Application and Practice of Microservice Architecture in Multidimensional Electronic Channel Construction

To cite this article: Hao Zhang et al 2019 J. Phys.: Conf. Ser. 1168 022023

View the article online for updates and enhancements.
Application and Practice of Microservice Architecture in Multidimensional Electronic Channel Construction

Hao Zhang¹, Ying Xu², Wenjie Cao¹*, Xiaohui Xu², Chun Zhou¹, Yin Liu¹

¹ Beijing GuoDianTong Network Technology Co.Ltd, Beijing 100070, China
² State Grid Zhejiang Electric Power Company, Hangzhou 310007, China

Abstract. State Grid information construction has gained great achievements in the wave of mobile Internet development and impact of the "Internet plus" action plan. At the same time, there are some problems in the informationizing of multi-dimensional marketing e-service channels, such as the overlapping of channels, the insufficient integration of channels and lack of data sharing in various channels. The complex business operation of the channels in the traditional single-architecture mode. After the development of the microservices architecture in the current industry research shows that microservices-based architecture model system can be very well to solve the problems encountered in the construction of information technology. The final multidimensional electronic channel system construction using Docker container technology as a microservices architecture technology solutions, which greatly improved the system capacity and operation and maintenance the efficiency of system, and also provided a strong technical support for the construction of national network information.

1. Introduction

In recent years, the development of the "Internet plus" program and the mobile Internet technology have been vigorously developed and widely used. All walks of life continue to use the “Internet plus” as the driving force, we carry out the construction of e-channels by making full use of the major platforms of the Internet, to provide our clients with high quality service. Online and offline marketing channels complement each other and embody the three-dimensional and high-efficiency of marketing services. The development of electronic channels also presents a new ecology of business marketing. The State Grid Marketing System has expanded the 95598 intelligent interactive platform, the State Grid Mall, the Power at Hand, and the 95598Pay and other e-channels have greatly enhanced the online marketing service capabilities of the State Grid Corporation. But, in the process of building electronic channels such as Power at Hand and 95598Pay etc, problems such as single operating methods and poor operational results have been exposed. At the same time, there are overlaps in business demands among various channels, and there are crossovers in the management boundaries, which has led to repeated construction between channels. Aiming at the above problems, the recently developed rapid microservice architecture model[1,2,3] is adopted. The Docker container technology is used specifically to implement the microservice system. The entire service of each channel is split into microservices with specific functions, therefore each service is operated in separate processes. In this way, each functional module can be independently developed into a service, and the services can coordinate and cooperate with each other. The overlapping of business needs forms an independent public basic service for the use of various channels to solve the problem that the overlapping of business needs of various channels cannot be integrated.
2. Introduction to Microservice Architecture

Microservice Architecture is an architectural model\cite{4}. Its main idea is to split large systems into small systems, thereby reducing system complexity, reducing system upgrades, operational risks and costs. The specific implementation is to split the application of the monolithic architecture into small services and maintain the cooperation among the services to provide users with calls. The main thing is that each split service runs in its own process. The collaboration between each service uses a lightweight communication mechanism such as the RESTful API. In this architecture, each service is built around a specific business and deployed independently into the production environment. In addition, for specific services, technical languages can be selected for development based on business needs and team needs. The commonly components of the microservice include: service registration, service discovery, load balancing, service gateway, configuration center, API management, integration framework, distributed transaction, invocation chain, support platform and so on. At present, each component has excellent and good performance solutions provided by major manufacturers.

2.1 Benefits of Microservice

For large-scale complex systems, the microservice Architecture has the following advantages over the traditional monolith architecture model:

- Development efficiency. For complex projects, adopting the microservices Architecture will increase the workload for early design and communication, but as the project continues to advance, the development efficiency can be maintained at a stable level; the monolith architecture model will be significantly less efficient in the later stages of the project.

