Application of microservice architecture in cloud environment project development

Ling Zheng*, and Bo Wei
School of North China Electric Power University, Beijing, China

Abstract. With the development of the information age, business systems are becoming more and more complex. System development and maintenance are facing huge challenges. In response to this problem, a unified application development platform based on the microservice architecture is proposed. Compared with the traditional single-architecture architecture, the microservices architecture can split a large and complex application system into a series of service modules that can be independently developed, tested, deployed, operated, and upgraded. This enables the application expansion and application reduction for a large number of Internet companies. Developing complexity and implementing agile development provide more effective methods. This article through a detailed case analysis - the development of the cloud platform system, describes the specific application of the microservice architecture in the actual project development, and discusses the advantages of the traditional single architecture model for the service architecture to build the system. Through research and analysis, it is concluded that the microservice architecture has certain guiding significance for solving problems that may be encountered in enterprise-level applications.

1 Introduction

Cloud computing provides a model for enterprise applications that can allocate computing resources on demand. Enterprises can deploy applications on IaaS, PaaS, or SaaS service platforms. In this paper, we take PaaS platform-as-a-service cloud platform system as an example [1]. The PaaS cloud platform construction goal based on microservice architecture and Docker container technology is to provide our developers with a set of services for rapid development, deployment, operation and maintenance management, Continue to develop continuous integration processes [2]. The platform provides resources such as infrastructure, middleware, data services, cloud servers, etc. Developers only need to develop business code and submit it to the platform code base, do some necessary configuration, the system will automatically build, deploy, and achieve agile development and rapid application. Iterate. In the system architecture, PaaS cloud platform is divided into three parts: micro service architecture, Docker container technology, and DveOps.

The microservice architecture plays a crucial role in the architecture of the cloud platform. Its general access path is: external requests → load balancing → service gateways

* Corresponding author: zhengling@ncepu.edu.cn

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).
(GateWay) → microservices → data services/message services [2]. From its access path, we can see that implementing microservices requires a lot of technical effort to develop infrastructure, which is obviously unrealistic for many companies. However, the industry has a very good open source framework for our reference. The mature microservice frameworks in the industry include Netflix, Spring Cloud and Ali's Dubbo. Spring Cloud is a set of frameworks for implementing microservices based on Spring Boot. It provides the components needed to develop microservices. When used with Spring Boot, developing a cloud service for microservices architecture becomes very convenient.

The content of this article is organized: The first part introduces the basic application and development background of the microservice architecture and the advantages of the cloud platform system application for the service architecture. The second part combines the monolithic architecture model and the microservice architecture model that are actually applied to the cloud platform system. Both of them describe the structure and function. The third part uses specific experimental data to illustrate the difference between the two architectures, discusses the advantages of the microservice architecture in the establishment of the cloud platform system, and further explains the advantages of the microservice architecture in project development. The fourth part is the summary of the article and explains the further work.

2 Monolithic Architecture and Microservice Architecture

This chapter lists in detail the structure and function of two different architectures in the development of the PaaS-based cloud platform management system, and analyzes the influencing factors such as the efficiency of the two models according to the structure. Taking the three parts of the virtual machine module, the host module, and the cluster module in the cloud platform system as examples, the structure of two different architecture methods is shown.

2.1 Monolithic architecture model

The monolithic architecture model is a more traditional architectural model. An archive package (for example, the war format) contains all the application's application functions. We usually call it monolithic applications. The monolithic application methodology of the architecture is what we call the monolithic application architecture. Figure 1 shows the main structure of the monolithic architecture model.

![Fig. 1. Figure with monolithic architecture model.](image)

The monolithic architecture model also has many advantages [1]:
1) It is well known: most of the existing tools, application servers, frameworks, and scripts are such applications;
2) IDE-friendly: Development environments such as NetBeans, Eclipse, and IntelliJ are designed for single applications such as development, deployment, and debugging;
3) Easy to test: Once a single application is deployed, all services or features can be used, which simplifies the testing process because there is no additional dependency, and each test can be started immediately after the deployment is completed;
4) Easy to deploy: Just copy a single archive file to a single directory.

With the wide application of the concept of cloud computing, we must fully consider the high scalability of the entire system, the ultra-large scale of virtualization, the variability of interfaces, and the higher-level security during software development and design. The development of the entire system is more complex and changeable, and the traditional monolithic architecture mode is no longer applicable. For example, if you use the monolithic architecture mode to develop a cloud platform management system, the structure will appear to be single and unstable. The following figure shows the design of the monolithic architecture of the system [3].

![Fig. 2. Figure with monolithic architecture mode under cloud platform.](image1)

The monolithic architecture under the cloud platform shown in the figure above contains all the functional modules. It is highly complex. The entire project contains many modules, the boundaries of the modules are blurred, and the dependencies are not clear. The quality is uneven and it is easy to have bugs in actual development.

### 2.2 Microservice architecture model

The concept of microservices was proposed by Martin Fowler and James Lewis in 2014. In just a few years, more and more companies have begun to try to use the microservice architecture to build a distributed system that is centered on business and fine-grained.

In short, the microservices architecture style development approach is to develop a standalone application system by developing a small set of services. Each of these small services runs in its own process and often uses the lightweight mechanism of the HTTP resource API to communicate with each other [4]. These services are built around business functions and can be deployed independently through a fully automated deployment mechanism. These microservices can be written in different languages and can use different data storage technologies. We only do minimal centralized management of these microservices.

![Fig. 3. Figure with microservice architecture model under cloud platform.](image2)
Applying the microservices architecture can solve many problems in development, but at the same time it also faces some challenges.

