Serverless architecture in cloud computing

**Abstract：Serverless computing is a new cloud computing model. In this model, applications are broken down into fine-grained functions and uploaded to the platform. Developers only need to write and upload code, and the cloud platform will automatically manage and allocate resources based on the actual used computing power. Developers do not need to worry about server configuration, maintenance, updates, and expansion, etc. Although the name contains the word "serverless", actual physical servers are still needed to support service operation, but the management and maintenance of these servers are transparent to developers and fully managed by cloud service providers.**

**I. Introduction**

**In recent years, as consumer needs have changed and business complexity has increased, traditional application development and deployment methods have faced many challenges. The serverless architecture provides enterprises with higher flexibility and cost-effectiveness through the event-driven model and on-demand billing strategy. Moreover, with the rise of modern microservice architecture, the combination of serverless architecture and microservices makes it more efficient and convenient to build and deploy complex applications. However, serverless architecture is not without controversy. It also brings new challenges such as cold start delay, security, and vendor lock-in. This paper will conduct in-depth discussions on the basic concepts, advantages, and challenges of serverless architecture.**

**II. Research status of serverless architecture in cloud computing at home and abroad**

**1. Current situation in China**

**Chinese universities have accumulated a lot of research results in the theoretical basis, model construction, and application scenarios of serverless computing. The research scope covers cloud service optimization, application migration, and cost management. Chinese cloud service providers (such as Alibaba Cloud, Tencent Cloud, and Baidu Cloud) have launched serverless computing products to help enterprises achieve higher flexibility and cost-effectiveness. Many industries (such as finance, the Internet, and education) have begun to try serverless architecture to reduce operational costs and improve development efficiency.**

**2. Current situation abroad**

**Some universities and research institutions have conducted in-depth research on the performance, scalability, and security of serverless architecture. For example, focusing on the cold start problem, performance evaluation and optimization. Many technology gi\*\*\* Azure have provided mature serverless solutions on their platforms, promoting the popularization and application of server**

**Ⅲ. Basic Concepts**

**Serverless Architecture is a design pattern in cloud computing that allows developers to focus on writing code and business logic without worrying about the management and maintenance of underlying servers. Its basic concepts include:**

**1. Event-driven: Serverless architecture is typically based on an event-driven model, where code is designed as functions that respond to specific events such as user requests, database changes, or message queue events.**

**2. Pay-per-use: In serverless architecture, users only pay for the actual used computing resources. This means that when a function is not called, users will not incur any charges.**

**3. Automatic scaling: Serverless platforms are typically able to automatically scale, dynamically allocating resources based on demand during high load and automatically reducing resources during lower load, which can improve application availability and performance.**

**4. No infrastructure management: Developers do not need to manage or maintain servers or infrastructure, as cloud service providers take care of these tasks, allowing developers to focus more on application logic and user experience.**

**5. Microservices architecture: Serverless architecture is often combined with microservices architecture, where an application is decomposed into small, independent services that can be deployed and scaled independently.**

**The advantages of serverless architecture include reduced operational costs, improved development efficiency, and accelerated product launch speed. However, challenges such as cold start time, debugging difficulty, and vendor lock-in must also be considered.**

**Ⅳ. Application scenarios**

**The application scenarios of serverless architecture include "low cost, high elastic expansion, and simplified operation and maintenance," making it suitable for various event-triggered, load-fluctuating, and backend processing business scenarios.**

**1 Event-triggered scenarios. When a certain event occurs, a Serverless function is called to process it, such as file upload, message queue message event, timer-triggered event, and IoT device event processing scenarios.**

**2 Traffic spike scenarios. For some internet businesses with traffic spikes, Serverless architecture can be used to achieve high elastic expansion.**

**3 Data processing scenarios. For scenarios where large amounts of data need to be processed periodically, such as data ETL operations, large-scale data analysis and computation.**

**4 Backend processing scenarios. As the backend of mobile applications and Web applications, it can implement the backend service of HTTP applications.**

**Ⅴ.Study on Optimising Function Launch Latency in Cloud Computing**

**A significant aspect of serverless architecture is its ability to reduce the number of function instances to zero, meaning users are not charged when a function is deleted. When a deleted function is called again, the platform restarts a new instance and redeploys the runtime environment and code. This process, known as a cold start, can cause delays in function startup, negatively impacting the responsiveness of serverless cloud platforms. Currently, some platforms implement basic optimization strategies like preheat plugins and container pools. However, these strategies often follow a fixed execution cycle without considering the platform's operational state, leading to resource wastage and increased load.**

**To mitigate cold starts and enhance application execution efficiency, developing appropriate optimization strategies is essential. One common approach is the use of warm-up plugins, which periodically activate a user-defined function set to keep them ready for immediate use. For instance, the Serverless WarmUpPlugin typically wakes up functions at user-specified intervals, with a default setting of five minutes. This ensures the function containers remain active, thereby reducing the likelihood of cold starts. This method is effective and cost-efficient for cold start optimization.**

