Scholar Publishing Group
International Journal of Big Data Intelligent Technology
https://doi.org/10.38007/IJBDIT.2025.060216
ISSN 2790-0932 Vol. 6, Issue 2: 147-155

Design and Implementation of High-Concurrency Service Architecture in Advertising Data Platform

Jingtian Zhang

Georgia Institute of Technology, Atlanta 30332, Georgia, USA

Keywords: High Concurrency, Advertising Data Platform, Distributed Architecture, Load Balancing

Abstract: Faced with the explosive increase in data volume in the advertising industry, building a service architecture that can meet high concurrency demands has become particularly crucial. This article explores the architecture design and implementation path of an advertising data platform in high concurrency scenarios. Analyze aspects such as load balancing, data storage and distributed processing, cache mechanism optimization, and service fault tolerance and recovery. By introducing a distributed system architecture, optimizing traffic allocation, strengthening intelligent caching strategies, and enhancing fault tolerance, the platform can smoothly handle massive data requests, ensuring the smooth operation and high availability of the system. The proposed architecture design provides a feasible solution to address the high concurrency challenges of advertising data platforms, and provides practical references for the future expansion and optimization of the platform.

1. Introduction

In today's digital wave, an efficient advertising data platform is one of the core elements for the success of the advertising industry. With the continuous increase in demand, traditional architecture design can no longer meet the growing demand, and the processing efficiency and response time of the system have become limiting factors. How to create a service architecture with powerful concurrent processing capabilities, flexible scalability, and high stability while ensuring the accuracy of data and system reliability is an urgent problem that advertising platforms need to solve. This article will deeply analyze how to use advanced architecture design and technical measures to overcome the performance limitations encountered by the platform in the context of high concurrency, and propose practical optimization solutions to ensure the continuous and stable operation of the platform in high traffic environments.

2. Overview of High concurrency Service Architecture for Advertising Data Platform

2.1. Basic functions of advertising data platform

As the core tool for advertising placement and effectiveness evaluation, the advertising data platform undertakes the important tasks of data collection, processing, and analysis. The main function of the platform is to collect and process massive amounts of advertising information data in real-time. By utilizing various data collection methods, the system can capture real-time interaction details between users and advertisements. The management of advertising activities is also one of the key functions of the platform, and its system design should be able to cope with the coordination and optimization of a large number of advertising activities[1], ensuring the timeliness and accuracy of advertising releases. In the data analysis stage, the platform demonstrates its powerful evaluation capability, which can output detailed performance analysis reports to assist advertising publishers in adjusting strategies based on actual data. The platform also has the ability to analyze and predict user behavior, using big data technology and machine learning to predict potential user behavior trends, thereby providing decision support for advertising strategies. Figure 1 shows the core architecture of the advertising data platform, covering key functional modules such as data collection, processing, storage, analysis, and feedback:

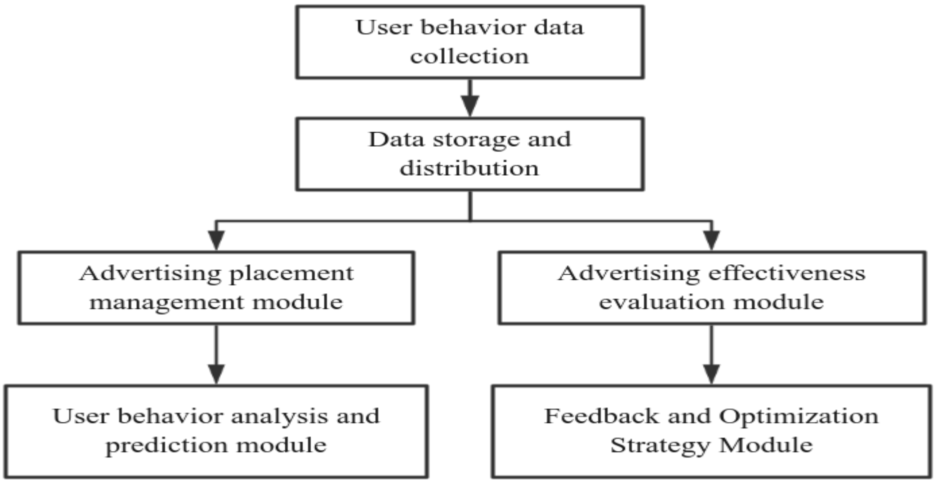
User behavior data collection
Data storage and distribution
Advertising placement management module
Advertising effectiveness evaluation module
User behavior analysis and prediction module
Feedback and Optimization Strategy Module