- System design. In the microservice architecture model, each service corresponds to a service module. Each service is independent of each other. Therefore, data and code are also physically isolated. It is easier to achieve high cohesion and low coupling. In the development process, the monolith architecture model is easy to cause all the services in the system to be coupled together at the entire data level. Therefore, it is difficult to achieve high cohesion and low coupling.

- System expansion. Because in the microservice, each service is in a cooperative relationship with each other, each microservice runs in a separate process, so it is easy to extend functions according to business requirements; in the monolith architecture model, modifications need to be made on existing systems, and it is easy to be highly coupled with existing services.

- System Upgrade. In the microservice architecture model, each service component can be upgraded independently, and the impact between each service is small, which can greatly improve the development efficiency; In the monolith architecture model, it is necessary to be familiar with the entire system before it can be modified, and it is easy to cause unexpected failure.

- Hardware requirements. In a system based on the microservice Architecture, how many micro-services are deployed in the system only needs to start the corresponding running container; in the monolith architecture system, resources need to be allocated for the entire system, which easily leads to redundancy.

- Project costs. Based on the microservice Architecture system, the cost of project early and late project changes is relatively flat; in the monolith architecture system, the cost of early projects is relatively low, and the latter is relatively large.

The above is the advantage of the microservice Architecture compared to the monolith architecture model. The microservice architecture also has its shortcomings. The microservice is a distributed application which is difficult to build, deploy and maintain.

2.2 Development of Microservice Based on Container Technology

In recent years, with the development of container technology, container technologies such as Docker and Kubernetes have developed into an industry standard for containerization and orchestration. Since the container technology uses a kernel interface, and the technology can realize that different containers can share the same kernel and isolate between containers, the container technology can largely solve the problems caused by the micro service architecture.
After several years of development, the use of Docker container technology to support the
development and operation of the microservices has been tested and verified by the vast majority of
manufacturers in the industry and has formed excellent technical solutions. This also shows that
without the flourishing development of Docker, there is no landing and blossoming of the
microservice Architecture.

3. Practice of Microservice Architecture in the Construction of Electronic Channels

3.1 Practice of microservice Architecture
The construction of electronic channels is fully in line with the microservice architecture usage
scenarios, the business of each channel is complex, not only the demand but also the scale is large.
Therefore, the development based on the traditional monolithic architecture model will result in late-
stage expansion and huge maintenance costs. Therefore, using Docker container technology\[5\] as a
support to build an electronic channel service system based on microservice Architecture. At present,
the entire system has been built and supports the business functions of each channel, and it also
reflects its high scalability. Based on the new requirements, the unified log management module and
the application-aware monitoring module have been extended in the microservice. The system
architecture based on the microservice in the electronic channel construction can be briefly presented
in Figure 1.

![System Architecture Diagram Based on Microservice Architecture.](image)

In Figure 1, the uppermost layer is the application layer. Each electronic channel service function
can be developed by different teams using different technologies and deployed at the application layer.
The access layer involves the stability of load balancing and high availability. The service layer
mainly provides service registration, service-to-service communication, and gray-scale release of
services. The last two layers mainly provide more infrastructure function support, which involves the
normal operation of the container cluster and the management of the mirror. At the same time, the
advanced operation and maintenance capabilities are improved through advanced services such as
container orchestration, automatic repair, and application monitoring. By doing this, we can shield the
underlying infrastructure so that each team can focus on business application development. In the
microservices architecture, many components are mainly implemented using open source and mature
technology solutions in the industry. A large number of mature third-party technology solutions have
greatly reduced the difficulty of developing micro-service architecture systems and provided stable
technical support for electronic channel construction.
3.2 Key Issues in Microservice Practice

Building a microservices, how to implement service registration, service discovery, load balancing, and service gateways are key issues in the overall system construction\textsuperscript{[6]}. The following content is the solution adopted in practice.

