**Indonesian Journal of Computer Science**
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

# Harnessing the Power of Distributed Systems for Scalable Cloud Computing: A Review of Advances and Challenges

**Hanan A. Taher[1], Subhi R. M. Zeebaree[2]**
hanan.taher@dpu.edu.krd, subhi.rafeeq@dpu.edu.krd
[1] ITM Dept., Technical College of Administration, Duhok Polytechnic University, Duhok, Iraq
[2] Energy Eng. Dept., Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq

**Article Information**

Submitted : 8 Mar 2024
Reviewed: 14 Mar 2024
Accepted : 1 Apr 2024

**Keywords**

Cloud Computing, Fault Tolerance, Proactive Techniques, Predictive Techniques, Cloud Availability.

**Abstract**

In the realm of cloud computing, the literature defines scalability as the inherent ability of a system, application, or infrastructure to adapt and accommodate varying workloads or demands efficiently. It encompasses the system's capability to handle increased or decreased usage with compromising performance, responsiveness, or stability. In this paper, a comprehensive review is presented regarding the scalability in the cloud computing network. In addition, the research community define the scalability as a dynamic attribute, emphasizing its ability to facilitate both horizontal and vertical scaling. Horizontal scalability involves adding or removing instances or nodes to distribute workloads across multiple resources, while vertical scalability focuses on enhancing the capacity of existing resources within a single entity. They established a global frameworks to evaluate scalability, often emphasizing response time, throughput, resource utilization, and cost-efficiency as critical metrics. These metrics serve as benchmarks to assess the system's ability to scale effectively without compromising performance or incurring unnecessary costs [1]. The literature underscores scalability's interconnectedness with elasticity, highlighting the need for on-demand resource provisioning and de-provisioning to maintain an agile and adaptable infrastructure. Overall, in academic papers, cloud scalability is portrayed as a fundamental attribute crucial for modern computing infrastructures, enabling systems to flexibly and efficiently adapt to dynamic computing needs.

## A. Introduction

The utilization of computing tasks in remote computing clusters has become a prevalent practice as a result of the rapid expansion of distributed systems and the increasing requirements for computing power. This approach employs a strategy of partitioning the problem into distinct components and processing each component independently. This allows each processing element to focus only on its assigned portion of the problem simultaneously [2]. Different software frameworks are used to improve the efficiency of distributed systems in these situations. These frameworks are employed for the administration of large-scale data storage. Hadoop is an exceptionally beneficial software framework for harnessing data in distributed systems. This software streamlines the process of forming machine clusters and allocating tasks among them, while guaranteeing adherence to appropriate formatting. Hadoop comprises two primary elements: the Hadoop Distributed File System (HDFS) and MapReduce (MR). By leveraging Hadoop, we can efficiently analyze, compute, and distribute the frequencies of individual words in a sizable file, enabling us to ascertain the significance of each phrase. The authors in study [3] provided a comprehensive explanation of this particular software and its structure. The traditional model of a general-purpose computer that operates independently is being substituted with a more varied, transparent, and adaptable paradigm. The broadening of the cloud computing concept, coupled with alterations in its industry framework, has radically revolutionized the service paradigm. Cloud computing is available in three distinct forms: the public cloud, the private cloud, and the hybrid cloud [4]. Furthermore, a variety of services are offered, which may be categorized into three main classifications: (a) Software as a Service (SaaS), (b) Infrastructure as a Service (IaaS), and (c) Platform as a Service (PaaS) [5]. Although SaaS, PaaS, and IaaS technologies improve the convenience of applications, a critical issue in cloud computing is how to offer distinct services to clients and enable them to select their desired service level on various devices or platforms. Therefore, it is essential to give priority to the functioning of the cloud computing platform in order to enable comprehensive evaluation and improvement [6]. Cloud computing designers must prioritize scalability as a critical consideration. Scalability is employed to determine if a cloud system can handle a substantial volume of simultaneous application requests [7]. The scholarly community dedicated significant attention to this subject. Hwang et al. [8] introduced the importance of scalability in cloud computing. Chen et al. [9] examined the ability of video streaming solutions to handle increasing demands in a multi-cloud environment. Anandhi and Chitra [10] examined the scalability of a cloud database by considering the consistency attribute. Tian et al. [11] employed a stochastic service decision network model to examine the scalability of cloud computing in terms of performance, with energy consumption serving as a limiting factor. Scalability

benchmarking is crucial in cloud computing networks as it allows for the dynamic allocation of resources in response to varying workloads, ensuring adaptability and real-time adjustments [12] [13]. This adaptability ensures systems can meet varying demands without compromising performance or incurring downtime [14]. Also, Cloud scalability directly impacts performance optimization. By efficiently allocating resources, systems maintain responsiveness even during peak usage, ensuring consistent service levels and user experiences. In addition, scalability promotes cost efficiency by allowing resource provisioning based on actual demand. It mitigates over-provisioning, preventing unnecessary expenses during low-usage periods, thus optimizing cost-effectiveness. The contributions of this paper are to dive in studies regarding the scalability within cloud computing, determining the suitable metrics that increase the effeminacy of the systems, and listing the limitations and challenges in this field.

## B. Background and Preliminary

### 1. The mechanism of cloud computing

Efficiently distributing tasks and resources is a crucial area of research in cloud computing systems, which must accommodate a high number of users and workloads. This necessitates the design of a scheduling mechanism that can scale effectively [15]. Within the realm of cloud computing, clients make requests for a variety of resources, such as compute capacity and access to storage. The request type of a specific device is determined by its access mode and may vary from the requests made by other devices. The request can be transmitted either by wireless broadcasting or through unicast/multicast in a specified local area network (LAN). The request scheduling layer has the capability to assign many servers to meet the needs of a client, allowing for the representation of various configurations. For instance, a specific broadcast request may be viewed as a scenario in which this device is controlled by all the servers in the request schedule layer and consistently transmits the requests to them concurrently [16]. Considering the disparities in the duration it takes for links to numerous servers and other external factors, the broadcast request can be seen as an assemblage of devices supervised by a single server.

