**WITECH: Jurnal Teknologi Rekayasa Komputer dan Jaringan**
Vol. 2 (No.2), 2026: 81-89
https://journal.uwais.ac.id/index.php/witech/article/view/37

# Scalability of Distributed Systems Based on Decentralized Architecture

*Skalabilitas Sistem Terdistribusi Berbasis Arsitektur Terdesentralisasi*

**Haswindayani Hamzah[1], Rahmaniar[2]**
[1]Universitas Negeri Makassar, Indonesia; haswindayanihamzah4@gmail.com
[2]Universitas Negeri Makassar, Indonesia; rahmaniarzahraqueen@unm.ac.id

**Article History**
Received : 2026-09-10
Revised : 2026-09-25
Accepted : 2026-09-30
Published: 2026-10-01

**Keywords:**
*Scalability, Distributed System, Decentralized Architecture*

**Corresponding author:**
rahmaniarzahraqueen@unm.ac.id

**Paper type:**
Research paper

POLITEKNIK WAHDAH ISLAMIYAH MAKASSAR
Program Studi Teknologi Rekayasa Komputer dan Jaringan, Politeknik Wahdah Islamiyah Makassar, Indonesia

***Abstract***

*Scalability is a fundamental requirement in modern distributed systems due to the rapid growth of users, services, and data volumes. Distributed systems are expected to maintain performance, reliability, and availability while handling increasing workloads and operating in dynamic environments. This study aims to analyze scalability challenges in modern distributed systems and to identify architectural and operational approaches that support scalable system design. The research adopts a qualitative approach using a systematic literature review of recent scientific publications related to distributed systems. Data were collected from reputable international journals and conference proceedings that discuss system architecture, coordination mechanisms, communication models, and fault tolerance. The analysis results indicate that scalability is influenced by several interconnected factors, including architectural design, coordination overhead, communication efficiency, and failure handling mechanisms. Decentralized architectures and microservices-based designs are shown to improve scalability by distributing workload and reducing system bottlenecks. In addition, adaptive coordination strategies and asynchronous communication models contribute to improved system flexibility and performance under dynamic conditions. This study concludes that achieving scalability in modern distributed systems requires an integrated design approach that balances performance, reliability, and system complexity. The findings provide theoretical insight and practical guidance for researchers and practitioners in developing scalable and robust distributed systems.*

**Abstrak**

Skalabilitas merupakan kebutuhan utama dalam sistem terdistribusi modern seiring dengan pesatnya pertumbuhan jumlah pengguna, layanan, dan volume data. Sistem terdistribusi dituntut untuk tetap menjaga kinerja, keandalan, dan ketersediaan meskipun menghadapi peningkatan beban kerja serta lingkungan operasional yang dinamis. Penelitian ini bertujuan untuk menganalisis tantangan skalabilitas dalam sistem terdistribusi modern serta mengidentifikasi pendekatan yang mendukung perancangan sistem yang skalabel. Metode penelitian yang digunakan adalah pendekatan kualitatif melalui kajian pustaka sistematis terhadap publikasi ilmiah terkini yang membahas sistem terdistribusi. Data dikumpulkan dari jurnal internasional dan prosiding konferensi yang membahas arsitektur sistem, mekanisme koordinasi, model komunikasi, dan toleransi kesalahan. Hasil analisis menunjukkan bahwa skalabilitas dipengaruhi oleh beberapa faktor yang saling berkaitan, seperti desain arsitektur, biaya koordinasi, efisiensi komunikasi, dan mekanisme penanganan kegagalan. Arsitektur terdesentralisasi dan pendekatan berbasis

mikroservis terbukti mampu meningkatkan skalabilitas dengan mendistribusikan beban kerja dan mengurangi hambatan sistem. Selain itu, strategi koordinasi adaptif dan model komunikasi asinkron meningkatkan fleksibilitas dan kinerja sistem dalam kondisi yang dinamis. Penelitian ini menyimpulkan bahwa pencapaian skalabilitas memerlukan pendekatan desain terintegrasi yang menyeimbangkan kinerja, keandalan, dan kompleksitas sistem.