- Service Registration, Service Discovery. The microservice is a distributed network structure composed of multiple microservices. The communication between services uses a lightweight mechanism to communicate. The communication process between services includes requiring the service provider to register the service address to the service registry, and the service caller finds the service address to be called from the registry. This process involves service registration and service discovery. In the construction of an electronic channel microservices architecture system, Netflix's microservice solution was adopted using, Eureka as a service registration center, and Ribbon was used for service discovery and load balancing.

- Load balancing. The coordination between microservices requires high network requirements, and the physical location of the microservices is dynamic. Therefore, mature solutions are used to support load balancing when setting up the system. When deploying services, the physical location of each microservice may be changed. For this reason, when the microservice is created, a load balancer and a domain name will be correspondingly created for the service.

- Service Gateway. In the construction of electronic channel microservices architecture system, the system will contain many microservices due to the numerous business channels. At the same time, the system needs the API of these microservices for the front-end modules of the system and the APP to invoke. Therefore, the service gateway is required to provide a unified interface to invoke the microservice API and API authentication. The Spring Cloud Zuul technology solution is adopted here to implement the service gateway function.

- Unified log system. Because there are many major electronic channel services, in order to solve the problem of log file management when deploying service processes, a unified log management system based on ELK technology architecture is built. It greatly improves the efficiency of log management and query analysis under the microservice architecture.

- User-aware monitoring system. In the whole system construction, an application monitoring system was built to optimize the service architecture system structure and troubleshooting problems. The service monitoring module of the whole microservice system is established through log analysis in the unified log system. The response time and error rate of each service are monitored in real time, which improves the system operation and maintenance efficiency. At the same time, in the face of business expansion needs, the user-aware monitoring module is extended in the application monitoring module. Under this module, the services developed by different teams can be deployed to the corresponding independent processes according to the business needs for different user-aware indicators, and the service expansion capabilities of the system under the service architecture are embodied.

4. Summary

The rapid development of container technology has made the combination of microservices and containers a new architectural design concept. This paper mainly introduces the process of using microservice architecture system supported by container technology in the construction of multi-dimensional electronic channels, facing the complex business needs and existing problems of various channels. The whole process shows that the container technology can greatly simplify the microservice architecture, the service is easy to develop, the system is easy to maintain, and the business expansion capability can be guaranteed. At the same time, it provides powerful technical support for the subsequent construction of electronic channels.

Acknowledgments

This work is supported by the science and technology project of State Grid Corporation of China under the Grants No. 52110417001D.
References

[1] Genc Mazlami, Jürgen Cito, Philipp Leitner. (2017) "Extraction of Microservices from Monolithic Software Architectures", Web Services (ICWS) 2017 IEEE International Conference on, pp. 524-531.

[2] Tugrul Asik, Yunus Emre Selcuk. (2017) "Policy enforcement upon software based on microservice architecture", Software Engineering Research Management and Applications (SERA) 2017 IEEE 15th International Conference on, pp. 283-287.

[3] D Lu, DJ Huang, Andrew Walenstein, Deep Medhi. (2017) "A Secure Microservice Framework for IoT", Service-Oriented System Engineering (SOSE) 2017 IEEE Symposium on, pp. 9-18

[4] Florian Rademacher, Sabine Sachweh, Albert Zündorf. (2017)"Differences between Model-Driven Development of Service-Oriented and Microservice Architecture", Software Architecture Workshops (ICSAW) 2017 IEEE International Conference on, pp. 38-45.

[5] Hui Kang, Michael Le, Shu Tao. (2016)"Container and Microservice Driven Design for Cloud Infrastructure DevOps", Cloud Engineering (IC2E) 2016 IEEE International Conference on, pp. 202-211.

[6] Rory O'Connor, Peter Elger, Paul M. Clarke. (2016)"Exploring the Impact of Situational Context — A Case Study of a Software Development Process for a Microservices Architecture", Software and System Processes (ICSSP) 2016 IEEE/ACM International Conference on.