ISSN 2616-5775 Vol. 8, Issue 1: 80-86, DOI: 10.25236/AJCIS.2025.080111

Design and Implementation of Elastic Architecture for Big Data Information System Based on Cloud Computing

Hang Yang[1,*], Wanqi Zhang[2]

[1]College of Software and Convergence Computer Engineering, Inha University, Incheon 22212, Korea
[2]Fashion Marketing, Sungkyunkwan University, Seoul 03063, Korea
[*]Corresponding author

Abstract: *With the continuous development of information technology and the rapid growth of data scale, traditional data processing and storage architectures have gradually exposed many bottlenecks, making it difficult to meet the high performance, high availability, and high scalability requirements of big data applications. This article proposes a resilient architecture design and implementation scheme for big data information systems based on cloud computing. This architecture utilizes the resource elasticity and scalability of cloud computing platforms to achieve dynamic adjustment and load balancing of big data processing systems. It can automatically allocate and release computing resources based on real-time load requirements, ensuring the stability and efficiency of the system when facing massive data and high concurrency access. Specifically, this article first analyzes the characteristics of big data information systems and their requirements for elastic architecture. Then, a cloud computing based elastic architecture model is designed, and containerization technology and microservice architecture are used to achieve modularity and scalability of the system. In addition, based on specific cases, this article has implemented a big data processing system based on a cloud computing platform, verifying the effectiveness and feasibility of the proposed architecture. The experimental results show that this architecture can significantly improve the performance, maintainability, and resource utilization of the system, with strong adaptability and scalability, providing a new solution for big data applications.*

Keywords: *cloud computing, big data, elastic architecture*

1. Introduction

With the rapid development of information technology, cloud computing and big data have become important forces driving modern technological progress. Cloud computing, as a new type of computing model, has gradually become the core support for enterprise information transformation due to its characteristics of resource sharing, on-demand allocation, and elastic expansion. At the same time, big data technology enables decision-making in various industries to become more data-driven and intelligent through the processing, analysis, and mining of massive amounts of data. In this context, the combination of cloud computing and big data technology is not only a trend in the field of information technology, but also an important engine for promoting the development of the digital economy.

However, in the increasingly widespread use of cloud computing platforms and big data applications, how to build an efficient, stable, and reliable system architecture has become a key issue facing technicians. Especially when facing complex scenarios such as sudden high loads, equipment failures, and resource waste, how to design a highly resilient system that can maintain excellent performance and availability in various environments has become an urgent technical challenge to be solved. The proposal of elastic architecture is precisely to address these challenges. By flexibly scheduling resources and dynamically scaling computing capabilities, elastic architecture can automatically adjust in the event of system load fluctuations or failures, ensuring high availability and stability of the system.

This paper focuses on the research topic of "Design and Implementation of Elastic Architecture for Big Data Information Systems Based on Cloud Computing", aiming to explore how to provide an efficient, scalable, and self-healing big data processing platform in a cloud computing environment through reasonable architecture design and technical means. We will conduct research around the following aspects: firstly, analyze the basic requirements of cloud computing and big data systems, and clarify the necessity of elastic architecture; Secondly, design a cloud based elastic architecture that covers core mechanisms such as resource scheduling, automatic scaling, load balancing, and fault tolerance; Then, implement the architecture and verify its effectiveness and feasibility through a series of performance tests and evaluations; Finally, based on the experimental results, further optimize the architecture design and look forward to future development directions.

This study has significant theoretical and practical implications. From a theoretical perspective, the research on elastic architecture not only provides new technological paths for big data processing systems, but also offers innovative ideas for resource optimization and management in cloud computing environments. From a practical perspective, with the popularization of cloud computing and big data, the demand for elastic computing platforms among major enterprises is increasing. The results of this study can provide more efficient, stable, and intelligent architecture solutions for cloud platforms, helping enterprises better cope with the problems of explosive data growth and business demand fluctuations, and improve overall operational efficiency.

