STM JOURNALS

Journal of
Open Source Developments

ISSN: 2395-6704
Volume 12, Issue 1, 2025
January–April
DOI (Journal): 10.37591/JoOSD

https://journals.stmjournals.com/joosd

Review JoOSD

A Study of Open-source Cloud Computing Platforms

V. Basil Hans*

Abstract

Cloud computing has modernized the way organizations manage, store, and process data, offering scalable and cost-effective solutions. Open-source cloud computing platforms have emerged as a key enabler in this transformation, providing flexibility, transparency, and community-driven innovation. This study discusses the most prominent open-source cloud platforms, including OpenStack, CloudStack, and Kubernetes, highlighting their architecture, features, and use cases. The study deals with their comparative strengths, limitations, and adoption challenges in various industries. Furthermore, it explores the role of open-source platforms in fostering collaboration, reducing vendor lock-in, and accelerating technological advancements. Through a comprehensive analysis, the present study aims to guide enterprises, developers, and researchers in selecting and leveraging open-source cloud solutions to meet their specific requirements. The findings underscore the pivotal role of open-source platforms in shaping the future of cloud computing. This study explores the architecture, features, and adoption challenges of leading open-source cloud platforms. It also offers a comparative analysis, enabling organizations and developers to make well-informed choices when adopting these solutions. By examining their contributions to the cloud computing ecosystem, this study highlights the role of open-source platforms in driving innovation, democratizing access to cloud technologies, and shaping the future of IT infrastructure.

Keywords: Open source, cloud platforms, access, IT infrastructure, open source platforms, open-source projects

INTRODUCTION

Cloud computing has become a cornerstone of modern digital infrastructure, enabling organizations to access computational resources and services over the internet. Its widespread adoption is driven by the promise of scalability, cost efficiency, and the ability to innovate rapidly. Within the broad spectrum of cloud computing solutions, open-source cloud platforms stand out as a vital subset due to their inherent flexibility, transparency, and collaborative development model.

Open-source platforms offer an alternative to proprietary cloud services by empowering organizations to build, deploy, and manage cloud environments tailored to their needs. These platforms foster innovation by allowing developers to modify and enhance source code, thereby reducing dependency on specific vendors and encouraging the adoption of standards-based solutions.

Prominent open-source platforms, such as OpenStack, CloudStack, and Kubernetes, have gained significant traction in recent years. They support a wide range of applications, including infrastructure as a service (IaaS), platform as a service (PaaS), and container orchestration. These platforms are widely used in industries such as healthcare, finance, and education, where data security, compliance, and customization are critical.

***Author For Correspondence**
V. Basil Hans
E-mail: vhans2011@gmail.com

Research Professor, Department of Management & Commerce, Srinivas, University, Mangaluru, Karnataka, India

Received Date: January 29, 2025
Accepted Date: February 03, 2025
Published Date: March 07, 2025

Citation: V. Basil Hans. A Study of Open-source Cloud Computing Platforms. Journal of Open Source Developments. 2025; 12(1): 25–36p.

The potential scalability given by cloud computing has been leveraged on the technology that allows the abstraction of physical resources as virtual ones. For those who are struggling to adopt this technology, others have taken parallel steps on how to provide and manage the cloud services. Given that, several firms have made available their APIs, allowing the exposition of resources and services, and thus making the cloud private platforms and marketplaces a new possibility. The buildup of these platforms requires the cloud infrastructure itself; otherwise, how one could manage resources in the cloud without having access to them? In addition, large data centers in education and research deployed in several countries propelled some of the most known cloud computing solutions.

This study explores the architecture, features, and adoption challenges of leading open-source cloud platforms. It also provides a comparative analysis to help organizations and developers make informed decisions about implementing these solutions. By examining their contributions to the cloud computing ecosystem, this study highlights the role of open-source platforms in driving innovation, democratizing access to cloud technologies, and shaping the future of IT infrastructure.

LITERATURE REVIEW

The literature on open-source cloud computing platforms reveals a dynamic interplay between user control, service accessibility, and security considerations. Buyya *et al*. laid the groundwork by discussing the limitations of user customization within cloud services, particularly under the Software as a Service (SaaS) model [1].

Building on these insights, Salih provide a comparative analysis of open-source and closed-source platforms in cloud computing [2]. They note that while entirely open-source platforms primarily cater to Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), the scarcity of open-source Software as a Service (SaaS) applications presents a notable gap. Their findings suggest that open-source solutions may offer cost advantages and enhanced security due to user control, yet they also point out significant challenges related to documentation and design, which can hinder user experience and system performance.