Cloud-hosted virtual servers or clustered systems efficiently deliver computational services and manage virtual resources, guaranteeing rapid client answers [17]. Upon receiving the scheduling request, clients are allocated virtual servers that are tailored to their precise computational and storage needs. Concurrently, every virtual server provides information about its current state to a scheduling layer [18]. If a status report is not received within the specified interval, the request scheduling system considers it as a passive timeout report, indicating the failure of a certain virtual server. The virtual server will undergo live migration based on the virtual management requirements or as advised by the request

scheduling layer. The former typically arises from issues related to load and power management. However, the request scheduling layer oversees the rearrangement of resource distribution in order to meet client demands or improve performance [19] [20].

## 2. Scalability in Cloud Computing

Figure 1 illustrates the performance of two distinct systems, F and G. The example illustrates the relationship between the average response time and the number of requests that the system needs to handle every hour. Thus, the system is evaluated based on a specific frequency of inquiries over an extended duration. System F has superior performance compared to G for both 100 and 200 requests per hour. Nevertheless, G surpasses F in performance when the number of requests per hour approaches 300 [21]. Based on Figure 1, it is not evident which system exhibits superior scalability. However, it appears that system G possesses more favorable scaling characteristics. In order to address such arguments, it is necessary to have a precise understanding of scalability.

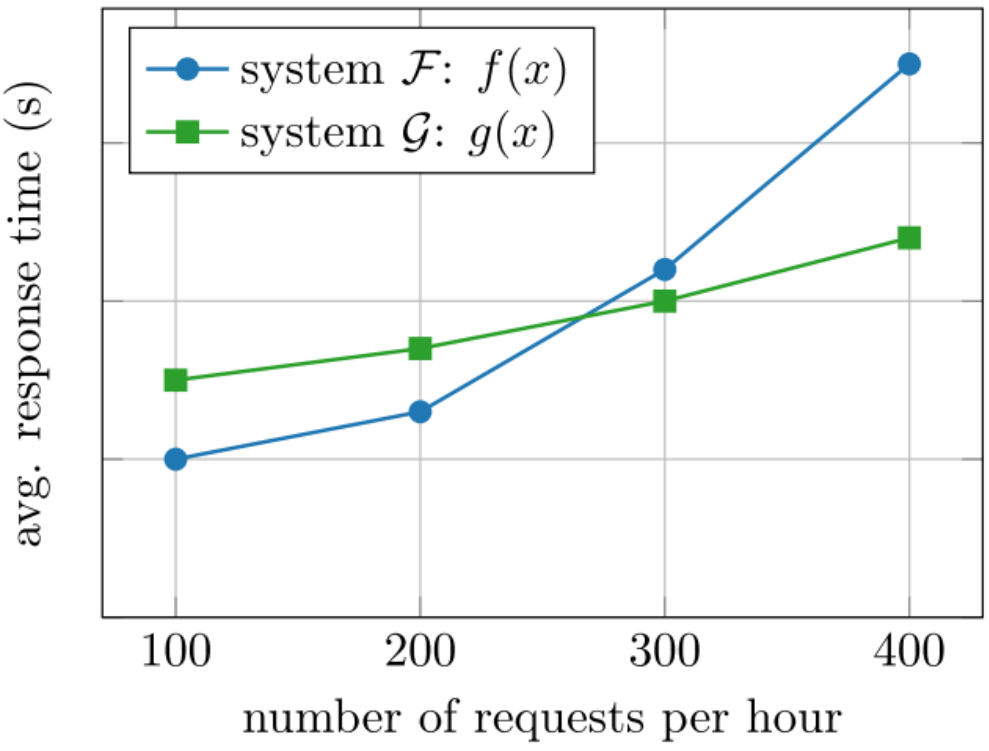


**Figure 1.** Shows the performance graphs for two systems, F and G. [21].

## 3. Definition and metrics on scalability

When evaluating the scalability of a system, it is essential to establish a clear and unequivocal definition of scalability. This part explores several facets of scalability and provides a thorough explanation of the formal concept of scalability for cloud applications, which serves as the foundation of this study. According to Henning and Hasselbring [22], scalability is the capacity of a system to fulfill its service level objectives for all levels of demand within a specified range, maybe by utilizing more resources. The notion is derived from the definition provided by Herbst et al. [23], wherein scalability of cloud systems refers to the system's ability to handle increasing workloads while maintaining satisfactory performance, which includes the incorporation of hardware resources. Scalability is based on the different attributes described by Weber et al. [24] as follow:

1. **Load Intensity:** refers to the workload that a system needs to manage.
2. **Provisioned Resources:** explain the least computational resources necessary to handle a specific level of workload intensity.
3. **Service Level Objects (SLOs):** specify the minimum performance standards that the system being tested must meet.
4. **Efficiency**: two fundamental principles constrain the efficiency of a distributed system:
5. **Latency:** it is the time that required for a message to travel between two points in a distributed system.
6. **Bandwidth:** refers to the rate at which data may be transmitted or transferred during a given time period while maintaining a consistent condition. Furthermore, it is imperative to incorporate two prominent indicators of efficiency in this context:
7. **Response Time:** refers to the duration required to receive a result following the submission of a job for processing by the system.
8. **Throughput:** refers to the capacity of a system to efficiently process and handle large amounts of data or tasks. Put simply, it refers to the rate at which things are completed within a given timeframe. The objective of the distributed computing system is to optimize performance by minimizing latency and reaction time while simultaneously boosting throughput [25].
9. **Elasticity**: Conversely, it assesses the dynamic alterations of a system and measures the system's capacity to adjust itself across shorter time periods. Elasticity refers to the system's capacity to dynamically adjust its resource allocation in response to changes in workload, ensuring that the available resources closely align with the current demand at any given time [23]. Elastic computing is the process of adjusting the capacity of computer resources to align with a fluctuating workload, hence ensuring optimal use. Elasticity in communication networks pertains to the network's capacity to modify its functioning and redistribute resources (resource supply) in

accordance with temporal and spatial variations in traffic and service demand (resource demand). This may entail the utilization of computer and communications resources, specifically for the purpose of overseeing computational resources in software and virtualized networks, such as those employed in 5G systems. Figure 2 illustrates the temporal elasticity of a system by contrasting the system's demand and supply of resources. Scalability refers to the ability to add resources to a system. Nevertheless, the definition fails to specify the process by which these resources are included.