2. Fundamentals of cloud computing and big data technology

2.1 Overview of cloud computing

Cloud computing refers to a computing mode that provides shared computing resources, storage space, applications and services through the Internet [1]. It provides users with on-demand self-service through a centralized server cluster, where users do not have to worry about hardware facilities, software deployment, and resource management issues. They only need to access cloud resources according to their needs and pay based on usage.

The core features of cloud computing include on-demand self-service, extensive network access, resource pooling, elastic scaling, and metering services [2]. Users can automatically configure and manage resources through a simple interface, obtain services on demand without interacting with service providers, thereby improving flexibility and saving manual intervention and configuration time. Cloud computing services can be accessed from anywhere through the network, whether it's a PC, smartphone, or other terminal device, and users are not limited by physical devices or geographical location. Cloud service providers centralize computing resources, storage, and network bandwidth in data centers and use virtualization technology for pooling management, allowing multiple users to share the same physical resources and dynamically adjust allocation to achieve efficient utilization. Cloud platforms can automatically expand or contract resources according to changes in demand, helping enterprises cope with load fluctuations and save costs. Cloud computing also adopts a usage based billing method, where users pay based on their actual computing power, storage capacity, or bandwidth consumption, without the need for pre investment or management of hardware resources[3].

The service models of cloud computing are usually divided into three types: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). IaaS provides virtualized computing resources, such as virtual machines, storage, and network infrastructure, which users can manage and use independently. PaaS provides a platform for developing and running applications, including operating systems, databases, middleware, etc. Developers can focus on application development without managing underlying hardware and operating systems. SaaS provides software applications through the network, and users do not need to install and maintain the software. They only need to access and use it through a browser. Typical examples include Google Workspace and Salesforce.

2.2 Overview of big data

Big Data refers to large, complex, and rapidly generated datasets that cannot be processed by traditional data processing tools. These data not only include structured data, but also semi-structured and unstructured data, covering various forms such as text, images, audio, video, sensor data, etc. The key characteristics of big data are the "5V" model: Volume, Velocity, Variety, Veracity, and Value.

The volume of big data is very large. With the development of the Internet, social media, e-commerce and the Internet of Things continue to generate a large amount of data. The scale of these data has exceeded the processing capacity of traditional databases. The generation speed of big data is also very fast, such as real-time updates of financial transactions, sensor data, and social media information. How to process these data in near real time has become a major challenge. Big data not only includes structured tabular data, but also involves unstructured data from different sources such as text, audio, video, and images, which puts higher demands on data processing technology. The sources of big data are extensive and complex, with varying levels of quality, and may contain noise, errors, or incomplete data. Extracting true and accurate information from them is a challenge in analysis. The true value of big data lies in how to extract useful information through analysis, support decision-making, discover trends, create business opportunities, and more. If value cannot be effectively extracted, massive data can only be useless' noise '.

Big data technology encompasses multiple key areas, including data storage and management technology, data processing and analysis technology, data mining and machine learning, as well as data visualization technology. Data storage and management technologies, such as distributed file systems (such as HDFS) and NoSQL databases, can efficiently handle the storage and access of large-scale data; Data processing and analysis techniques, such as MapReduce and Spark distributed computing frameworks, enable parallel processing of large-scale data; Data mining and machine learning provide support for intelligent prediction and decision-making by analyzing patterns and patterns in data; Data visualization technology presents analysis results through charts and dashboards, helping users understand the meaning and trends behind the data.