Figure 1. Schematic diagram of advertising data platform architecture

2.2. Characteristics of High Concurrent Service Architecture

The stability and response speed of the advertising data processing center depend on the high-performance concurrent service architecture it adopts. The most prominent advantage of this architecture is its enormous processing power, which means that the system needs to efficiently handle massive synchronous requests in a very short amount of time. Whether it is collecting advertising placement information or conducting real-time analysis of user behavior data, platforms must maintain extremely high processing efficiency. The strict requirements of the platform for response time cannot be ignored, especially when providing real-time feedback on advertising effectiveness and predicting user behavior, reducing latency is particularly crucial[2]. By improving data storage and computing methods, the platform aims to shorten data processing time and enhance overall response speed. Faced with the continuous increase in the number of users, the platform needs to maintain excellent scalability to ensure that the architecture can quickly adjust with the rise in request volume to adapt to the constantly expanding data volume and user access needs. The fault tolerance and high availability of the architecture are also essential to ensure that even in the event of component failures, the platform can still operate normally and advertising delivery will not be interrupted. In high concurrency architecture, load balancing mechanism is also a very important part, which can allocate requests to various server nodes reasonably, prevent individual nodes from being overloaded, and ensure the stable operation of the system when facing high concurrency requests.

3. Design of high concurrency service architecture for advertising data platform

3.1. High concurrency service architecture design goals and principles

When designing a high concurrency service architecture for an advertising data platform, the first step is to clarify the design objectives. Its core goal is to ensure that the platform can maintain its stability and efficient operation even in the face of massive concurrent requests. During the design process, designers must comprehensively evaluate the platform's processing capacity, response speed, and scalability. In the basic principles of architecture design, ensuring that the platform can achieve horizontal scalability is a core requirement, which involves the platform being able to flexibly adjust resource allocation according to actual load to prevent negative impact of resource constraints on service performance. During the design process, it is necessary to pay attention to the platform's fault tolerance capability, ensure its high availability, and reduce the interference of failures on business. Finally[3], the architecture needs to achieve high efficiency and accuracy in data processing, especially in real-time processing of advertising placement and user behavior data. The timeliness and precision of data processing are crucial.

3.2. System Architecture Module and Component Design

Advertising data platforms typically consist of multiple core modules, including data collection module, data storage module, computing and processing module, analysis and reporting module, etc. The data capture component is responsible for collecting information from different advertising channels and user interfaces, while also performing basic data filtering and preprocessing work. The function of data storage components is to store the collected information in a distributed database or cloud storage system, ensuring the stable availability and elastic expansion of the information. The data processing component conducts in-depth analysis of data according to established algorithms, involving advertising effectiveness evaluation, user behavior analysis, and other content. The data analysis and report components are linked with the database to achieve intuitive display of data and automatic generation of reports, helping advertising advertisers optimize their strategies in real time[4]. Each component must be connected through efficient interfaces to ensure efficient data transmission and processing speed. In the face of high concurrency, asynchronous communication should be used for interaction between components to prevent system congestion, improve processing efficiency and response performance.

3.3. Key Technologies and Tool Selection for High Concurrent Architecture

Choosing the appropriate technical architecture is crucial in the face of high concurrency requests from advertising data platforms. Firstly, a multi-level load balancing strategy is the foundation for ensuring concurrent processing capability, which can evenly distribute user and advertiser requests to various servers, preventing system paralysis caused by single point overload. Secondly, an efficient caching strategy is crucial for improving system performance, especially in scenarios where data hotspots are frequently accessed. By utilizing caching to reduce database burden, response speed can be accelerated and processing efficiency can be improved. Thirdly, the data storage solution must be carefully selected. Given the complexity and variability of advertising data, distributed databases can significantly enhance the read-write performance of data and ensure the flexible expansion of the system. In order to ensure real-time and efficient data processing, stream processing frameworks such as Apache Kafka and Apache Flink are adopted. Fourthly, in terms of service management, microservice architecture is used to enable independent deployment and expansion of each service module, which not only simplifies the system structure but also enhances maintainability. The construction of the overall architecture relies on efficient distributed technology and big data processing frameworks, ensuring that the advertising data platform can maintain efficient operation even when handling massive requests.