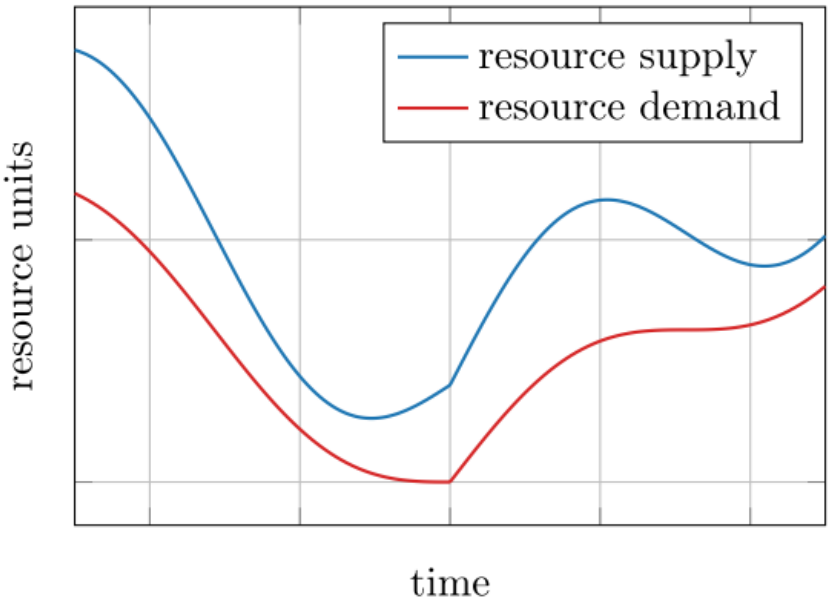


**Figure 2.** The elastic relationship between resource supply and demand [23].

There are two methods for adding resources: vertical and horizontal scaling.

1. **Vertical scaling:** Vertical scaling refers to the process of increasing the resources allocated to individual processing nodes. An effective approach to achieve this is by enhancing the system with supplementary hardware. Implementing vertical scaling is typically straightforward as it often does not necessitate a specific system architecture. However, to enable the use of additional CPUs or CPU cores, it is essential for the software to support parallel processing. Moreover, it is crucial to acknowledge that vertical scaling is constrained by the economic impracticality of operating high-performance technology and the presence of fundamental physical constraints.
2. **Horizontal scaling:** Horizontal scaling refers to the process of augmenting a distributed system by including additional computing nodes. Horizontal scaling offers the advantage of unrestricted expansion and is

generally more cost-effective compared to vertical scaling, as it does not require expensive high-end equipment. Horizontal scaling has a drawback in that it is not universally compatible with all systems. Moreover, a higher frequency of occurrences can lead to issues about uniformity or accessibility.

**The challenges and limitations of cloud scalability**

Scalability is a pivotal aspect of cloud computing, enabling businesses to flexibly expand or contract their resources based on demand. While the cloud offers remarkable scalability advantages, it also presents a myriad of challenges and limitations that organizations must navigate to ensure optimal performance and reliability [26][27][28].

## 4. Challenges of Cloud Scalability

1. **Architectural Complexity:** cloud scalability is often impeded by the complexity of the architecture itself. Distributed systems, varying infrastructure, and interdependencies among components can complicate scaling efforts. As applications grow, managing these complexities becomes increasingly challenging [29].
2. **Cost Management:** while scalability provides the ability to adjust resources dynamically, it can also lead to cost inefficiencies. Over-provisioning resources might occur, especially when scaling anticipates future demand inaccurately, resulting in unnecessary expenses.
3. **Performance Bottlenecks:** scaling applications might reveal performance bottlenecks that were previously hidden. Issues like latency, network congestion, or limitations in underlying infrastructure can surface when scaling, affecting overall system performance.
4. **Data Management and Storage:** managing big data in a scalable manner is a significant challenge. Ensuring data consistency, integrity, and accessibility across distributed systems becomes complex as data storage needs expand [30].
5. **Security Concerns:** scalability can potentially heighten security risks. As the system expands, there might be more entry points for attackers, and ensuring robust security measures across all components becomes crucial but challenging [31] [32].

## 5. Limitations of Cloud Scalability

1. **Resource Constraints:** despite the promise of infinite scalability, cloud providers still have resource constraints. Instances of sudden, massive scaling might hit these limits, causing disruptions or service degradation.

2. **Legacy Systems Integration:** integrating legacy systems with scalable cloud infrastructure poses a challenge. Often, older applications and infrastructure lack compatibility with modern cloud environments, impeding seamless scalability [33].
3. **Scalability Planning:** accurately predicting future scaling needs is challenging. Organizations must strike a balance between under-provisioning (which can lead to performance issues during traffic spikes) and over-provisioning (resulting in increased costs during periods of lower demand).
4. **Operational Complexity:** scalability introduces operational complexities, especially in terms of monitoring, configuration, and management. Orchestrating and maintaining large-scale systems require specialized skills and robust tooling [34].

### 6. Addressing Scalability Challenges

To mitigate these challenges and limitations, organizations must adopt several strategies:

1. **Design for Scalability:** Building applications with scalability in mind from the outset can streamline future scaling efforts.
2. **Automate Scaling Processes:** implementing automated scaling mechanisms based on predefined metrics helps in efficient resource allocation.
3. **Utilize Hybrid Cloud Solutions:** The integration of public and private cloud services offers more adaptability and reduces the potential for vendor lock-in.
4. **Continuous Monitoring and Optimization:** regularly monitor and optimize infrastructure to identify and address performance bottlenecks and cost inefficiencies [35].

Cloud scalability is undeniably a transformative capability, yet it demands careful planning, proactive management, and continuous optimization to harness its full potential while navigating its inherent challenges and limitations.

## C. Literature Survey

Scalability is a crucial aspect in evaluating the cost-efficiency relationship in large-scale distributed service systems. Numerous research have been undertaken on this subject for each distinct system, although only a handful of them have tackled the overall definition of scalability itself.