Bhuiyan *et al.* further expanded on the discourse by investigating the influential factors affecting cloud adoption, particularly concerning security objectives [3]. Their work outlines a comprehensive taxonomy of cloud computing services and delves into the complexities of security service level agreements (SLAs) within cloud environments. Their exploration of trust-enhanced security in SaaS applications, particularly from the perspective of small and medium enterprises (SMEs) in contrasting geographical contexts, highlights the varying security challenges faced by organizations. Their analysis of the OPTIMIS framework for holistic cloud service provisioning underscores the need for robust security measures in fostering user trust and facilitating broader cloud adoption.

Moreover, the article touches upon the importance of standardization and interoperability in cloud computing, which are essential for fostering an open cloud environment. The authors reference the Open Cloud Manifesto, advocating for interoperability between providers and scalability for applications. This initiative is critical, as it addresses the fragmentation that often plagues cloud services and emphasizes the need for a cohesive framework that can facilitate seamless integration across different platforms.

However, while the article adeptly identifies these challenges, it could benefit from a more detailed exploration of practical solutions or case studies that illustrate successful implementations of interoperability and security measures in open-source cloud platforms. Additionally, the discussion on the implications of these challenges for developers, engineers, and service providers could be expanded to provide a more holistic view of the ecosystem surrounding cloud computing.

The article "Survey and comparison for Open and closed sources in cloud computing" by Salih and Zang provides a comprehensive overview of the landscape of open-source cloud computing platforms, highlighting both their advantages and limitations [2]. The authors effectively categorize the available tools and applications primarily at the Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) levels, noting a significant scarcity of Software as a Service (SaaS) open-source applications. This observation is critical as it underscores the potential gaps in service offerings within the open-source ecosystem, particularly for organizations seeking comprehensive solutions across all service models.

One of the key insights presented in the article is the cost-effectiveness of open-source solutions compared to their closed-source counterparts. The authors argue that the open-source methodology tends to be more economical, which is a compelling factor for organizations looking to optimize their cloud computing expenditures. However, this advantage is tempered by the challenges associated with documentation and design, especially when novice volunteers contribute to open-source projects. This aspect raises valid concerns regarding the usability and reliability of such platforms, which could deter potential users who may lack the technical expertise to navigate these shortcomings.

The article also touches on the security implications of using open-source cloud platforms. Salih and Zang suggest that the inherent user control in open-source systems may lead to enhanced security compared to closed systems [2]. This point is particularly relevant in today's digital landscape, where data breaches and security vulnerabilities are prevalent.

In the article "An enhancement of toe model by investigating the influential factors of cloud adoption security objectives", provides a comprehensive examination of the multifaceted challenges associated with cloud computing, particularly focusing on security and service-level agreements (SLAs) [3]. The main thrust of the article is to enhance the existing Technology-Organization-Environment (TOE) framework by identifying key factors influencing cloud adoption, especially concerning security objectives.

Bhuiyan *et al.* outlined the taxonomy of cloud computing services, which is crucial for understanding the different models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and their respective security implications. The article highlights that each service model presents unique security challenges that organizations must navigate to effectively implement cloud solutions. This taxonomy not only categorizes the services but also serves as a foundational element for analyzing security considerations, thus providing a structured approach for organizations to assess their cloud strategies.

OVERVIEW OF CLOUD COMPUTING

Cloud computing is a dynamic technology and a powerful method for delivering various services over the Internet. Cloud computing enhances Information Communication Technology (ICT) services and offers computing resources anywhere, anytime. To maximize the benefits of cloud computing, understanding of technical issues and the state of open-source cloud computing platforms is required, particularly in building Infrastructure as a Service (IaaS) clouds. With the wide range of cloud computing services being accessible through online means, the development of cloud services is inevitable in the technology field. In support of this, cloud computing frameworks are constructed using open-source cloud platforms, which allow for freedom of choice. They have augmented service providers providing free usage, helping researchers deploy cloud applications and allowing for the sharing of research outcomes. Furthermore, the adoption of cloud technology has become widespread in retail and academic circles as a cost-efficient computation model. It is expected that the cloud computing market value will grow substantially into the next decade, focusing on the Asia Pacific region due to its growing technological innovation. Iqbal *et al*. enumerate and confirm an overview of cloud computing, its characteristics, along with its services [4]. A comparative evaluation of Nimbus, OpenStack, OpenNebula, and Eucalyptus in the development of IaaS clouds is determined. By providing detail in this area, scholars interested in cloud technologies can uncover an understanding of the complex cloud technology, helping the creation of IaaS technology using open-source platforms possible.