3.4. Safety and Maintainability Design

In the design of high concurrency architectures, security and maintainability are essential considerations. In order to achieve platform security, data encryption and permission restriction policies need to be addressed from the beginning of the design. At the data management level, encryption operations are performed on critical information during transmission to avoid the risk of data tampering or leakage during transmission[5]. By implementing a multi-level permission management mechanism, it ensures that only authorized users have access to the corresponding data resources, thereby strengthening the security defense of the system. For maintainability in high concurrency scenarios, the design process should focus on modularity and loose coupling of the system. Each functional unit can operate and maintain independently, which not only reduces the complexity of the system, but also facilitates future functional upgrades and expansions. In terms of system monitoring, by integrating logging and real-time monitoring tools, anomalies can be detected and handled in real time during system operation, ensuring the reliable operation of the system. The architecture design should also include automatic deployment and version control, utilizing containerization technology and continuous integration methods to simplify the system maintenance process and ensure stable operation of the system in high concurrency situations.

4. Implementation strategy for high concurrency service architecture of advertising data platform

4.1. Load balancing and traffic scheduling

In high concurrency environments, load balancing is the key to ensuring the stable and efficient operation of advertising data platforms. The primary step in load balancing is to implement real-time monitoring of server nodes and evaluate the load status of each node by analyzing traffic dynamics. When a user initiates a request, the load balancer distributes the request to various servers according to a set allocation rule, such as polling, weighted polling, or minimum number of connections. Given the significant changes in traffic on advertising data platforms, equalizers must be equipped with automatic scaling capabilities[6]. When encountering a surge in traffic, the system should be able to automatically create server instances and direct requests to these newly added instances. In order to further optimize traffic scheduling, a content-based routing mechanism is adopted to ensure that various requests can be allocated to the corresponding service nodes for processing. This process not only improves the effective utilization of server resources, but also reduces the risk of single node overload. In the face of significant fluctuations in advertising data platform traffic, the automatic expansion function of load balancers is particularly important. Once the traffic surges, the system can automatically activate additional server instances to achieve effective traffic allocation, as shown in Figure 1. This figure illustrates in detail how a load balancer performs dynamic scheduling based on traffic data and optimizes traffic allocation through scheduling strategies and automatic scaling.

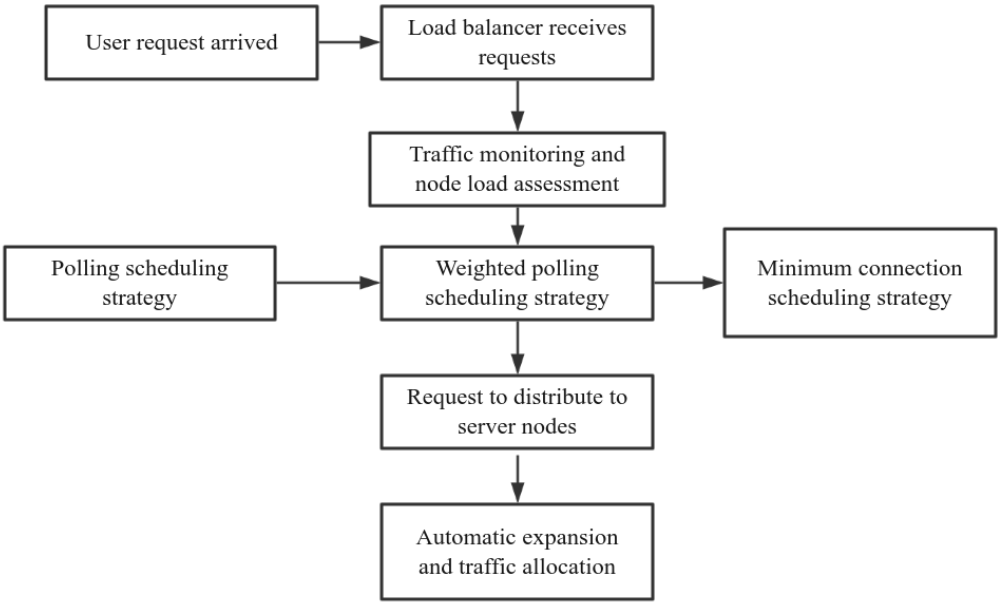
User request arrived
Load balancer receives requests
Traffic monitoring and node load assessment
Polling scheduling strategy
Weighted polling scheduling strategy
Minimum connection scheduling strategy
Request to distribute to server nodes
Automatic expansion and traffic allocation

Figure 2. Flow Chart of Load Balancing and Traffic Scheduling

Table 1. Data Storage and Distributed Processing Technology for Advertising Data Platform

Technical type	technical realization	Applicable scenarios	key features
Structured data storage	MySQL Cluster, CockroachDB	Store structured data to ensure data consistency and high availability	Support ACID transactions, high availability, and data consistency
Unstructured data storage	HDFS, Ceph	Store large-scale log data and unstructured data	Efficient storage and strong scalability
Distributed Processing Framework	Apache Kafka, Apache Flink	Real time data stream processing, real-time calculation of advertising clicks and display data	High throughput, low latency, streaming processing
Sharding technology	Distributed database sharding	Improve data access speed and decentralized storage of large-scale data	Uniform data distribution and improved read and write efficiency