The authors Brataas et al. [36] have defined scalability as "the capacity of a service to expand its capabilities by utilizing additional resources within the resource space." This definition has been further expanded upon by Lehrig et al.

[37]. They introduced two metric functions, namely resource and cost, to measure scalability. The resource scalability meter function demonstrates the relationship between the capacity of a cloud software service and its utilization of cloud resources. This statistic is primarily related to a particular kind of cloud resource, such as the specifications of application virtual machine (VM) instances. Upon the introduction of novel cloud resources, the cost scalability metric function is deployed. This function establishes the relationship between the capability of a cloud-based software service and the expenditure on cloud resources. The suggested metric functions enable researchers to analyze the impact of job features and quality criteria on the scalability of the service. The scalability of the Cloud Store's measure functions was evaluated through empirical analysis in both the public Amazon Web Services (AWS) and a private OpenStack-based environment. There were a total of 53 measurements taken for 21 different AWS settings. The results indicated that the proposed scalability metric functions provided thorough insights, including the identification of cost-efficient resource combinations for a specific capacity. In a prior publication [38], the authors presented a theoretical approach that specifically addresses the issue of scalability in mobile multi-agent systems. Nevertheless, it is crucial to acknowledge that this framework is confined to theoretical notions and modeling outcomes.

In addition, a study [39] established standardized metrics to assess the scalability of stream processing engines. The authors suggested two measurements based on established concepts of scalability in cloud computing: a load capacity function and a resource demand function. Both measures assess the allocation of resources and the intensity of demand, while also ensuring the attainment of specific service level goals. Their demonstration showcased the use of these metrics for scalability benchmarking and examined their superiority compared to commonly employed metrics in stream processing engines and other software systems.

Similarly, a study [40] proposed a Theodolite approach for measuring distributed system scalability in cloud contexts, as well as how resource requirement grows with increasing workloads. This study laid the groundwork for undertaking extensive cloud scalability evaluations. Moreover, the proposed approach may be used to compare different stream processing engines and assess their scalability across different workload dimensions.

Another study [41] presented a detailed and formal representation of the scalability of the request scheduling process in cloud computing. It provided a clear explanation of the scalability notion. The authors assessed the scalability of the scheduling server by modeling it as a stochastic preemptive priority queue. The research encompassed both theoretical and numerical methodologies, exploring many structures and diverse environmental configurations.

De Donno et al. [42] classified scalability into two distinct categories: scale-up elasticity and scale-down elasticity. Scale-up elasticity entails the incorporation of extensive processing, storage, and network resources to fulfill user needs. Its main objective is not cost optimization, but rather fulfilling operational necessities. In contrast, the ability to scale down allows for the steady-state expenditure to closely align with the steady-state workload, particularly in small- and medium-sized firms and organizations that have limited computing resources [43].

Tiwari et al. [44] demonstrated the ability of multiprocessor systems to handle larger workloads, as evaluated by the efficiency measured in terms of speedup. The execution time on a system with a size of n and k processors is denoted as Time (k, n). The ideas of speedup and efficiency can be precisely described as follows: The speedup (k, n) is the quotient of the time required to finish a task using one processor and the time required to finish the same work using k processors. The efficiency (k) is the speedup (k, n) divided by k. The Universal Scalability Law is a precise performance model that appropriately characterizes the scalability of universal systems. This study investigates the practicality of integrating the Universal Scalability Law into the Theodolite benchmarking framework for cloud-native stream processing systems. Theodolite evaluates scalability by employing service level objectives (SLOs) [45].

Similarly, a study [46] proposed employing the Universal Scalability Law, a performance model that precisely characterizes the scalability of universal systems. The study investigated the practicality of integrating the Universal Scalability Law into the Theodolite benchmarking framework for cloud-native stream processing systems. Theodolite evaluated scalability by employing service level objectives (SLOs).

Two criteria, latency and throughput, were introduced by other authors [47] to assess the scalability of cloud computing services. These measurements are frequently employed for this purpose, as indicated by a study [48]. Furthermore, the study observed the utilization of CPU and network resources to assess the effectiveness of cloud services. The researchers employed three cloud services, specifically Apache Flink, Apache Storm, and Apache Spark Streaming. The findings indicate that Storm and Flink exhibit comparable performance, whereas Spark Streaming exhibits significantly higher latency but offers greater throughput.

Other authors proposed applying cloud computing approach in an area that has lack in such technology. This study presented and implemented a model of Human Resource (HR) for management system to address HR issues in this domain by leveraging Cloud Technology. The suggested solution has sixteen basic modules and was constructed by utilizing many technologies [49][50].

The purpose of this study [51] was to gather data on the performance of a cutting-edge stream processing framework in relation to various execution parameters and scalability factors. The authors systematically evaluated the

scalability of five contemporary stream processing systems. A comprehensive series of experiments were carried out on Kubernetes clusters in both the Google cloud and a private cloud [52]. Throughout the testing, they effectively deployed and ran a maximum of 110 micro service instances simultaneously, with each instance capable of processing up to one million messages per second. Each of the benchmarked frameworks demonstrates a nearly proportional increase in performance when an adequate amount of cloud resources are allocated. Nevertheless, the frameworks exhibit significant disparities in the rate at which additional resources must be incorporated to handle a growing workload.

Conversely, a research publication [22] developed a benchmarking technique that enables the research community to do empirical scalability assessments of cloud-native apps and frameworks. The effort involved conducting scalability measurements, developing quantification techniques, and creating a scalability benchmarking tool specifically designed for cloud systems. Several separate tests were undertaken to assess whether the specified service level goals (SLOs) are met under different combinations of load and resources. This benchmarking methodology provides the chance to adjust the configuration in a manner that attains a harmonious combination of user-friendliness and the capacity to replicate results. Additionally, it enables providers to choose the balance between the overall execution duration and statistical grounding. The objective was accomplished by employing two stream processing frameworks, Kafka Streams and Flink, alongside conducting investigations in two public clouds and one private cloud. The results showed that regardless of the cloud platform utilized, determining the requirements for Service Level Objectives (SLOs) only necessitated a limited number of iterations (5) and a short execution time (5 minutes) [53]. The study concluded that the proposed method allows for the evaluation of scalability within a reasonable duration. The study suggested enhancing the efficiency of benchmark execution by incorporating a heuristic based on the Universal Scalability Law (USL). Furthermore, there is a requirement for a tool that enables the analysis of benchmark results in relation to the Upper Specification Limit (USL).