Copyright @ 2026 Authors.

**Cite this article:**
Haswindayani Hamzah, Rahmaniar. (2026). Scalability of Distributed Systems Based on Decentralized Architecture. *WITECH: Jurnal Teknologi Rekayasa Komputer dan Jaringan*, 2(2), 81-89. https://journal.uwais.ac.id/index.php/witech/article/view/37.

This work is licensed under a Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

## 1. Introduction

Distributed systems are now widely employed in many different industries, including cloud computing, big online platforms, distributed databases, and data-intensive applications, thanks to the rapid development of information technology. Microservices architecture is widely used in contemporary distributed systems to increase the flexibility, modularity, and scalability of cloud-based applications (Dragoni et al., 2017; Khazaei et al., 2022; Yang, 2025). Distributed systems improve performance, availability, and service reliability by distributing workloads among several nodes and enabling parallel processing. However, maintaining system scalability, dependability, and responsiveness to growing user demands is severely hampered by the increase in workload and system complexity.

By dividing programs into separate services, microservices design increases the scalability of distributed systems (Qin, 2025). Large programs can be broken down into smaller, independent services thanks to microservices design, which facilitates the development, implementation, and scalability of contemporary distributed systems (Shabani et al., 2021; Pothuri & Koppula, 2024). Because microservices design offers more flexibility and fault tolerance than traditional monolithic architecture by allowing functions to be broken into independent modules that can be developed, tested, and scaled separately, many organizations have embraced it (Pothuri & Koppula, 2024). In the context of distributed systems, scalability is the system's capacity to accommodate growing user counts, data volumes, and workloads by adding or modifying resources without compromising performance. Large programs can be broken down into smaller, independent services thanks to microservices design, which facilitates the development, implementation, and scalability of contemporary distributed systems (Shabani et al., 2021; Pothuri & Koppula, 2024).

Furthermore, further research shows that distributed and adaptive processing techniques might enhance scalability and performance, particularly when handling complicated data streams (Siachamis et al., 2023).

There are still gaps in understanding how to combine architectural design, coordination mechanisms, communication models, and fault tolerance strategies to achieve optimal scalability, despite the fact that numerous studies have looked at

different approaches to improve scalability, such as the use of microservices and adaptive streaming processing. Decomposing functions and managing containers are crucial for achieving scalability and efficiently managing workloads, according to research on the shift from monolithic architecture to microservices architecture. Additionally, systematic reviews of earlier research show that approaches that connect operational and architectural elements within a thorough scaling framework are still lacking (Gurung et al., 2025). These drawbacks show that many research studies concentrate on particular topics, such as load balancing, consensus techniques, or communication protocols, and fail to incorporate all connected components into a comprehensive understanding. In the meantime, significant theoretical viewpoints on the difficulties of developing and assessing the performance of large distributed systems are offered by major scientific conferences devoted to distributed computing. However, there is still a dearth of comprehensive studies on the useful contributions and applications of integrative approaches in a variety of dynamic situations.

This paper evaluates recent academic contributions in the field of distributed systems and scalability using a systematic literature review method in order to overcome these constraints. This study's primary goals are to determine the critical elements influencing scalability, examine current methods, and emphasize design concepts that can be used to create distributed systems that are dependable and scalable (Artiom, 2025). In order to help system designers and researchers create effective and scalable distributed architectures, this study is anticipated to offer both theoretical contributions and useful suggestions.

## 2. Literature Review

### 2.1 Scalability in Distributed Systems: An Overview and Definition

In distributed systems, scalability typically refers to the system's capacity to sustain service availability and performance as the workload grows by adding more resources.(for example, adding service instances or nodes). Scalability in the context of contemporary service systems is frequently linked to the capacity to expand horizontally while preserving fault tolerance in the event of increased traffic or component breakdowns (Pothuri & Koppula, 2024). According toearlier studies, service granularity and the system's capacity for dynamic resource management both have an impact on scalability in microservices systems. Scalability should also be seen as the capacity to flexibly adjust to shifts in workload and data properties. This is demonstrated in streaming data processing, where data velocity and distribution may fluctuate over time. (Richa, 2017).