2.3 Integration of cloud computing and big data

The integration of cloud computing and big data is one of the key technologies driving the development of the digital economy and smart society. The combination of the two provides more efficient and flexible data storage and processing capabilities for enterprises and individuals, making data analysis more real-time and intelligent. Cloud computing provides a powerful infrastructure for big data, meeting the storage and computing needs of big data. Through an elastic and scalable cloud platform, it supports efficient processing of big data. With the widespread development of big data applications, the functionality and architecture of cloud computing platforms are constantly being optimized to provide more computing resources, distributed computing frameworks, and data analysis tools to meet the storage, analysis, and mining needs of big data. Meanwhile, the elastic scalability of cloud computing provides strong support for big data analysis, allowing enterprises to dynamically adjust resources as needed to ensure the efficiency and reliability of data processing. In addition, cloud platforms ensure data security through multi-level security mechanisms, while data governance and quality management in big data analysis ensure the accuracy and credibility of analysis results. Through the integration of cloud computing and big data, enterprises can achieve data-driven decision-making, improve operational efficiency and innovation capabilities, and promote digital transformation in various industries. In short, the combination of cloud computing and big data has opened up new development space for modern technology applications, promoted intelligent and data-driven future development, and provided new development opportunities for various industries.

3. Elastic requirements and architecture design of big data information systems

With the widespread application of big data technology, enterprises and organizations have an increasing demand for the flexibility of information systems, especially when dealing with massive amounts of data. Systems need to have the ability to respond quickly, expand flexibly, and recover reliably. In order to meet these requirements, the architecture design of big data information systems must have resilience to ensure stable operation under high load conditions and rapid recovery in the event of failures. This article will discuss in detail the analysis of elastic architecture requirements, elastic architecture design solutions, and fault-tolerant and recovery mechanism design.

3.1 Analysis of elastic architecture requirements

The elastic architecture design of big data information systems requires comprehensive analysis from multiple aspects. Firstly, the system must have scalability, which allows for dynamic expansion of computing resources and storage capacity as data volume grows, supporting both vertical and horizontal scaling. Secondly, high availability is crucial, as the system should be able to automatically switch to a backup system or node in the event of hardware failure or network issues, ensuring data continuity and accessibility. In order to meet real-time requirements, the architecture also needs to have low latency data processing and stream processing capabilities to ensure timely feedback of analysis results. The ability to recover from failures is equally important. The system should have self-healing

ISSN 2616-5775 Vol. 8, Issue 1: 80-86, DOI: 10.25236/AJCIS.2025.080111

capabilities and design reasonable backup mechanisms and recovery strategies to quickly restore normal services. In addition, data consistency also needs to be guaranteed, and data synchronization issues between multiple nodes must be considered in the design to ensure that data is not lost or erroneous in the event of node failures or updates. Based on the above requirements, the design of big data information systems must take into account flexibility, stability, scalability, and efficiency.

3.2 Elastic architecture design scheme

On the basis of meeting the elastic requirements of big data, the elastic architecture design scheme should focus on multiple key aspects. Firstly, distributed architecture is the foundation for achieving elastic scalability. By dispersing data and computing tasks across multiple nodes, the system can achieve horizontal scaling and increase computing and storage resources, such as using distributed frameworks like Hadoop or Spark. Secondly, using the microservice architecture can achieve modular design. Each microservice can be expanded and deployed independently to avoid the failure of a single module affecting the overall performance. At the same time, flexible resource scheduling and load balancing are supported through API communication. In addition, containerization technologies such as Docker and container orchestration tools such as Kubernetes can achieve rapid deployment, elastic scaling, automated scaling, load balancing, and fault recovery. Automated resource scheduling achieves on-demand scalability by dynamically allocating computing resources based on load conditions, such as elastic computing services provided by cloud platforms. In terms of data storage, a distributed storage system is adopted to achieve load balancing and high availability through data sharding and replication technology, ensuring that data will not be lost even in the event of a node failure. Finally, the load balancing mechanism dynamically allocates request traffic to ensure even system load, thereby enhancing system resilience and performance, as shown in Figure 1.