In reference to [54], The authors suggest a scalable cloud-native architecture, known as a cloud-native solution, for a mobility management entity. The premise of this statement is that incorporating virtualization into the central network of the 5G cloud native architecture can significantly decrease the expenses associated with implementation. The primary goal of this design is to operate as a data production center utilizing micro services. The design provides excellent scalability and the capability to automatically adjust the required micro services, allowing for load balancing of the entire system. Similarly, the authors in study [55] examined the concept of database management system (DBMS) by conducting a comprehensive literature review and systematic analysis. It investigated the

principles of distributed database management systems (DDBMS) and identified suitable architectures for DDBMS solutions. The article also offers justified recommendations based on the needs and perceptions of users.

Many studies made a survey about distributes systems and how they impact on the efficiency of such systems. For example, a research [56] investigated the impact of the distributed-memory parallel processing technique on enhancing the efficiency of multicomputer multicore systems. Furthermore, several methodologies have been presented for utilization in distributed-memory systems [57]. The objective of the study was to ascertain the optimal strategy for improving the performance of multicore processors in distributed systems. The most efficient strategies were employing an operating system known as gun/Linux 4.8.0-36, an Intel Xeon 2.5 CPU, and the python programming language. The authors in these studies [58] [59] Implemented a model to facilitate users in doing composite jobs interactively with little processing time. Distributed-Parallel-Processing and Cloud Computing are two highly advanced technologies that excel in rapidly processing and resolving customer problems. The proposed system exhibited enhanced efficiency and surpassed other systems in terms of parallel processing.

In recent years, there has been a significant increase in the utilization of machine learning, deep learning, and reinforcement learning algorithms. Machine learning enables researchers and engineers to quickly predict and estimate the best answers to research challenges. Machine learning is progressively shifting towards operating on cloud-native systems, primarily because of the advantages offered by the cloud's vast processing resources for the learning process [54]. The driving force behind this trend is the enhancement of software service design. The scalability and flexibility of cloud native architecture efficiently cater to the resource requirements of machine learning. While cloud native features and benefits can improve machine learning operations, they nevertheless face issues in load balancing [60].

### D. Discussion and Comparison

Based on the reviewing of studies in the literature, there were different opinions about a precise definition of the scalability and how this metric can be evaluated to obtain high performance for the cloud computing systems. For instances, studies [36], [37], [39], [40] defined the scalability based on parameters which are capacity of the system to consume resources and cost. The authors claimed that when new cloud resources are created, the cost scalability metric function is deployed.

This function denotes the correlation between the capability of a cloud software service and the expenditure of the cloud resources it necessitates. The suggested metric functions enable researchers to analyze the impact of task parameters and quality standards on the scalability of the service. Other research

[42], [43] proposed the utilization of an elasticity parameter to characterize the scalability in two modes: scale-up elasticity and scale-down elasticity. Significant processing, storage, and network resources were combined to meet the demands of users in terms of scalability, which is a necessity for operations rather than a means of cost reduction. Scale-down elasticity ensures that steady-state expenditure is in line with workload, particularly in small- and medium-sized enterprises and organizations that have limited computing resources. In investigations [44], [46], the authors introduced the Universal Scalability Law, a performance model that effectively characterizes the scalability of universal systems. Scalability is evaluated by the legislation using service level objectives (SLOs). Other authors [47], [48] presented two criteria, latency and throughput, to assess the scalability of cloud computing services. However, previous studies [22], [51] have introduced a benchmarking approach that enables researchers to do empirical scalability assessments of cloud-native apps and frameworks. These frameworks offer the chance to customize the configuration in a manner that achieves a harmonious blend of user-friendliness and consistency. Additionally, it enables providers to choose the balance between the overall execution duration and statistical grounding. In recent years, there has been a growing trend in the utilization of machine learning, deep learning, and reinforcement learning algorithms [61]. Machine learning enables researchers and engineers to quickly predict and estimate the best possible solutions to research challenges. Table 1 displays the synopsis of the literature survey.

**Table 1**. The Summary of Studies on the Literature Survey

| **Reference** | **Metrics** | **Implementation** | **Results** |
|---|---|---|---|
| [36], 2017 | Resource and cost | In public and private clouds | The suggested scalability metric functions offer comprehensive analysis by discovering the most cost-effective combinations of resources for a given capacity. |
| [41], 2019 | Configuration of different structural parameters | Executed the task scheduling procedure in the domain of public cloud computing. | Analyzing, comparing, and drawing conclusions can serve as a significant point of reference for designing cloud networks |
| [42], 2019 | The concepts of scale-up and scale-down elasticity | Explored the fundamental distinctions among Cloud computing, Fog computing, Edge computing, and IoT, and their interconnectedness in the progression of the | The results showed the significance of Fog computing and asserted that Fog server as the cohesive element that connects IoT, Cloud, and Edge computing. |

| | | | |
|---|---|---|---|
| | | computing paradigm. | |
| [47], 2019 | Latency and throughput | Apache Flink, Apache Storm, and Apache Spark which are private clouds. | Our results demonstrate that Storm and Flink exhibit comparable performance, whereas Spark Streaming exhibits significantly higher latency despite offering greater throughput. |
| [49], 2020 | _ | Elastic Compute Cloud (EC2) and Amazon Web Service (AWS). | The method make the small and medium Businesses to handle their finances and paperwork. The method also made communication better and saved time, money, and effort. |
| [40], 2020 | Resources demand and workloads | Developed a comprehensive framework for evaluating the scalability of a system, this can be utilized to execute specific benchmarks for a given use case and workload magnitude. The study employed Kafka Stream clusters. | The study demonstrated that our benchmarking technique is capable of assessing the stream processing engines' scalability using various workloads. Furthermore, our results demonstrate that Kafka Streams exhibits scalability across many use cases and deployment options. |
| [44], 2021 | Execution time, utilization ratio and power consumption | In virtual machines (VMs) and cloudlets, parameters are generated for the Shortest Job First (SJF), Hungarian, and First-Come, First-Served (FCFS) methods. | The suggested algorithm achieved higher speed in utilization ratio than other methods. |
| [39], 2021 | Load capacity function and a resource demand function | Used private clouds networks: Kafka Streams and Flink | The results showed the same performance for both Kafka Streams and Flink at reasonable demand and capacity, but they presented different behavior after increasing the metrics. |
| [22], 2022 | Loads and Resources | There are two frameworks for processing streams. The study utilized Kafka Streams and Flink frameworks across one private cloud and two | The findings indicated that, irrespective of the cloud platform used, it only required a small number of iterations (5) and a brief duration of execution (5 minutes) to ascertain the needs for Service Level Objectives (SLOs). |