OPEN-SOURCE SOFTWARE AND ITS IMPORTANCE

Until the modern technological landscape took its pace, proprietarily developed software thoroughly dominated the industry. However, with the open-source movement that began in the late 20th century, the importance of software that is completely free in terms of distribution and source code started to become more evident. Open-source software (OSS) can be comprehended as software whose source code has been made public by the developer. In its essence, it is a way of setting software free [5]. A major difference in between open-source and proprietary software is the business model behind each development process. Proprietary software is developed by companies whose main aim is profit. On the other hand, open-source software development is based on the principles of collaboration and community involvement. A source code accessible to anyone is a key to this philosophy. This had a tremendous, far-reaching effect toward facilitating innovation that anticipates radical alterations and presents opportunities for radically improved solutions across various sectors. Open-source attributes like freedom of source code, and openness to public scrutiny have played an integral part in contributing to agility and stimulating ingenuity, bringing about operational models and facilitating resource alignment that would not involve open-source alternatives [4]. Additional benefits of open-source software over its proprietary counterpart include cost-effectiveness and, to some extent, flexibility.

KEY FEATURES AND COMPONENTS OF CLOUD COMPUTING PLATFORMS

Cloud computing platforms have rapidly evolved to become essential tools for businesses, governments, and organizations. A cloud computing platform can deliver services to a wide range of users or consumers via the internet. In general, cloud computing platforms offer a variety of service levels, ranging from Infrastructure as a Service, Platform as a Service, to Software as a Service. Some cloud platforms also deliver dedicated services such as security or even Artificial Intelligence as a service. This wide variety of services reflects the inherent complexity and versatility of cloud computing platforms.

At its core, cloud computing platforms provide a shared pool of configurable computing resources to a variety of devices in multi-tenancy mode. This implies that physical hardware is connected over a network with front-end interfaces to deliver a broad range of computational services on demand. Cloud computing platforms are driven by a combination of hardware technologies, such as networking and virtualization, and middleware technologies, such as services orchestration and cloud development frameworks. The move towards software enabled virtual infrastructures has uncoupled the hardware-computational resource service model. Virtual resources can now be instantiated on any physical resource in the data center. This greatly facilitates resource pooling and scalable services. These developments were quickly followed by the emergence of cloud data centers and commercial cloud providers. Cloud data centers possess an enormous capacity to host thousands of physical servers and physical storage resources.

Virtualization

Cloud computing is based on the concept of data remoting: the idea is to have large data centers, possibly far away from each other and from geographically distributed users. The data centers provide sprinkled services and it is a high performance network to link them with a vast heterogeneity of clients (users, appliances). Users and appliances access the data centers using standard protocols defined in service level agreements and, depending on the agreement, they are provisioned with services (virtual machines, software, data). In this remoting model, resources are virtualized and may no longer be physically owned or managed by the provider of the service.

Virtualization is the key technology absconding, such as computing power or network bandwidth, and encapsulating these assets within a virtual instance, freeing the client from details and idiosyncrasy of setups. It enables the usage of resources concurrently without being aware of the physical location and reduces costs by improving resource efficiency, sharing and isolation. The application of this technology to different resources has taken different forms: it can be considered regarding computing (as in traditional sense, where it mostly resides), but storage and network are also important abstractions taking equally considerable attention.

Virtualization allows for a better utilization of resources by preventing resources from being idle. It provides isolation, hence a fault in an application or in an operating system does not affect the others. Moreover, for security reason an attacker can share the same physical system but does not have access to others.

Orchestration

A crucial aspect in the field of cloud computing involves the automation of the creation, scaling and termination of machines or containers that may provide services for them. This automaton, termed orchestration, aims to hide complexity and effort needed to operate one or more services so that they may be easily scaled and increased. Orchestration tools automate tasks such as resource provisioning and deployment of applications or services. By modeling services, frameworks or workflows, they enable the coordination of different services and resources throughout the same orchestrator, making more efficient workflows and automating otherwise complex ones since it avoids the manual operation of many elements of a system. On a lower level, orchestration tools may help the co-located services or applications to improve latency and resource usage. It can also react to specific changes on other services and scales its own resources to better cope with operational demands.