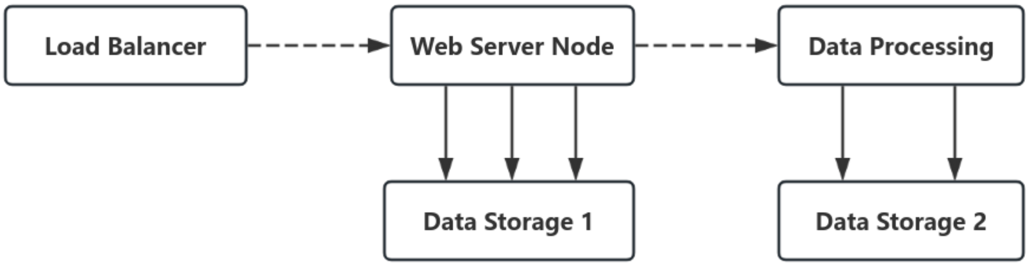

Figure 1. Schematic diagram of elastic architecture design

3.3 Fault tolerance and recovery mechanism design

Fault tolerance and recovery mechanisms are the core of ensuring the stability of big data information systems. In elastic architecture design, fault tolerance capability and fault recovery strategies must be provided to cope with sudden failures and network interruptions. To this end, redundancy design ensures that when a node fails, the system can automatically switch to a backup node through multi node redundancy. Multiple replicas are used to store data and consistency protocols (such as Paxos or Raft) are used to ensure data consistency and avoid data loss. Heartbeat detection and automatic fault switching mechanism regularly check the health status of nodes, and faulty nodes will immediately trigger task transfer to healthy nodes. The data recovery strategy ensures data redundancy by regularly backing up to remote storage and using distributed file systems such as HDFS, effectively avoiding single points of failure. The task retry mechanism automatically retries failed tasks during data processing and reduces the impact of repeated failures through retry limit and rollback strategies. In addition, a detailed fault logging and analysis mechanism records the cause, handling process, and recovery results of each fault, providing a basis for subsequent fault analysis and prevention, as shown in Figure 2.

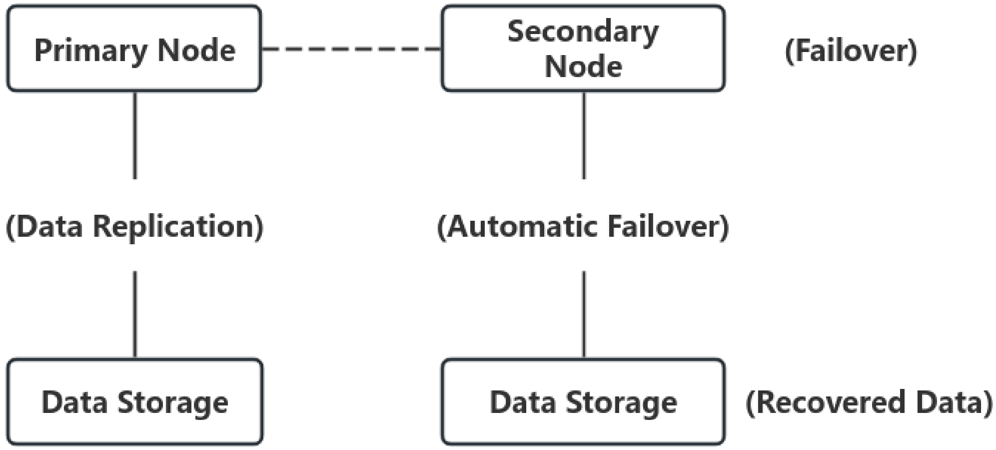

Figure 2. Schematic diagram of fault tolerance and recovery mechanism

Through the above design scheme, the big data information system can maintain high availability and quickly restore operation in the face of sudden failures, ensuring the security and consistency of data.

In actual production environments, the design of resilient architecture and fault-tolerant mechanisms needs to be combined with specific business requirements and technology stacks, taking into account various factors such as performance, cost, and complexity. With the help of modern cloud computing platforms and distributed technology, efficient and reliable elastic architectures can be effectively built, providing a solid foundation for big data processing.

4. System implementation and testing

The implementation and testing of the system are key steps in ensuring the smooth delivery and stable operation of software projects. It not only involves reasonable technology selection and system architecture design, but also includes a comprehensive testing process and later optimization and improvement.