### 2.2 Methods and Architectural Models for Increasing Scalability

Microservices architecture, which divides an application into several small, independent services, is one of the most often discussed strategies for increasing scalability. By breaking down applications into independent small services, microservices architecture can increase the flexibility and scalability of distributed systems, but it also presents new challenges, like distributed transaction management and data consistency. (Bashtovyi & Fechan, 2024) Additionally, the adoption of container management and service orchestration techniques that support large-scale distributed systems is made possible by the use of microservices architecture. (Rajesh Kesavalalji, 2024).

Vol. 2 (No.2), 2026: 81-89
https://journal.uwais.ac.id/index.php/witech/article/view/37

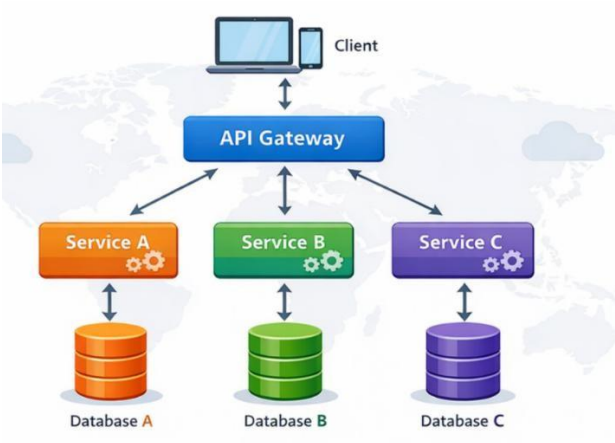


Figure 1. Distributed System Architecture Based on Microservices

Because each part may be scaled separately, unlike with a monolithic structure, the burden does not "pull" the entire system (Pothuri & Koppula, 2024). Another study describes the transition from monolithic architecture to microservices as a decomposition process that necessitates the definition of domain boundaries.

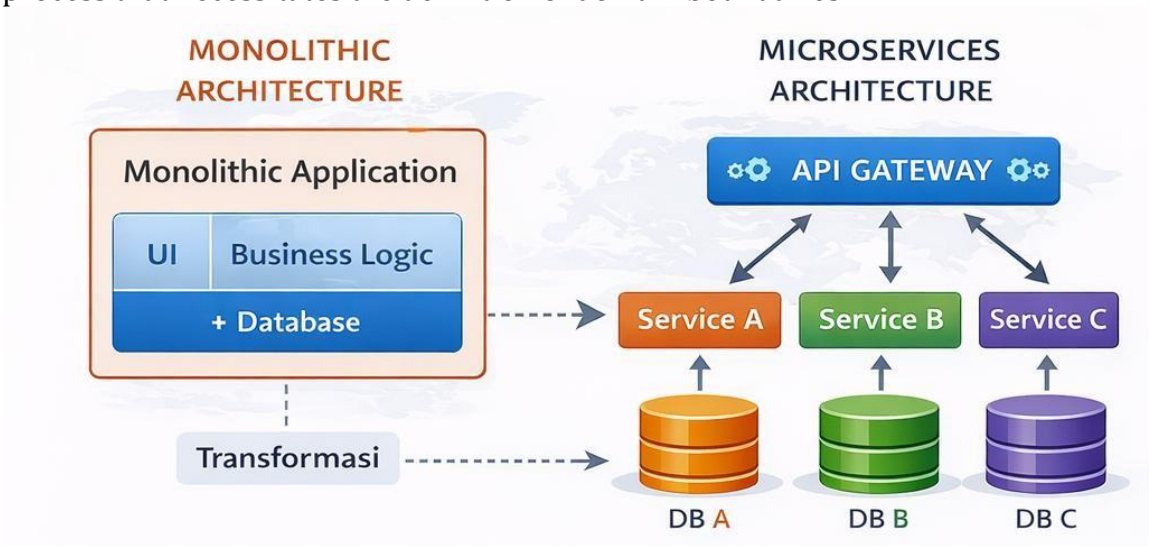


Figure 2. Monolithic and Microservices Architecture Comparison

Deployment, monitoring, and scaling are made easier by service interface design and technical assistance like orchestration and containerization. As a result, architectural choices and operational procedures (such CI/CD and observability) that facilitate system modifications and expansion are just as important to scalability as technology.