Key Functionalities

The key functionalities that can be performed by orchestration platforms are: (i) simplifying the deployment of applications, by providing an API and necessary tools to render easier the process of packaging and deploying applications; (ii) scaling the application across machines, by facilitating the operation that scales the application horizontally, adding similar services and resources in order to better distribute the load among them; (iii) managing the lifecycle of the deployment, by handling the very beginning and the very end of the deployment, and also the mid-term operations, such as response time guarantees; (iv) managing the infrastructure, by monitoring the state of the infrastructure to take decisions on a macro scale about new deployments and killed or out-of-use VMs; (v) managing the application, by allowing deployment with a specified plan, and guaranteeing that applications are not deployed until all necessary services are up, and also manage service-to-service connectivity [6].

Mature orchestration frameworks such as Swarm, Kubernetes or Mesos come as great time savers due to the many additional functionalities they have on top of “vanilla” Docker. With their CLI, developers are able to: specify a certain plan on which the application has to be deployed; manage meta-knowledge about the deployment that could be then used to optimize the deployment plan; expose data about the state of the application and infrastructure. With a more microservices-oriented mindset, tools like automated rolling updates or canary releasing became a must in order to keep a robust multicontainer application.

Seeing a service as a distributed bundle of a container containing the service’s main business logic, and additional containers for caching, messaging, load balancing or database can help in thinking in a more service-oriented way. It can be quite beneficial for better applications optimization, and for providing a better user experience to the customers. The rise of orchestration frameworks led to a more mainstream usage of microservice architecture. Varied applications could re-use the same underlying storage services or use the same DB that could be too distant if not using DC-locals [7].

On the other hand, the various scaling models, such as the ones provided by Kubernetes "Horitzonal Pod Autoscalar" and "Cluster Autoscalar", which vary between individual pods scaling within a set of resources, and pods scaling based on the global level. Unfortunately, multi-dimensional optimization problems are NP-hard and being on a large scale adds further complications to state. This is in part due to the added complexity of the system. In a commonly desired scenario, the most efficient service should be chosen among a great number of the same services that can be launched across various instances concurrently.

Networking

Cloud networking is essential in cloud computing because it provides the ability to connect or communicate between cloud services, within a cloud data center, or between the cloud data center and another network. The underlying network infrastructure of a cloud data center forms an important foundation of cloud operations. Poor network connections could degrade significantly the performance of moving data into the cloud. Therefore, a cloud user heavily relies on the elasticity and upgraded underlying networking infrastructure of the public cloud provider. Cloud networking implies Internet-based computing that provides shared processing resources and data to computers and other devices on demand. Public cloud networks can provide different VPCs for isolating compute instances concurrently by creating a virtual network and subnet, respectively, and assigning such network resource items to them. In terms of network peering, it is granted the capability of connecting to other service providers and exchanging traffic via the peering link. High-throughput bandwidth networking is the underlying infrastructure of cloud-based services for accessing big data workloads. Bandwidth management can be eventually equivalent to optimizing cloud performance. There are about 450 subsea cables spanning over 1.3 million km worldwide. A fast increase in transatlantic subsea cables has been constructed because of the time and the huge capacity trade-off to gain Internet routing optimization.

POPULAR OPEN-SOURCE CLOUD COMPUTING PLATFORMS

The Shift from Cloud Computing to Edge Computing: Open-Source Cloud Computing Platforms

Popular Open-Source Cloud Computing Platforms

In the world of cloud computing, several open-source cloud computing platforms have been developed and made available to the public as free software. Those platforms offer services to help manage and deploy cloud-based solutions, making it easier for small-to-large enterprises to deploy and use cloud platforms without the need for creating their own cloud infrastructure from the ground up. Such platforms have a large community that gathers around them and offers both free software support (often in the form of online versions of the user or administration manual and forum, web-based support for reporting and debugging) and paid support offerings. For smaller businesses, open-source cloud computing platforms can fill the role of a flexible, personalized, low-cost cloud environment adapted to their needs. These platforms, however, are also used by medium-sized and large businesses [2]. The study will perform an in-depth comparison and analysis of the following open-source platforms: OpenStack, CloudStack, and an open-source, but somewhat narrowed down, Kubernetes. The study will extensively showcase the strengths and weaknesses of each analyzed platform as well as their practical use (i.e. how they can be fitted into various businesses depending on their needs: sizing, costs, etc.). Each platform will be analyzed from the perspective of the cloud client. What services can the client get and how?