| | | | |
|---|---|---|---|
| | | public clouds. | |
| [46], 2022 | Recourses and workload | Theodolite uses service level goals (SLOs) to evaluate scalability. | The findings indicate that certain ExplorViz micro services exhibit linear scalability, but the overall scalability of the system is mostly limited by a single micro service. |
| [51], 2023 | Resource demands | Conducted trials for a total of 740 hours using Kubernetes deployed on a private cloud and the Google cloud. | Each of the benchmarked frameworks demonstrates a nearly proportional increase in performance as long as an adequate amount of cloud resources are allocated. |
| [54], 2023 | Resource management | Conducted a thorough analysis and comparison of resource management in edge computing that is cloud native. | The topics discussed included resource management through the use of software architecture, network slicing, virtualized network operations, and containerization. In addition, they made forecasts on cloud native research trends and other issues. |

## E. Recommendations

In cloud computing, scalability is essential for effectively managing a range of workloads. The following recommendations will help to guarantee scalability:

- Employ Auto Scaling: enabling cloud service capabilities such as Auto Scaling to automatically modify resources in response to demand. This makes it easier to handle traffic fluctuations without the need for human intervention.
- Micro services Architecture: using a micro services-based methodology when designing applications. This makes it possible to scale separate components independently, optimizing the use of resources.
- Load balancing: Using load balancers to load balance inbound traffic across multiple servers or instances. This keeps certain servers from experiencing overload and guarantees equitable resource use.
- Elastic Storage Solutions: Select expandable storage options such as Azure Blob Storage, Google Cloud Storage, or Amazon S3. These services scale automatically in response to storage requirements.
- Horizontal Scaling: designing systems to scale horizontally by adding more instances, instead of depending just on vertical scaling that raises the power of individual instances.
- Analytics and Monitoring: setting reliable monitoring technologies to monitor system performance.

- Fault Tolerance: Implementing redundant systems, failover mechanisms, and backup solutions to ensure continuous operation during failures.
- Optimize Resource Utilization: Regularly analyze resource usage and optimize configurations to ensure efficient utilization. Remove unused or underutilized resources to save costs.
- Testing for Scalability: Perform regular load testing to identify bottlenecks and limitations in the system. Use this data to improve scalability.
- Cost management: Scalability frequently results in higher expenses. Continually track and control spending with the help of cost analysis tools offered by cloud service providers.

These recommendations can greatly improve the scalability of the cloud-based systems and can efficiently manage different workloads while maximizing efficiency and cost.

## F. Conclusions and Future Trends

This study presented papers in the scholar that were extensively detailed the metrics and measurements regarding scalability assessments, emphasizing the significance of response time, throughput, resource utilization, and cost-effectiveness as critical evaluative criteria. These metrics serve as guiding that enabling researchers and providers to test a system's ability to scale effectively without losing performance or cost. Significant processing, storage, and network resources were combined to meet customer demands for scale-up elasticity, which is a necessary operational aspect rather than a cost-saving measure. Scale-down elasticity ensures that steady-state expenditure is in line with workload, particularly in small- and medium-sized enterprises and organizations that have limited computing resources. In recent times, there has been an increasing trend in the utilization of machine learning and deep learning algorithms. Machine learning empowers researchers and engineers to rapidly forecast and approximate optimal solutions to research problems. Therefore, for future work, cloud-based machine learning methods enhance scalability by accurately forecasting demand patterns and efficiently allocating resources to accommodate different workloads. They implement auto-scaling methods that dynamically adapt resources, assuring optimal usage and cost-efficiency. These algorithms optimize performance by evaluating data, enhancing operations during scaling tasks. In addition, machine learning enables cloud systems to dynamically adjust and forecast their scaling capabilities.

## G. References

[1] L. M. Haji, S. R. M. Zeebaree, K. Jacksi, and D. Q. Zeebaree, "A State of Art

Survey for OS Performance Improvement," *Sci. J. Univ. Zakho*, vol. 6, no. 3, pp. 118–123, 2018.

[2] S. R. M. Zeebaree, A. B. Sallow, B. K. Hussan, and S. M. Ali, "Design and Simulation of High-Speed Parallel/Sequential Simplified DES Code Breaking Based on FPGA," in *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 2019, pp. 76–81. doi: 10.1109/ICOASE.2019.8723792.

[3] H. M. Zangana and S. R. M. Zeebaree, "Distributed Systems for Artificial Intelligence in Cloud Computing: A Review of AI-Powered Applications and Services," *Int. J. Informatics, Inf. Syst. Comput. Eng.*, vol. 5, no. 1, pp. 1–20, 2024.

[4] I. M. Ibrahim, S. R. M. Zeebaree, H. M. Yasin, M. A. M. Sadeeq, H. M. Shukur, and A. Alkhayyat, "Hybrid Client/Server Peer to Peer Multitier Video Streaming," in *2021 International Conference on Advanced Computer Applications (ACA)*, IEEE, 2021, pp. 84–89.

[5] S. Mustafa, B. Nazir, A. Hayat, A. ur R. Khan, and S. A. Madani, "Resource management in cloud computing: Taxonomy, prospects, and challenges," *Comput. Electr. Eng.*, vol. 47, pp. 186–203, 2015, doi: https://doi.org/10.1016/j.compeleceng.2015.07.021.