4.2. Data Storage and Distributed Processing

The key to ensuring the stability of an advertising data platform for high concurrency needs lies in an efficient data storage solution and distributed data processing capabilities. When choosing a distributed storage solution, the system needs to adapt to the specific characteristics of the data type[7]. For storage of fixed format data, distributed SQL databases (such as MySQL Cluster or CockroachDB) can be used to ensure data integrity and stable system operation; For data without a fixed format or large amounts of log information, it is suitable to choose a distributed file storage system (such as HDFS or Ceph) to achieve efficient storage. In the distributed storage architecture of data, a sharding strategy is implemented to distribute a large amount of data among numerous nodes, aiming to enhance the read and write speed of data and system efficiency. As shown in Table 1, various data storage and processing methods need to be selected based on actual application requirements to ensure that the system can efficiently and stably process massive amounts of data in high concurrency situations.

In terms of data processing, in order to handle massive amounts of data under high concurrency requests, the system introduces a streaming computing framework for real-time data stream processing. The data stream is collected from various sources into the Kafka cluster, and Flink processes the stream data upon arrival, calculating real-time information such as advertising effectiveness and user behavior analysis. Through the integration of data storage and distributed processing, the platform has achieved rapid processing and storage of data under high concurrency conditions.

Table 2. Optimization Strategy and Technical Implementation of Cache Mechanism for Advertising Data Platform

optimization strategy	technical realization	role
Cache range determination	Using Redis or Memcached	Cache frequently accessed and updated stable content, such as advertising materials, user profiles, etc.
Cache preheating	Load hotspot data into cache during system startup	Ensure that hotspot data is cached at system startup and respond quickly to user requests.
Cache expiration mechanism	TTL（Time-to-Live）set up	Ensure that cached data is not outdated and reload from the database when cache fails.
Cache penetration and cache breakdown protection	Bloom filter, locking mechanism	Prevent abnormal requests for caching data in high concurrency situations and ensure stable data provision.
Cache update strategy	Regularly update cache, asynchronously update cache	Ensure that cached data is consistent with the data in the database, reflecting the latest changes.

4.3. Cache mechanism optimization strategy

In order to enhance the performance of advertising data platforms in high concurrency scenarios, deep adjustment of caching strategies is a key step. The primary task of the platform is to determine the application scope of the cache, select data with high access frequency and low update frequency[8], and use distributed caching technology for data storage. In order to increase the success rate of caching, the platform must implement cache preloading and automatic refresh mechanisms. During the system initialization phase, high-frequency access data is automatically loaded into the cache for quick processing of user requests. Implementing time management of cached data by setting TTL values to ensure data freshness. Once cached data expires, the system will automatically extract the latest data from the database. The platform also uses Bloom filters and locking mechanisms to ensure that cached data can be stably provided to users in high concurrency situations. The cache update strategy cannot be ignored, and the platform needs to design two methods: timed refresh and asynchronous refresh of cache to maintain synchronous updates between cached data and the database. Based on different business requirements and data characteristics, the improvement of this series of caching mechanisms ensures that the advertising data platform can respond quickly when facing high concurrency requests, effectively reducing the burden on the backend database. Table 2 summarizes the optimization strategies and corresponding technical implementations of the caching mechanism.

4.4. Service Fault Tolerance and Fault Recovery Mechanism

The stability and availability requirements of advertising data platforms are very high in high concurrency environments. Therefore, the platform implements a service backup strategy, with core business systems evenly distributed across multiple server nodes and relying on load balancers to achieve reasonable resource allocation. Once a node fails, the system will automatically redirect the request to a functioning node to ensure seamless service integration. At the same time, the platform builds a service status monitoring system and regularly checks service instances to quickly detect and handle abnormal situations. Once a service instance anomaly is detected, the system will immediately activate an isolation mechanism to remove the node from the load allocation list, preventing the spread of the fault. The platform also needs to be equipped with an automated fault recovery process, leveraging cloud computing container technology to quickly deploy new containers in the event of a fault, taking over the work of the faulty instance and minimizing the time of system failure. At the data level, the platform has developed data backup and disaster recovery strategies, regularly copying critical data to remote storage to ensure rapid data recovery even in the event of major system failures, ensuring uninterrupted business operations.