2.3 Mechanisms for Coordination, Communication, and Autoscaling

Studies reveal that, in addition to architectural design, coordination costs, service communication techniques, and autoscaling strategies frequently result in scalability constraints. A comprehensive study of the literature on microservices scalability reveals that many studies divide talks into three categories: implementation issues in complex real-world situations, evaluation measures, and adaptation solutions

(such autoscaling).(Gurung et al., 2025). Improving the scalability and dependability of distributed systems based on microservices requires effective resource management techniques (Vali et al., 2025). Additionally, a number of studies show that message-driven design and asynchronous communication can improve system elasticity and lessen inter-service dependencies (Henning & Hasselbring, 2024). As shown in Figure 3, communication models between services in distributed systems can take either synchronous or asynchronous approaches.




Figure 3. Synchronous versus Asynchronous Communication in Distributed Systems: A Comparison

On the other hand, streaming research emphasizes how crucial it is to use data partitioning and adaptive load balancing strategies. To maintain low latency and steady throughput, these methods must be able to modify load distribution when data distribution shifts (Siachamis et al., 2023). Additionally, while implementing microservices in contemporary distributed systems, inter-service communication security is a crucial consideration (Hannousse & Yahiouche, 2020). This highlights the need for a combination of runtime adaptive mechanisms and architectural design for scalability in contemporary distributed systems.

### 2.3 Previous Research Results and Research Gaps

Significant contributions to scalability capabilities and microservices approaches have generally been found in prior studies. To lessen communication problems and boost service dependability, they contain auxiliary elements like service registry, API gateway, load balancer, and service mesh.(Pothuri & Koppula, 2024). According to other studies, breaking down monolithic systems is crucial since it speeds up feature development and increases scalability, but it also makes maintaining inter-service relationships and synchronizing distributed data more difficult. Results reported in the DISC proceedings show that problems like consensus, fault tolerance, and large-scale system dynamics continue to be major concerns in the context of distributed algorithms and computational theory. Consequently, the basic constraints of distributed system coordination must be taken into account in scalability-oriented system design.

## 3. Method of Research

### 3.1 Design of Research

This study uses a qualitative methodology and a systematic literature review strategy. Based on results from earlier study, this method was selected to allow the researcher to obtain a thorough grasp of numerous ideas, difficulties, and strategies for attaining scalability in contemporary distributed systems. Researchers can spot trends, important discoveries, and research gaps from a variety of pertinent scientific sources in an organized and repeatable way by employing the systematic literature review method.

3.2 Method of Data Collection

The scientific literature from respectable conference proceedings and international journals that address distributed systems and scalability was reviewed in order to gather research data. Online academic repositories that offer peer-reviewed articles served as the data sources. The search process was conducted using keywords such as distributed systems, scalability, microservices architecture, coordination mechanisms, and fault tolerance. This analysis only considered English-language papers that were pertinent to the setting of contemporary distributed systems.

A total of 85 publications from different scientific databases were found during the literature search procedure. Seventy articles remained after duplicates were eliminated, and they were subsequently vetted using abstracts and titles. Forty publications were found to be pertinent to the research topic after this screening process. After a full-text review, 28 papers that satisfied the eligibility requirements were found. Fifteen of these papers were chosen as the main sources for this study's analysis.