OpenStack

OpenStack is currently the most popular platform of the three analyzed platforms. It is possible to create and manage a device using the most popular programming languages, such as Java, Python, Ruby, and PHP, or with the web-based administration interface offered by OpenStack as Horizon. OpenStack is modular software for cloud computing, which is used for the management of platform resources such as computation, disk storage, or broadband. Individual services which are used to do particular job in OpenStack were named individual parts. Each project is maintained separately but operates within a framework [8].

Apache CloudStack

Apache CloudStack is probably the least complex cloud computing framework of the three analyzed frameworks, and is roughly similar to OpenStack with a modular design, however, it is less modular and the user is more enclosed in the EC2 compatible glue code layer, which makes it easier to manage EC2-compatible tasks and many have web-based interfaces, which from the user's perspective separates the operation of virtual machines from the operation of the framework. Creating and managing cloud devices with CloudStack is possible with Horizon, basic web-based administration interface or with Apache Server API (CSAPI) interface.

Kubernetes

While Kubernetes itself is not a full cloud computing service, it is one of the principal components for creating cloud computing as Container as a Service (CaaS), which is a microservice architecture as a service. With the creation of containers and their orchestration, Kubernetes is able to individually manage the services that make up particular solution. This orchestration is used to manage and automate certain components. For example, the applications in containers are restarted when they exceed the threshold of CPU/memory.

Comparison

Open-source cloud solutions offer ways to automate software installation and configuration, provide ready-to-use images or virtual machines, and help transfer workloads between cloud locations. In turn, open-source software gives freedom to modify the code as needed and meet the clients' personalized cloud requirements. Major industries began to back away from the concept of cloud computing and started looking for other solutions, such as so-called re-clouding, the return migration of IT infrastructure from cloud computing platforms to local environments. For Linux, many open-source solutions complementing and sometimes replacing the functions of popular proprietary platforms have been created. The new wave comes from Linux community, a wide range of open-source solutions that enhance and provide security for public cloud platforms, such as the widely-recognized IaaS, PaaS, or SaaS models.

OpenStack

Cloud computing provides scalability and flexibility, allowing businesses to adapt quickly to evolving needs on a large scale. Numerous enterprises are relying more and more on scalable, reliable, and low-cost IaaS cloud solutions. Though plenty of established web services already exist, it is easy to put up own IaaS cloud to benefit from the advantages of cloud computing. Regarding fast evolving landscape, open source IaaS cloud solutions are seen as more future-proof investment. Open-source software is becoming more acceptable by developers nowadays; not only because of the lower cost, it also empowers the developers to change the solution as per their needs.

A systematic study and comparative evaluation of the existing open-source IaaS cloud platforms with respect to feature content and implementation maturity have been performed. The studied IaaS cloud platform solutions are from the open source software sector. CloudStack, OpenStack, Eucalyptus, OpenNebula, and Nimbus are widely known and most actively developed open-source cloud platforms.

Apache CloudStack

This section delves into a study on open-source cloud platforms that can be used to build Infrastructure as a Service (IaaS) clouds, where the most suitable platform must be chosen for building a cloud infrastructure. There are many open-source cloud platforms available for academic and industrial cloud deployments. Some of the popular platforms are [9]: OpenStack, CloudStack, and Eucalyptus, which are the most widely used platforms around the world. While some platforms are used as end-to-end solutions for converting infrastructure into a cloud, other platforms are composed of multiple solutions to build a private cloud by integrating them together. To choose the most suitable platform among the aforementioned options, a detailed comparison article on the platforms will describe different deployment aspects and characteristics, where the platforms are compared covering architecture, networking, platform, hypervisor, and the language supported [4].

Apache CloudStack, which is a collective solution for IaaS cloud deployment, is an open-source software designed to deploy and manage large networks of virtual machines, as a highly available, scalable cloud-computing platform. CloudStack is known for providing an Infrastructure as a Service (IaaS) platform for delivering cloud computing.

CloudStack has a central server that provides a web interface for managing the cloud and data center infrastructure nodes for running the cloud deployment. The web interface of CloudStack is more user-friendly compared to OpenStack. CloudStack has all the necessary features in a unified user-friendly environment aligned under a consistent tab layout. Occasional experience of displaying disconnects when working with API and the web interface, under high loads. The architecture of CloudStack is much simpler compared to OpenStack geo-distributed applications.

Kubernetes

Kubernetes is an open-source platform primarily written in Go programming language that deals with container orchestration. Containers have been quite a popular way of packaging and deploying applications for a few years now due to its high portability, efficient resource management, and easy dependency management. Nonetheless, orchestrating containers is a tedious task, especially when aiming to deploy containers at a large scale. Kubernetes, as a container management platform, deals with scaling, deployment, networking, workload balancing, storage, updating, configuration management, and runtime. This sub-section explains the architecture and internals of Kubernetes, delineating the key components, their functionality, and how they work in harmony to provide containerized applications with a distributed, self-healed environment for execution.