[6] S. Akhshabi and C. Dovrolis, "The evolution of layered protocol stacks leads to an hourglass-shaped architecture," in *Dynamics On and Of Complex Networks, Volume 2: Applications to Time-Varying Dynamical Systems*, Springer, 2013, pp. 55–88.

[7] Z. Li, Y. Zhang, and Y. Liu, "Towards a full-stack devops environment (platform-as-a-service) for cloud-hosted applications," *Tsinghua Sci. Technol.*, vol. 22, no. 01, pp. 1–9, 2017.

[8] K. Hwang, X. Bai, Y. Shi, M. Li, W.-G. Chen, and Y. Wu, "Cloud performance modeling with benchmark evaluation of elastic scaling strategies," *IEEE Trans. parallel Distrib. Syst.*, vol. 27, no. 1, pp. 130–143, 2015.

[9] W. Chen, J. Cao, and Y. Wan, "QoS-aware virtual machine scheduling for video streaming services in multi-cloud," *Tsinghua Sci. Technol.*, vol. 18, no. 3, pp. 308–317, 2013.

[10] R. Anandhi and K. Chitra, "A challenge in improving the consistency of transactions in cloud databases-scalability," *Int. J. Comput. Appl.*, vol. 52, no. 2, pp. 12–14, 2012.

[11] Y. Tian, C. Lin, Z. Chen, J. Wan, and X. Peng, "Performance evaluation and dynamic optimization of speed scaling on web servers in cloud computing," *Tsinghua Sci. Technol.*, vol. 18, no. 3, pp. 298–307, 2013.

[12] I. M. I. Zebari, S. R. M. Zeebaree, and H. M. Yasin, "Real time video streaming from multi-source using client-server for video distribution," in *2019 4th Scientific International Conference Najaf (SICN)*, IEEE, 2019, pp. 109–114.

[13] H. Fawzia *et al.*, "A REVIEW OF AUTOMATED DECISION SUPPORT TECHNIQUES FOR IMPROVING TILLAGE OPERATIONS," *Rev. AUS*, vol. 26, pp. 219–240, 2019.

[14] D. Thakkalapelli, "Cost Analysis of Cloud Migration for Small Businesses," *Tuijin Jishu/Journal Propuls. Technol.*, vol. 44, no. 4, pp. 4778–4784, 2023.

[15] Z. Zhong, K. Chen, X. Zhai, and S. Zhou, "Virtual machine-based task scheduling algorithm in a cloud computing environment," *Tsinghua Sci. Technol.*, vol. 21, no. 6, pp. 660–667, 2016.

[16] M. M. S. Maswood, M. D. R. Rahman, A. G. Alharbi, and D. Medhi, "A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment," *IEEE Access*, vol. 8, pp. 113737–113750, 2020.

[17] S. M. Mohammed, K. Jacksi, and S. Zeebaree, "A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 552–562, 2021.

[18] D. Alboaneen, H. Tianfield, Y. Zhang, and B. Pranggono, "A metaheuristic method for joint task scheduling and virtual machine placement in cloud data centers," *Futur. Gener. Comput. Syst.*, vol. 115, pp. 201–212, 2021.

[19] S. Zeebaree and I. Zebari, "Multilevel client/server peer-to-peer video broadcasting system," *Int. J. Sci. Eng. Res.*, vol. 5, no. 8, pp. 260–265, 2014.

[20] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, "A load balancing algorithm for the data centres to optimize cloud computing applications," *IEEE Access*, vol. 9, pp. 41731–41744, 2021.

[21] T. Hossfeld, P. E. Heegaard, and W. Kellerer, "Comparing the Scalability of Communication Networks and Systems," *IEEE Access*, 2023.

[22] S. Henning and W. Hasselbring, "A configurable method for benchmarking scalability of cloud-native applications," *Empir. Softw. Eng.*, vol. 27, no. 6, p. 143, 2022.

[23] N. R. Herbst, S. Kounev, and R. Reussner, "Elasticity in cloud computing: What it is, and what it is not," in *10th international conference on autonomic computing (ICAC 13)*, 2013, pp. 23–27.

[24] A. Weber, N. Herbst, H. Groenda, and S. Kounev, "Towards a resource elasticity benchmark for cloud environments," in *Proceedings of the 2nd International Workshop on Hot Topics in Cloud service Scalability*, 2014, pp. 1–8.

[25] P. Tran-Gia and T. Hoßfeld, *Performance Modeling and Analysis of Communication Networks: A Lecture Note*. BoD–Books on Demand, 2021.

[26] L. Abualigah and A. Diabat, "A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments," *Cluster Comput.*, vol. 24, pp. 205–223, 2021.

[27] N. Mansouri, R. Ghafari, and B. M. H. Zade, "Cloud computing simulators: A comprehensive review," *Simul. Model. Pract. Theory*, vol. 104, p. 102144, 2020.

[28] A. K. Sandhu, "Big data with cloud computing: Discussions and challenges," *Big Data Min. Anal.*, vol. 5, no. 1, pp. 32–40, 2021.

[29] H. Shukur, S. Zeebaree, R. Zebari, O. Ahmed, L. Haji, and D. Abdulqader, "Cache coherence protocols in distributed systems," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 3, pp. 92–97, 2020.

[30] Z. M. Khalid and S. R. M. Zeebaree, "Big data analysis for data visualization: A review," *Int. J. Sci. Bus.*, vol. 5, no. 2, pp. 64–75, 2021.

[31] Y. S. Jghef *et al.*, "Bio-Inspired Dynamic Trust and Congestion-Aware Zone-Based Secured Internet of Drone Things (SIoDT)," *Drones*, vol. 6, no. 11, p. 337, 2022.

[32] T. M. G. Sami, S. R. M. Zeebaree, and S. H. Ahmed, "A Comprehensive Review of Hashing Algorithm Optimization for IoT Devices," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 6s, pp. 205–231, 2023.

[33] S. R. Zeebaree, "DES encryption and decryption algorithm implementation based

on FPGA," *Indones. J. Electr. Eng. Comput. Sci*, vol. 18, no. 2, pp. 774–781, 2020.