5. Conclusion

In high concurrency environments, the architecture design and implementation of advertising data platforms focus on load balancing, distributed data storage, improving caching strategies, and enhancing service fault tolerance and fault recovery capabilities. These strategies work together to cope with the dual pressure of massive data and concurrent requests. With the continuous increase of advertising data volume, the scalability and adaptability of the platform have become crucial. By continuously improving technical solutions and architectural layout, the platform can accelerate response time, enhance processing efficiency, and meet the constantly rising demands of the advertising industry. In the future, with the continuous innovation of technology, the architecture of advertising data platforms will also evolve and upgrade to better adapt to the dynamics of the market and the direction of technological advancement.

References

[1] *Lai S, Sun M, Huang B. Design of Accurate Placement Method for Film and Television Advertisements Based on Digital Twin and Data Mining. Media and Communication Research, 2023, 4(9)*

[2] *Yu Y, Sheng C. A Study on Data Monitoring and Effect Optimization of Programmed Advertising Platform: Taking “Ocean Engine” as an Example. Journal of Physics: Conference Series, 2021, 1883(1):*

[3] *John W, Nicole H, Giang T. Finding creative drivers of advertising effectiveness with modern data analysis. International Journal of Market Research, 2023, 65(4):423-447.*

[4] *Xiaoling G, Sijie H. Research on Administrative Supervision of Internet Advertising in Guangdong Province in the Era of Big Data. Industrial Engineering and Innovation Management, 2022, 5(5)*

[5] *Kamala H. Development of an Effective Method of Data Collection for Advertising and Marketing on the Internet. International Journal of Mathematical Sciences and Computing (IJMSC), 2021, 7(3):1-11.*

[6] *Jing L. Research on "Precise Translation" of Commercial Advertising Based on Big Data.Journal of Physics: Conference Series, 2021, 1744(3):032121.*

[7] *Su H, Luo W, Mehdad Y, et al. Llm-friendly knowledge representation for customer support[C]//Proceedings of the 31st International Conference on Computational Linguistics: Industry Track. 2025: 496-504.*

[8] *J. Huang, "Performance Evaluation Index System and Engineering Best Practice of Production-Level Time Series Machine Learning System," 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India.*

[9] *J. Huang, "Research on Multi-Model Fusion Machine Learning Demand Intelligent Forecasting System in Cloud Computing Environment," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-7, doi: 10.1109/IACIS65746.2025.11210946.*

[10] *W. Han, "Using Spark Streaming Technology to Drive the Real-Time Construction and Improvement of the Credit Rating System for Financial Customers," 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-6, doi: 10.1109/ICICNCT66124.2025.11232932.*

[11] *M. Zhang, "Research on Joint Optimization Algorithm for Image Enhancement and Denoising Based on the Combination of Deep Learning and Variational Models," 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-5, doi: 10.1109/ICICNCT66124.2025.11232800.*

[12] *Y. Zou, "Research on the Construction and Optimization Algorithm of Cybersecurity Knowledge Graphs Combining Open Information Extraction with Graph Convolutional Networks," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-5, doi: 10.1109/IACIS65746.2025.11211353.*

[13] *Q. Xu, "Implementation of Intelligent Chatbot Model for Social Media Based on the Combination of Retrieval and Generation," 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Hassan, India, 2025, pp. 1-7, doi: 10.1109/IACIS65746.2025.11210989.*

[14] *B. Li, "Research on the Spatial Durbin Model Based on Big Data and Machine Learning for Predicting and Evaluating the Carbon Reduction Potential of Clean Energy, " 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-5, doi: 10.1109/ICICNCT66124.2025.11232698.*

[15] *Y. Zhao, "Design and Financial Risk Control Application of Credit Scoring Card Model Based on XGBoost and CatBoost," 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025, pp. 1-5, doi: 10.1109/ICICNCT66124.2025.11233033.*

[16] *Shen, D. (2025). Construction and Optimization of AI-Based Real-Time Clinical Decision Support System. Journal of Computer, Signal, and System Research, 2(7), 7-13.*

[17] *Hu, Q. (2025). The Practice and Challenges of Tax Technology Optimization in the Government Tax System. Financial Economics Insights, 2(1), 118-124.*

[18] *Wei, X. (2025). Deployment of Natural Language Processing Technology as a Service and Front-End Visualization. International Journal of Engineering Advances, 2(3), 117-123.*

[19] Ding, J. (2025). Research On CODP Localization Decision Model Of Automotive Supply Chain Based On Delayed Manufacturing Strategy. arXiv preprint arXiv:2511.05899.

[20] Wu Y. Software Engineering Practice of Microservice Architecture in Full Stack Development: From Architecture Design to Performance Optimization. 2025.