"Deployments" ensure that the containerized applications are running at the requested state and facilitate zero-downtime rolling updates. The convenience of deploying/dealing with complex, multi-container applications, defining multiple types of resources, and viewing and searching objects in a declarative way contributed to the widespread adoption of Kubernetes. The advantages of Kubernetes in diving and updating applications, its powerful and flexible networking model, executing distributed computing trivially, supported integration to various volume plugins for storage, and process-level resource isolation and group management within pods have made Kubernetes heavyweight containers in a multi-tenant environment, highly beneficial for managing and scaling containerized applications. In addition, Kubernetes flexibly supports various deployment patterns and strategies such as blue-green/canary deployment, A/B testing, and deploying canary versions of new templates with traffic weights allowing services for an application to be available via network. It has an extensible, pluggable architecture with a robust community and vast ecosystem backing that enhances or replaces its components using APIs and client libraries. Finally, Kubernetes has a wide range of real-world applications and use cases at various scales across a diverse range of sectors like online uses for batch data processing, and business intelligence, genomics, and particle physics, and academic Kerberos/NFS active directory usage, backing Jenkins-based CI system, job system implemented with Pykube/job spec, and on-running service mesh installation, to name a few of many.

COMPARATIVE ANALYSIS OF OPEN-SOURCE CLOUD PLATFORMS

The availability of open-source cloud computing platforms plays a vital role in enabling researchers and product developers to go beyond the state of the art for the development of cloud applications. In this context, a comparative performance analysis of five different open-source cloud computing platforms is presented, considering Infrastructure as a Service (IaaS) clouds. The selected open-source cloud computing platforms are Apache CloudStack, OpenStack, OpenNebula, Eucalyptus, and OpenStack. The comparative analysis of the aforementioned platforms in terms of performance, scalability, and service features enables informed decisions to be made on the adoption of cutting-edge open-source cloud solutions for the development of novel cloud applications [4].

Cloud computing is a dynamic and versatile field of computer science, which has received a great deal of attention in both industry and academia. A wide range of efficiency, flexibility, and reliability is offered by cloud computing for providing services and ensuring the servers' availability. When there is a high load for mandatory resources, the platforms are capable of high scalability VM [4]. However, the graphical interface is not always reliable because sometimes it is unable to execute specific requests, such as functions for migration and VM resizing. Another limitation is that there is a lack of enhanced features and active developers compared to other open-source cloud computing platforms. Finally, there is a community around CloudStack with a large number of users. Apache CloudStack provides free, open-source software, as a community-driven project. In the Apache CloudStack ecosystem, an extensive number of service providers, services, and third-party plugins are known to be introduced. In this framework of work, a case study is demonstrated where Apache CloudStack is utilized to establish and automate the end-to-end cloud service infrastructure. Among others, some of the basic functionality CloudStack incorporates is the capability of managing and controlling VM provisioning, resources allocation, networking, and policies [9].

Scalability

Scalability is the preeminent aspect considered. Both the elasticity to scale out and the productivity of the services are pertinent to the study. In cloud environments, auto-scaling and load-balancing mechanisms are implemented to alleviate the rapid adoption of the demand, and to enhance efficient resource utilization, respectively. Due to multi-variate and strong correlations between dependent/adaptive variables, capacity planning works better in blackbox environments [10]. Some methodologies and tools for examining scalability from utility and performance perspectives have been proposed, correlating it to the other properties of elasticity and efficiency. However, no direct algorithm or software models that explicitly connect the incoming workload parameters and the experienced cloud response time behavior of any public cloud platforms are discovered.