4.1 Technical selection and implementation

In the process of system development, technology selection is the foundation for ensuring efficient, stable, and scalable operation of the project. To ensure the long-term maintainability and high performance of the project, this study chose the following technology stack: the front-end part adopts the React framework, whose component-based design and virtual DOM technology improve page rendering efficiency, especially when dealing with complex interfaces and high-frequency interactions, it can maintain good performance. At the same time, Redux is used for global state management to ensure the unity of states between components and the predictability of data flow. The backend part adopts the Java Spring Boot framework, which improves development efficiency with its simple configuration and fast development capabilities. Its powerful ecosystem and support for microservice architecture ensure the scalability and maintainability of the system. In terms of data storage, MySQL and MongoDB are combined. MySQL is used to store structured data, providing powerful transaction processing and data consistency assurance, while MongoDB is used to process unstructured or semi-structured data, providing efficient document storage and flexible query capabilities. To ensure the scalability and high availability of the system, Docker containerization technology is adopted to deploy various services as containers and perform container orchestration through Kubernetes, achieving automated deployment, scaling, and monitoring. The entire development process follows the agile development philosophy, using Jira for task management and GitLab CI/CD for continuous integration and delivery, ensuring high code quality and rapid iteration, as shown in Figure 3.

ISSN 2616-5775 Vol. 8, Issue 1: 80-86, DOI: 10.25236/AJCIS.2025.080111

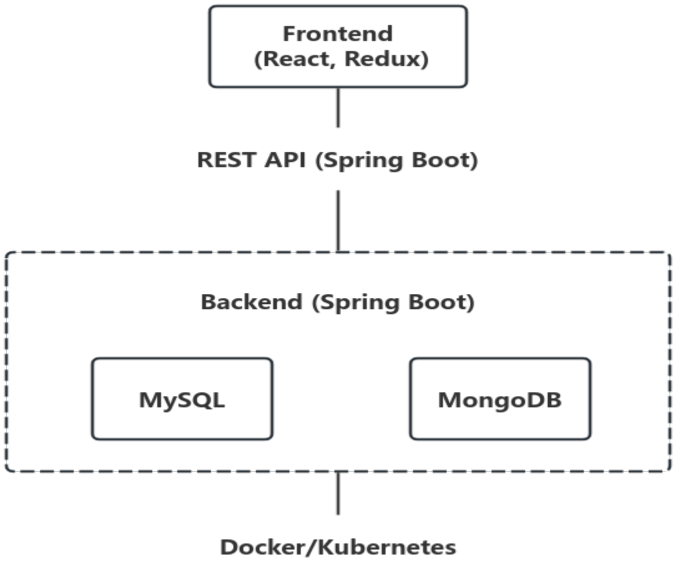

Figure 3. Technical architecture diagram of the system

4.2 System testing and evaluation

After the completion of system development, system testing is an important step in verifying whether the system can work as expected. We conducted comprehensive testing on the system, including functional testing, performance testing, security testing, compatibility testing, and recovery testing. Firstly, the purpose of functional testing is to ensure that all system functions operate normally according to the design document requirements. We have covered various functions of the system through a combination of manual and automated testing, such as user registration and login processes, data storage and retrieval functions, permission management and role control, and API interface testing provided. Through functional testing, we have confirmed that there are no major issues with the system's various functions, and all core functions can operate normally. Secondly, the purpose of performance testing is to verify the system's responsiveness under high concurrency and high load conditions. We conducted stress and load tests on the system using JMeter and Gatling tools. The test results showed that with a concurrent user count of 1000, the average response time of the system was 250ms, and it maintained low latency in high concurrency scenarios. In addition, the system can handle 500 requests per second and has good scalability, as shown in Table 1.