**Container pools are another widely adopted solution in serverless cloud computing for addressing cold starts. For example, the open-source platform Hssion utilizes container pools to maintain a reserve of free containers. When a function request is made, the platform can quickly deploy an empty container from the pool, significantly cutting down the time needed for container creation. More advanced container pools categorize runtime environments by programming language, such as Python or Java, preloading the necessary environments and maintaining separate sub-pools for each language. This not only accelerates container deployment but also reduces the time required for setting up the runtime environment.**

**Ⅵ Research on Cloud Resource Management Techniques without Servers**

**Cloud computing leverages virtualization technology to offer users a highly scalable computing model. However, data centers face a significant challenge with low utilization of cluster resources. Serverless architecture has emerged as a preferred approach for application development, as it allows enterprises to lower deployment and operational costs by eliminating the need for server provisioning, maintenance, and capacity reservation for computing and storage services. A key feature of serverless architecture is its high scalability, characterized by on-demand scaling and a pay-per-use model.**

**From the perspective of cloud service providers, the widespread implementation of serverless architectures necessitates effective optimization and management of cloud resources to minimize operational costs. Thus, managing data center resources efficiently is crucial for reducing energy consumption, enhancing resource utilization, and lowering operating expenses.**

**Virtual Machine (VM) deployment involves setting up a series of virtual machines on servers that meet specific operational requirements, following a strategy aimed at fulfilling the needs of cloud users within a data center with limited resources. Poorly planned VM deployment can lead to suboptimal resource utilization, higher service costs, and unnecessary energy waste. VM scheduling, on the other hand, is the process of assigning tasks to the appropriate virtual machines for execution. The effectiveness of the VM scheduling algorithm directly influences user satisfaction and the overall quality and efficiency of the cloud platform. Therefore, implementing a sound VM scheduling algorithm is essential for addressing cloud resource management challenges.**

**The Ant Colony Optimization (ACO) algorithm is particularly effective for quickly finding optimal solutions in smaller-scale task allocation scenarios and is considered a viable option for managing cloud resources.**

**Ⅶ Research on Cloud-Native Applications Based on Serverless Architecture**

**Serverless computing represents a modern approach to developing, deploying, and maintaining cloud-native applications. Key features include being event-driven, stateless, having short execution times, enabling agile automatic scaling, and offering cost efficiency. Applications are broken down into multiple stateless, fine-grained functions that can be developed, updated, and reused independently. Each function adheres to the single responsibility principle, meaning it performs one specific task. This stateless nature facilitates horizontal scaling and supports high business concurrency. Functions run within containers, which are easy to package, deploy, and transfer across different platforms. Once a function completes its execution, its container can quickly instantiate a new function or execute on a schedule, helping to keep the container warm and minimizing cold starts. The serverless model requires no maintenance, allowing developers to concentrate on business logic while cloud providers handle the infrastructure and runtime, simplifying overall operations. Additionally, usage-based billing helps lower costs. Serverless architecture effectively decouples infrastructure from application logic.**

**Ⅷ conclusion**

**This study investigated the advantages and disadvantages of the serverless architecture in cloud computing, as well as the areas that need improvement. The research findings show that the serverless architecture is suitable for fast-growing and highly scalable application scenarios, providing flexible, efficient, and economical solutions for enterprises, with automatic scaling capabilities to increase or reduce computing resources according to load, providing better performance and response time. However, it may not be ideal for tasks that require continuous operation or high-performance computing. In the future, we can further study to improve the computing performance and security of the serverless architecture in cloud computing, so that it can be better applied in real life.**

**Ⅸ outlook**

**Serverless architecture, as a new computing model, has a broad future development prospect. However, there are still some challenges and problems that need to be solved. For example, how to handle real-time data processing, high-concurrent event processing, and ensure the security of data during transmission and storage. Future research can be conducted in these areas to further improve the practical application value of the serverless architecture in cloud computing. In addition, we also look forward to the future serverless architecture providing more advanced functions and services, covering artificial intelligence, machine learning, and big data processing, etc. Cloud service providers launch more native support for serverless services, enabling developers to easily deploy complex。**

References

1. 徐政钧.无服务器云计算平台中函数启动加速器的设计与实现[D].北京邮电大学,2020.DOI:10.26969/d.cnki.gbydu.2020.000956. .
2. 裴云曼.基于无服务器架构的云资源管理技术研究[D].华北电力大学(北京),2020.DOI:10.27140/d.cnki.ghbbu.2020.001005.
3. 曾世伟.云计算与大数据在企业信息化中的融合应用研究[J].网络安全技术与应用,2024,(09):124-126.
4. 崔华.基于云计算的应用服务高并发优化技术研究[J].中国新技术新产品,2024,(15):35-38.DOI:10.13612/j.cnki.cntp.2024.15.040..
5. 柴琳.唐磊：云计算创新平台推动企业数字化转型升级[J].中国商人,2024,(08):104-105.
6. 蒲云鹏;.无服务器架构与应用分析[J].信息与电脑(理论版),2022,(04):36-39.
7. 张国生;.基于无服务器架构的云原生应用软件研究[J].中国电子科学研究院学报,2022,(02):55-61.