Flexibility

Concerning the organization of cloud resources, it is apparent that OpenNebula provides the broadest range of tools needed to set up, manage, and extend a cloud platform for various groups of users. This conclusion includes a system for setting resource usage limits (CPU, memory, system_disk, VM disk, VM network); the option of creating separate groups of users and then associating certain resources, templates, or virtual machines with them; accounting the usage of resources; and providing practical monitoring and statistics data on the cloud resource utilization [9]. Second place in terms to Flexibility is all about Eucalyptus. On the one hand, it allows the use of the libvirt API, which is the standard in the world of the so-called QEMU/KVM and Xen virtualization. Eucalyptus is also distinguished by a large number of services, which allows the cloud platform to be adapted to different changing organizational requirements. Moreover, it is more complicated; however, owing to the more extensive configuration options, Eucalyptus allows for better adjustment of the sizes and functionality on individual servers, increasing the efficiency of the environment. A significant advantage is the focus on maintaining compatibility with the market leader, Amazon AWS. On the one hand, this reduces the costs of training administrators and developers in the management of the cloud, and on the other hand, provides the possibility of relatively easy migration of parts of the cloud to public services. In accordance with the promises, it is perfectly possible to connect instances from the Amazon AWS and Eucalyptus services. Since Eucalyptus is the innovation leader on the market or focuses on maintaining compatibility with Amazon AWS, it is the only open-source cloud platform that offers a ready-made AWS CloudWatch service replacement in its free version. It should be remembered, however, that in a heterogeneous environment based mainly on open-source solutions, the unification process as to the use of closed solutions from the supplier may lead to undesirable dependencies.

Community Support

1. *Introduction:* As researchers investigate how the open source cloud computing phenomenon will progress in the coming years, community support has risen as a recurring topic at conferences and most prominently in up-coming research topics like this one. It is within these communities that the projects are conceived, born, nurtured, and matured. As the lifeline of the project, community governance and core membership is a pillar of evolving open-source projects. However, the importance of secondary and tertiary community mechanisms, along with the individual participant, cannot be overstated. Understanding these facets is crucial to further understanding open-source cloud computing as a whole.

2. *Results:* Community visibility of the project was the desired factor motivating user participation in the project forum. The quality of forum questions, polling options, and email alerts are both seen as areas for enhancement. Three projects were engaged as case studies: CloudStack, Eucalyptus, and OpenStack. Communication with prominent community members and project forum analysis was performed. It is shown that as cloud computing matures within the open source community, the governance and early adoption of software take on predictable patterns, and a generalizable model emerges. For new projects entering this realm of software, that model will be explained, along with the previously undocumented community processes.
3. *Community Support:* Open source is an enigma of efficiency and productivity. It is simple yet extraordinarily complex. It seems recreational or tedious depending on one's vantage, and fundamentally because of this, sometimes shocking or revelatory. Its success or failure appears random or expected and inevitable, and yet, with study, empirical mathematical models, applications, and systems can be derived. Responses are wholly from the community and individual advocate level despite requests for direct communication with the open source project. This demonstrates a deep, ongoing relationship translating what exists into a cohesive understanding: that common problems are addressed within the community and community backed solutions are applied.
4. *Governance Models:* The overall model of engagement and policy provide an adequate governance model. It is generalized to posts rather than focusing on a single issue. New issues are proposed related to participants and contributions, in particular, focusing realization on the spirit of the project or community involvement. Essentially, it is believed in this forum only in relation to the project, at an institutional level, not a fellowship one. The expectation is that any communication will come from a recognized name on the project homepage.

FUTURE TRENDS AND DEVELOPMENTS

The forecast for cloud computing platforms indicates rapid changes in the near future. Consequently, new paradigms and architectures with novel functionalities are fast emerging. Cloud computing platforms have undergone sweeping changes over the past decade [1]. As such, open-source cloud computing platforms are expected to rapidly evolve in order to cater to the new trends and provide advanced offerings. This section is devoted to forecasting trends and emergent developments in open-source cloud computing platforms. Technological advancements and new paradigms are anticipated to feature prominently.

The rise and integration of artificial intelligence (AI) and machine learning techniques are favored to be commonplace features in open-source cloud platforms to offer smart cloud operation in terms of automation, prediction, and decision-making. Besides, the importance of edge computing in open-source cloud is highly envisaged as this technology incurs computation closer to generating data for better performance. Multicloud strategies are becoming increasingly appealing as organizations are striving to avoid vendor lock-in and maximize the benefit from available resources.

RESULTS

Performance Analysis for High-Performance Computing (HPC)

An empirical study evaluated the performance of three popular open-source cloud systems: OpenStack, OpenNebula, and Nimbus, using the HPCC benchmark suite. The research highlighted that while cloud platforms offer scalable and adaptable resources beneficial for HPC tasks, the additional overhead due to virtualization must be considered. The study provided insights into the essential HPC performance metrics across these platforms.

Evolution of Open-Source Cloud Platforms

A case study on OpenStack examined its rapid development cycles, noting that the codebase expanded approximately tenfold within 2½ years since its inception. This rapid growth underscores the dynamic nature of open-source cloud platforms and the importance of understanding their evolving behaviors for effective deployment and optimization.