Table 1. Performance test data

System throughput and response time pressure test results		
Number of developed users	Throughput (requests/second)	Response time (ms)
100	200	180
500	450	230
1000	500	250
2000	480	300

In addition, in system testing, to ensure the security of the system, we conducted security tests such as SQL injection, XSS attacks, and identity authentication and permission control. The test results showed that the system can effectively prevent common security threats, and data encryption and permission control have been effectively implemented. Compatibility testing ensures that the system can run smoothly on different devices, operating systems, and browsers. We conducted tests on mainstream browsers such as Chrome, Firefox, Safari, and Edge to ensure consistent interface display and interactive functionality, while also verifying compatibility with Windows, macOS, Linux, and mobile devices (Android/iOS). Recovery testing verifies the system's fault tolerance and data recovery capabilities by simulating system failures such as database crashes, server crashes, etc. The system can automatically resume normal operation within seconds with a data loss rate of 0%, proving that the system has a good fault recovery mechanism.

4.3 Optimization and improvement

After the system went online, we continuously optimized and improved it to ensure its efficient and stable operation under high load conditions. Based on the test results and user feedback, optimization was first carried out to address the performance bottleneck issue, mainly including database query optimization. By optimizing the MySQL database index, the query speed was improved by 30%, and Redis caching mechanism was introduced to reduce database access pressure and significantly improve response speed, especially in high concurrency scenarios. Secondly, we introduce Nginx as a load balancer to evenly distribute requests to multiple server nodes, enhancing the system's concurrent processing capability; And based on Kubernetes' automatic scaling mechanism, it ensures that the system can automatically scale resources and maintain sufficient processing power when the load increases. In order to better monitor the system's operational status, we use Prometheus and Grafana to collect key indicators in real-time. If any abnormalities are found, the system will send alert notifications through Alertmanager to remind administrators to handle them in a timely manner. Finally, we will continue to monitor technological updates and evolution, regularly upgrade our systems to ensure they remain at the forefront of technology, and plan to introduce machine learning models for intelligent optimization in the future to automatically predict system loads and adjust resources.

Through the above optimization and improvement, we ensure that the performance, stability, and security of the system are always in the best state, and are prepared for future demand growth.

5. Conclusion and future work prospects

This article conducts in-depth research on the design and implementation of elastic architecture for cloud computing based big data information systems, and explores how to build an efficient, stable, and scalable system architecture in the constantly changing business requirements and technological environment. Firstly, this article analyzes the characteristics of big data processing and the advantages of cloud computing, and proposes an architecture design scheme that combines cloud computing resource pools for dynamic expansion to ensure that the system can cope with large-scale data traffic and complex computing requirements. Secondly, the article focuses on introducing the core technologies of elastic architecture, including load balancing, automatic scaling, fault tolerance mechanisms, etc., and elaborates on specific technical implementation details. These technologies can ensure the stability and efficiency of the system under high load and high concurrency conditions, and automatically adjust resources when the load fluctuates to achieve the optimal configuration of system resources.

The effectiveness of the proposed architecture design scheme in practical applications has been verified through experiments and performance testing. The test results show that cloud computing based big data information systems can maintain good response speed and system stability under changing loads and user demands, meeting the needs of modern big data processing and analysis.

In the future, with the continuous development of cloud computing technology and big data applications, the design of elastic architecture will continue to evolve towards a more intelligent and automated direction. Introducing more artificial intelligence and machine learning technologies, combined with real-time data stream analysis and prediction, can further enhance the intelligent scheduling capability of the system, enabling it to maintain efficient and stable performance in more complex and dynamic environments.

In summary, the elastic architecture of big data information systems based on cloud computing not only provides strong technical support for big data processing, but also offers valuable practical experience and theoretical guidance for system design in related fields, with broad application prospects and development potential.

References

[1] Huang J. Research on Cloud Computing Resource Scheduling Method Based on Machine Learning [J]. Heilongjiang Science, 2025, (1): 124-126.

[2] Lan L, Liu Z Y. Design of power automation monitoring system based on cloud computing [J]. Electronic Engineering & Product World, 2025, (1): 53-55.

[3] Zhao J. Revision and Application of The Standard for Cloud Computing Service Security Capability Requirements [J]. Information Technology & Standardization. 2024, (12): 61-65.