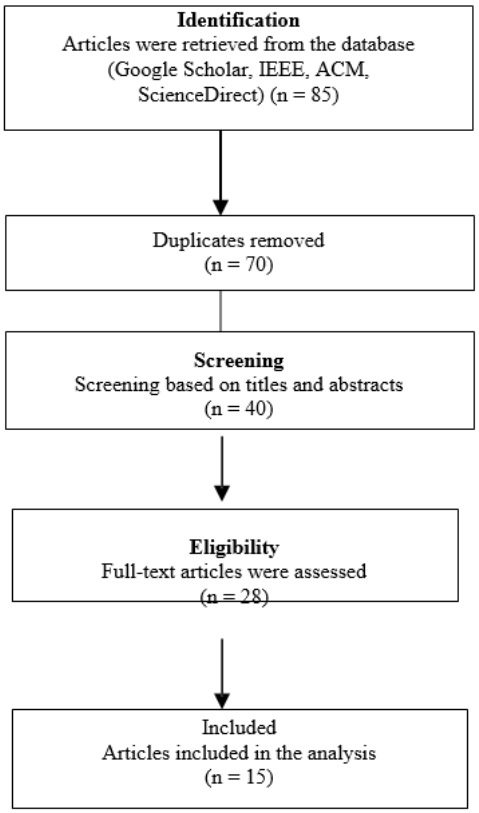


Figure 4. Literature Selection Process Using the PRISMA Method

3.3 Research Sample and Population

All scientific papers that address scalability in distributed systems make up the population of this study. Purposive sampling, which entails choosing papers that specifically address facets of architectural design, coordination mechanisms, communication models, or fault tolerance measures linked to scalability, was used to choose the study sample. Articles with unclear methodological justifications or those unrelated to the study's goals were not included in the sample.

3.4 Method of Data Analysis

Thematic analysis was used to examine the data. To determine the key topics, practical strategies, and conclusions pertaining to scalability, each chosen article was assessed. These themes were then divided into a number of areas, including ways for enhancing scalability, coordination and communication, fault tolerance mechanisms, and system architecture. This approach makes it possible to compare different studies and make methodical conclusions.

3.5 A tool for research

A literature review matrix, which was created to capture crucial details from every article, served as the study's instrument. This included the goals of the study, the methods that were used, the scalability issues that were examined, and the key conclusions. This matrix could be used to improve the research's transparency and reproducibility while maintaining consistency in the data collection procedure.

## 4. Findings and Conversation

### 4.1 Results

Several important conclusions on scalability in modern distributed systems were found after a thorough review of the literature. According to the analysis, architectural design, coordination methods, communication patterns, and fault tolerance strategies all interact to affect scalability. The first result emphasizes how crucial system architecture is to improving scalability.Decentralized distributed systems can lessen reliance on central components and distribute workloads more fairly. The system can more successfully adjust to growing workloads and the inclusion of additional services thanks to this tactic. The second discovery has to do with the coordination and communication between systems. Under varying workloads, systems that make use of adaptive coordination methods and asynchronous communication typically exhibit more consistent performance. This method makes it possible for processes to operate more effectively by reducing bottlenecks in data transmission between components. Additionally, it has been demonstrated that fault tolerance capabilities is an essential component of scalability. Compared to systems that depend on centralized coordination, those that are built to continue operating even in the event that some nodes fail show higher availability and are more suited for scaling. Figure 4 summarizes the major elements affecting distributed systems' scalability.




Figure 5. Key Factors Influencing the Scalability of Distributed Systems

A summary of the key factors influencing the scalability of distributed systems is presented in Table 1.

Table 1. Factors Influencing the Scalability of Distributed Systems

| **Key Factors** | **Description** | **Impact on Scalability** |
|---|---|---|
| System Architecture | Centralized vs. Decentralized | High |
| Coordination Mechanisms | Static Coordination vs. Adaptive Coordination | High |
| Communication Model | Synchronous vs. Asynchronous | Medium–High |
| Fault Tolerance | Failure Recovery Mechanisms | Medium |
| Load Management | Load Distribution and Load Balancing | Medium |

### 4.2 Discussion

This study indicates that achieving scalability in modern distributed systems should be viewed as an inherent property of the system as a whole. Scalability is not merely related to increasing resource capacity, but also involves the system's ability to handle complexity, coordination, and the dynamics of its operational environment. Decentralized architecture provides significant advantages in reducing performance bottlenecks and improving system flexibility. However, this approach also introduces new challenges related to managing dependencies and coordination among components. Therefore, adaptive coordination mechanisms are required to ensure system stability as it scales. The discussion highlights the important role of asynchronous communication in supporting scalability. By reducing direct dependencies between components, the system becomes more resilient to workload fluctuations and network disruptions. In addition, the implementation of fault tolerance enables the system to maintain service availability even in the presence of partial failures. Overall, these findings and analyses emphasize that designing scalable distributed systems requires an integrated approach that simultaneously considers architecture, coordination, communication, and fault tolerance. Such an approach is key to building modern distributed systems that can grow sustainably

