocation in Project Scheduling," *Annals of Operations Research*, vol. 102, no. 1, pp. 83–109, 2001, doi: 10.1023/A:1010949931021.

ER Publications

ISSN: 2319-7463, New Delhi, India

International Journal of Enhanced Research in Science, Technology & Engineering

UGC Certified International Peer-Reviewed & Refereed Journal

UGC Journal no. 7618

Certificate of Publication

Malege Sandeep

Research Scholar, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P

TITLE OF PAPER

Analysis of Resource Allocation Performance Using MDP and Genetic Algorithm

has been published in

IJERSTE, Impact Factor: 8.375, Volume 13, Issue 10, Oct. 2024

Paper Id: STE/2810202442
Date: 30-10-2024

Website: www.erpublications.com
Email: erpublications@gmail.com

Authorized Signatory

ER Publications

ISSN: 2319-7463, New Delhi, India

International Journal of Enhanced Research in Science, Technology & Engineering

UGC Certified International Peer-Reviewed & Refereed Journal

UGC Journal no. 7618

Certificate of Publication

Dr. Rohita Yamaganti

Associate Professor, Department of Computer Science and Engineering, P. K. University, Shivpuri, M.P

TITLE OF PAPER

Analysis of Resource Allocation Performance Using MDP and Genetic Algorithm

has been published in

IJERSTE, Impact Factor: 8.375, Volume 13, Issue 10, Oct. 2024

Paper Id: STE/2810202442
Date: 30-10-2024

Website: www.erpublications.com
Email: erpublications@gmail.com

Authorized Signatory