3 Experiments and results analysis

This chapter will use experimental data to discuss the specific data to explain why project development under the cloud environment is better using the microservice architecture model. In the previous chapter, we listed the structural diagrams corresponding to the two architectural patterns in cloud platform development [4]. We extracted three modules in the system, including cluster modules, host modules, and virtual machine modules. In addition, we added a S-module that does not depend on the database. The experimental part of this chapter takes these three modules as research objects and explores their overhead in the two architecture modes.

3.1 Experiment procedure

Both architectures are implemented using the Play Web framework. The Play framework is a lightweight, stateless, asynchronous cloud-friendly framework. Applications that use the Play framework can be deployed on Netty servers, a server that starts quickly and saves computing resources [5].

The system includes a relational database, and data manipulation is implemented through EBeans. The EBeans ORM is a fast and simple data access structure, and is Play's default object relational mapping model. Applications executed on the browser side are implemented in Angular.js because it supports Google very well [6].

The deployment of the monolithic architecture model adopts the structural model of figure 2, and several functions in figure 2 are a database instance. In addition, we also added a S-module that is not dependent on the database, that is, a cloud service instance.

The deployment of the microservice architecture pattern adopts the structure pattern of figure 3, and the different functional modules in figure 3 become a single microservice, and they are all database instances.

At this point, we have built two application projects under different architecture modes. In addition to the differences in the architecture modes, we fully ensure the consistency of other parameters in the two groups of comparative experiments, eliminating the impact of hardware and other factors on the experimental results. Run two sets of applications to count monthly expenses for different services and applications in the application. Finally, it shows the cost situation in the form of a table.

3.2 Experimental results and analysis

The system overhead of the two architecture modes is shown in table 1 and table 2. The bandwidth-related and bandwidth-related factors are consistent in both architectures.

Table 1. Infrastructure overhead for the singulated architecture deployment.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Hourly cost (USD)</th>
<th>Monthly operating hours</th>
<th>Monthly Expenditure (USD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational database instance</td>
<td>$0.464</td>
<td>72</td>
<td>$334.08</td>
</tr>
<tr>
<td>Cloud service instance</td>
<td>$0.090</td>
<td>72</td>
<td>$64.80</td>
</tr>
<tr>
<td>Monthly system overhead (USD)</td>
<td></td>
<td></td>
<td>$398.88</td>
</tr>
</tbody>
</table>
Applications executed on the browser side are implemented in Angular.js because it supports Google very well [6].

Both architectures are implemented using the Play Web framework. The Play framework can be deployed on Netty servers, a server that starts quickly and saves computing resources [5].

Applying the microservices architecture can solve many problems in development, but at the same time it also faces some challenges. In the previous chapter, we listed the structural diagrams corresponding to the two development under the cloud environment is better using the microservice architecture.

This chapter will use experimental data to discuss the specific data to explain why project module is under the responsibility of a dedicated development team. Independently of other micro-service development teams, only responsible for With its own micro service

In addition, we also added a S-module that is not dependent on the database, that is, a cloud service instance.

Table 2. Infrastructure overhead for microservice architecture deployment.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Hourly cost (USD)</th>
<th>Monthly operating hours</th>
<th>Monthly Expenditure (USD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microservices - Cluster Module</td>
<td>$0.067</td>
<td>72</td>
<td>$48.24</td>
</tr>
<tr>
<td>Microservices - Host Module</td>
<td>$0.232</td>
<td>72</td>
<td>$167.04</td>
</tr>
<tr>
<td>Microservices - VM Module</td>
<td>$0.067</td>
<td>72</td>
<td>$48.24</td>
</tr>
<tr>
<td>Cloud service instance</td>
<td>$0.090</td>
<td>72</td>
<td>$64.80</td>
</tr>
<tr>
<td>Monthly system overhead (USD)</td>
<td></td>
<td></td>
<td>$328.32</td>
</tr>
</tbody>
</table>

Analyze the performance of the two architectures on the premise that the business requirements in table 1 are satisfied. We use the same instance on AWS for comparison [7]. In the test, by configuring JMter to simulate a stable workload, Service 1 executes 30 requests per minute, and Service 2 executes 1100 requests per minute [8]. The average response time and the 90% response timeline for each of the two frameworks are shown in figure 4. As can be seen from the figure, both frameworks meet the business requirements presented in this chapter and validate the following conclusions:

1) Microservices will not delay the response time due to the excessive number of service hosts;

2) Microservices allow more granular instance types to reduce system overhead. In this case, the microservice architecture reduces infrastructure service costs by 17%.

Fig. 4. Figure with average response time of two service architectures.

Through the analysis of the above experimental data, the use of microservices architecture in the development of large-scale projects such as cloud environment projects can effectively improve operating efficiency and reduce system overhead [8]. In the choice of development method, different from the unified code library used in the development of the singulated architecture system, each development team of the micro-services architecture is relatively independent, they do not need to care about the details of the work of other micro-service development teams, only responsible for With its own micro service modules, each development team can use different technology stacks to implement micro services according to their technical characteristics and business requirements.

4 Summary

Through the analysis of the experimental results, we have seen some advantages and disadvantages of the microservice architecture. The most prominent advantage is that it can split a complex application into a series of well-defined service modules. Each service module is under the responsibility of a dedicated development team. Independently of other
service modules, you can independently develop, test, deploy, and expand upgrades. At the same time, unlike the unified code base model of the singulated architecture, each micro-service in the micro-services architecture application system has its own code base and is maintained separately. Therefore, in the development of projects under the cloud environment, applying the microservices architecture model is a good choice.

References