## 5. Conclusion

This study examines scalability as a crucial aspect in the development of modern distributed systems that face increasing workloads, growing numbers of users, and the complexity of operational environments. Through a systematic literature review method, this research analyzes various approaches and scientific findings related to factors influencing the ability of distributed systems to scale efficiently and reliably. The results indicate that scalability cannot be understood as a single technical capability, but rather as a system characteristic formed through the interaction of architectural design, coordination mechanisms, communication models, and fault tolerance strategies. A decentralized architectural approach has been shown to make a significant contribution in distributing workloads and reducing dependence on central components. Furthermore, the use of asynchronous communication and adaptive coordination mechanisms helps maintain system performance stability under dynamic workload

conditions. Fault tolerance capability also plays a critical role in ensuring service availability in the event of partial system failures.

## Referensi

Artiom, K. (2025). Microservices architecture: Accelerating feature development and scalability through monolith decomposition. *14*(01). https://doi.org/10.18535/ijecs/v14i01.4958

Bashtovyi, A., & Fechan, A. (2024). Distributed transactions in microservice architecture: Informed decision-making strategies. *15*, 449–459.

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: Yesterday, today, and tomorrow. In *Present and ulterior software engineering* (pp. 195–216). https://doi.org/10.1007/978-3-319-67425-4_12

Gurung, N., Shrestha, S., & Chulyadyo, R. (2025). Scalability in microservices: A systematic literature review. *Journal of Computer Science and Technology (Argentina), 25*(2), 128–143. https://doi.org/10.24215/16666038.25.e11

Hannousse, A., & Yahiouche, S. (2020). Securing microservices and microservice architectures: A systematic mapping study.

Henning, S., & Hasselbring, W. (2024). Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud. *Journal of Systems and Software, 208*. https://doi.org/10.1016/j.jss.2023.111879

Khazaei, H., Mahmoudi, N., Barna, C., & Litoiu, M. (2022). Performance modeling of microservice platforms. *IEEE Transactions on Cloud Computing, 10*(4), 2848–2862. https://doi.org/10.1109/TCC.2020.3029092

Pothuri, V., & Koppula, R. (2024). Scalable microservices architectures: Building resilient distributed systems for the modern web. *International Journal of Advancements in Computational Technology, 2*(3), 101–109. https://doi.org/10.56472/25838628/IJACT-V2I3P109

Qin, R. (2025). *Leveraging microservices architecture for large-scale distributed systems* (pp. 1–6).

Rajesh Kesavalalji. (2024). Scalable and fault-tolerant microservices architecture: Leveraging AI-driven orchestration in distributed cloud systems. *International Journal of Science and Research Archive, 13*(1), 3501–3511. https://doi.org/10.30574/ijsra.2024.13.1.1566

Richa, A. W. (2017). *31 International Symposium on Distributed Computing* (Vol. 91).

Shabani, I., Mëziu, E., Berisha, B., & Biba, T. (2021). Design of modern distributed systems based on microservices architecture. *International Journal of Advanced Computer Science and Applications, 12*(2), 153–159. https://doi.org/10.14569/IJACSA.2021.0120220

Siachamis, G., Psarakis, K., Fragkoulis, M., Papapetrou, O., Van Deursen, A., & Katsifodimos, A. (2023). Adaptive distributed streaming similarity joins. In *DEBS 2023: Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems* (pp. 25–36). https://doi.org/10.1145/3583678.3596891

Vali, A. A., Azizi, S., Shojafar, M., & Buyya, R. (2025). And edge computing: State-of-the-art and future directions. *ACM Computing Surveys, 1*(01), 1–46.

Yang, C. (2025). Theoretical analysis of distributed systems and their scalability. *Advances in Computer, Signals and Systems, 9*(1), 21–24. https://doi.org/10.23977/acss.2025.090104