[34] M. A. M. Sadeeq and S. R. M. Zeebaree, "Design and implementation of an energy management system based on distributed IoT," *Comput. Electr. Eng.*, vol. 109, p. 108775, 2023.

[35] B. S. Osanaiye *et al.*, "Network data analyser and support vector machine for network intrusion detection of attack type," *Rev. AUS*, vol. 26, no. 1, pp. 91–104, 2019.

[36] G. Brataas, N. Herbst, S. Ivansek, and J. Polutnik, "Scalability analysis of cloud software services," in *2017 IEEE International Conference on Autonomic Computing (ICAC)*, IEEE, 2017, pp. 285–292.

[37] S. Lehrig, H. Eikerling, and S. Becker, "Scalability, elasticity, and efficiency in cloud computing: A systematic literature review of definitions and metrics," in *Proceedings of the 11th international ACM SIGSOFT conference on quality of software architectures*, 2015, pp. 83–92.

[38] M. Woodside, "Scalability metrics and analysis of mobile agent systems," in *Workshop on Infrastructure for Scalable Multi-Agent Systems at the International Conference on Autonomous Agents*, Springer, 2000, pp. 234–245.

[39] S. Henning and W. Hasselbring, "How to measure scalability of distributed stream processing engines?," in *Companion of the ACM/SPEC international conference on performance engineering*, 2021, pp. 85–88.

[40] S. Henninga and W. Hasselbringa, "Theodolite: Scalability Benchmarking of Distributed Stream Processing Engines," *arXiv Prepr. arXiv2009.00304*, 2020.

[41] C. Xue, C. Lin, and J. Hu, "Scalability analysis of request scheduling in cloud computing," *Tsinghua Sci. Technol.*, vol. 24, no. 3, pp. 249–261, 2019.

[42] M. De Donno, K. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog," *Ieee Access*, vol. 7, pp. 150936–150948, 2019.

[43] D. Tao, Z. Lin, and B. Wang, "Load feedback-based resource scheduling and dynamic migration-based data locality for virtual hadoop clusters in openstack-based clouds," *Tsinghua Sci. Technol.*, vol. 22, no. 2, pp. 149–159, 2017.

[44] R. Tiwari, R. Sille, N. Salankar, and P. Singh, "Utilization and energy consumption optimization for cloud computing environment," in *Cyber Security and Digital Forensics: Proceedings of ICCSDF 2021*, Springer, 2022, pp. 609–619.

[45] S. R. M. Zeebaree, N. Cavus, and D. Zebari, "Digital Logic Circuits Reduction: A Binary Decision Diagram Based Approach," *L. LAMBERT Acad. Publ.*, 2016.

[46] S. B. N. F. A. Ehrenstein, "Scalability evaluation of explorviz with the universal scalability law." Kiel University, 2022.

[47] H. Nasiri, S. Nasehi, and M. Goudarzi, "Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities," *J. Big Data*, vol. 6, pp. 1–24, 2019.

[48] G. Hesse and M. Lorenz, "Conceptual survey on data stream processing systems," in *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, 2015, pp. 797–802.

[49] P. Y. Abdullah, S. R. Zeebaree, K. Jacksi, and R. R. Zeabri, "An hrm system for small and medium enterprises (sme) s based on cloud computing technology," *Int. J. Res.*, vol. 8, no. 8, pp. 56–64, 2020.

[50] Z. S. Ageed and S. R. M. Zeebaree, "Distributed Systems Meet Cloud Computing: A Review of Convergence and Integration," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 11s, pp. 469–490, 2024.

[51] S. Henning and W. Hasselbring, "Benchmarking scalability of stream processing frameworks deployed as event-driven microservices in the cloud," *arXiv Prepr. arXiv2303.11088*, 2023.

[52] H. Malallah *et al.*, "A comprehensive study of kernel (issues and concepts) in different operating systems," *Asian J. Res. Comput. Sci.*, vol. 8, no. 3, pp. 16–31, 2021.

[53] S. R. M. Zeebaree *et al.*, "Multicomputer multicore system influence on maximum multi-processes execution time," *TEST Eng. Manag.*, vol. 83, no. 03, pp. 14921–14931, 2020.

[54] S.-Y. Huang, C.-Y. Chen, J.-Y. Chen, and H.-C. Chao, "A Survey on Resource Management for Cloud Native Mobile Computing: Opportunities and Challenges," *Symmetry (Basel).*, vol. 15, no. 2, p. 538, 2023.

[55] I. Abdulkhaleq and S. Zeebaree, "State of Art for Distributed Databases: Faster Data Access, processing, Growth Facilitation and Improved Communications," vol. 5, pp. 126–136, Feb. 2021, doi: 10.5281/zenodo.4518872.

[56] H. S. Abdullah and S. R. M. Zeebaree, "Distributed Algorithms for Large-Scale Computing in Cloud Environments: A Review of Parallel and Distributed Processing," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 15s, pp. 356–365, 2024.

[57] S. R. Zeebaree and K. Jacksi, "Effects of processes forcing on CPU and total execution-time using multiprocessor shared memory system," *Int. J. Comput. Eng. Res. Trends*, vol. 2, no. 4, pp. 275–279, 2015.

[58] V. Lahoura *et al.*, "Cloud computing-based framework for breast cancer diagnosis using extreme learning machine," *Diagnostics*, vol. 11, no. 2. 2021. doi: 10.3390/diagnostics11020241.

[59] A. H. Ibrahem and S. R. M. Zeebaree, "Tackling the Challenges of Distributed Data Management in Cloud Computing-A Review of Approaches and Solutions," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 15s, pp. 340–355, 2024.

[60] D. A. Hasan, B. K. Hussan, S. R. M. Zeebaree, D. M. Ahmed, O. S. Kareem, and M. A. M. Sadeeq, "The impact of test case generation methods on the software performance: A review," *Int. J. Sci. Bus.*, vol. 5, no. 6, pp. 33–44, 2021.

[61] N. A. Kako, "DDLS: Distributed Deep Learning Systems: A Review," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 10, pp. 7395–7407, 2021.