Comparative Studies

A comparative study evaluated three open-source cloud platforms: CloudStack, OpenStack, and Eucalyptus. The research assessed their viability as open-source cloud solutions for business and research purposes, considering each platform's unique set of features, strengths, and limitations. This evaluation aids users in selecting the most appropriate platform based on specific requirements.

CONCLUSION AND RECOMMENDATIONS

Open-source cloud computing platforms have emerged as viable alternatives to proprietary solutions, offering flexibility, scalability, and cost-effectiveness. The research brings attention to the most significant findings:

1. *Performance Variability:* Open-source platforms like OpenStack, OpenNebula, and CloudStack exhibit differences in resource management, virtualization overhead, and performance, particularly for high-performance computing (HPC) workloads.
2. *Scalability and Evolution:* Platforms such as OpenStack have undergone rapid development, leading to increased complexity and a need for robust community support.
3. *Suitability for Different Use Cases:* Some platforms excel in private cloud deployments (e.g., OpenNebula), while others are better suited for hybrid or public cloud models (e.g., OpenStack).
4. *Security and Management Challenges:* While open-source solutions provide transparency, they require additional security configurations and expertise for effective deployment.

Recommendations

Based on the study's findings, the following recommendations are proposed:

1. *Platform Selection Based on Requirements:* Organizations should choose a cloud platform based on their specific needs, such as scalability (OpenStack), ease of deployment (CloudStack), or lightweight architecture (OpenNebula).
2. *Performance Optimization:* To improve efficiency, organizations should implement optimizations such as containerization, hardware acceleration, and fine-tuned resource allocation.
3. *Security Enhancements:* Additional security measures, including regular updates, encryption, and access control policies, should be adopted to mitigate potential vulnerabilities.
4. *Community and Vendor Support:* Organizations should leverage community forums, vendor-backed distributions, and professional support services to ensure smooth implementation and maintenance.
5. *Future Research and Development:* Continuous evaluation of emerging technologies such as edge computing, Kubernetes-based cloud orchestration, and AI-driven cloud management will help improve open-source cloud adoption.

By following these recommendations, businesses, researchers, and IT professionals can maximize the benefits of open-source cloud platforms while mitigating potential risks.

REFERENCES

1. Buyya R, Pandey S, Vecchiola C. Market-oriented cloud computing and the Cloudbus toolkit. Large Scale Network-Centric Distributed Systems. Wiley; 2013 Oct 18; 319–58.
2. Salih NK, Zang T. Survey and comparison for Open and closed sources in cloud computing. arXiv preprint arXiv:1207.5480. 2012 Jul 23.
3. Bhuiyan MY, Othman SH, Radzi RZ. An enhancement of TOE model by investigating the influential factors of cloud adoption security objectives. Int J Innov Comput. 2019 May 31; 9(1): 55–67.
4. Shirazi F, Seddighi A, Iqbal A. Cloud computing security and privacy: an empirical study. InHuman-Computer Interaction. Interaction Contexts: 19th International Conference, HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017, Proceedings, Part II 19. Springer International Publishing; 2017; 534–549.

5. Farooq MS, Chaudhry AH, Shafiq M, Berhanu G. Factors affecting students' quality of academic performance: A case of secondary school level. J Qual Technol Manag. 2011 Dec; 7(2): 1–4.
6. Rodriguez MA, Buyya R. Container-based cluster orchestration systems: A taxonomy and future directions. Software: Pract Exper. 2019 May; 49(5): 698–719.
7. 8Ciptaningtyas HT, Hariadi RR, Husni M, Ghozali K, Sholikah RW, Setyadharma IM. OpenStack Implementation using Multinode Deployment Method for Private Cloud Computing Infrastructure. In 2023 IEEE International Seminar on Intelligent Technology and Its Applications (ISITIA). 2023 Jul 26; 12–17.
8. Łątkowski M, Nowak R. OpenStack and Google Cloud performance comparison in Infrastructure as a Service model. arXiv preprint arXiv:2210.09691. 2022 Oct 18.
9. Barkat M, Nibou D, Amokrane S, Chegrouche S, Mellah A. Uranium (VI) adsorption on synthesized 4A and P1 zeolites: equilibrium, kinetic, and thermodynamic studies. CR Chim. 2015; 18(3): 261–9.
10. Al-Said Ahmad A, Andras P. Scalability analysis comparisons of cloud-based software services. J Cloud Comput. 2019 Jul 23; 8(1): 10.

