A Novel Think Tanks Evaluation System Based on Micro Service

Shiyong Xiong, Benguo Huang*

*Chongqing University of Posts and Telecommunications, Chong Qing 400065, China

*1030112547@qq.com

Abstract. With the development of new technology, the traditional single application architecture has been unable to meet a variety of needs of the existing system, in order to solve the inconvenience monomer architecture system maintenance, a series of problems such as poor extensibility, deployment of trouble, micro service architecture is to break up the system, each service function is simple, can complete the function of single responsibility, and deploy alone, without centralized management, a service outage will not affect other service work, reduce the coupling between modules, the system performance improvement. Therefore, micro-service architecture is adopted to develop a new think tank evaluation system. In response to the needs of local think tank evaluation, this paper designs a new think tank evaluation system based on micro-service. The system adopts micro-service architecture to design and develop, and splits the services according to the business of the system. The system was divided into basic service, registration service, audit service, declaration service, review service, scoring service and log service. The communication between the services was realized through RPC remote invocation based on Dubbo distributed framework and Zookeeper distributed application. The system realizes the separation of front and rear ends, each service of the server is implemented with Spring Boot framework, the front end is implemented with vue.js framework, the system use Mysql database, and Redis is used to improve the system performance.

Keywords: Micro service, Spring Cloud, A think-tank evaluation, Automatic grading

1. Introduction

With the continuous expansion of data scale, the evaluation system of various local think tanks is also gradually complicated and diversified. For the continuous development and maturity of think tanks, a perfect new think tank evaluation system must be established. Therefore, it is necessary to establish a reasonable evaluation system based on local needs and evaluate think tanks from various aspects. Only in this way can the think tank evaluation achieve the evaluation function and promote each think tank to constantly improve the indicators and move forward to a higher level of think tank.

Micro service[1-5] Architecture divides applications into small services with its advanced architectural style, and the integrated Model development method of MDA(Model Driver Architecture) also provides an implementation mechanism to solve the problems of software development efficiency and portability. Modeling development of micro-service architecture is realized by model driving, and micro-service modeling is divided into two processes: conceptual model and interaction model.
Scalability, agility, and reliability of the micro service architecture. In particular, vertical decomposition into appropriate granularity containing systems and micro services, as well as coupling, integration, scalability, and monitoring of micro services.

2. System function design

According to the process requirements of local think tanks, the overall process of the think tank evaluation system is shown in Fig.1. The overall process is as follows: the unit enters the system for registration, the social science administrator reviews the registered unit, and after the review is passed, the unit can log in the system for declaration. Otherwise, the unit must be re-registered for authentication. After the completion of registration, the social science administrator shall set up the reporting indicators according to the think tank evaluation system, and each unit shall make systematic reporting according to the reporting indicators. The manager of the unit shall conduct a preliminary examination of the materials declared by the unit. If the materials are not complete or the declaration is wrong, it shall be returned and declared again. After the unit administrator passes the examination, it shall be submitted to the administrator of social science for review. If the materials fail to pass the examination, the administrator of social science shall directly return the declaration to the applicant, who shall make a new declaration, and then proceed with the next operation after the declaration passes.

After the completion of the material declaration, the social science administrator will set the scoring index. The system will initially give a machine score according to the scoring indicators. The social science administrator will group all the experts and all the declared items into groups. The expert panel will select the items according to the random algorithm and manually review and score the items. The scoring results will be reviewed and checked by the social science administrator finally.

![Fig.1 The overall process picture of the think tank evaluation system](image)

3. System architecture

Under the architecture of micro services, the various services of the system are separated. The whole system is divided into: registration service, audit service, declaration service, unit management service, project management service, expert group service, expert rating service, authority service, log service, etc. This decoupling ensures that each service focuses on its area of responsibility and reduces coupling between services. Due to the small size of each service code, it is convenient to upgrade the service later. In addition, when the system is deployed for operation and maintenance, each micro service can be deployed to the server separately. When one service goes down, the normal use of other
services will not be affected. The architecture diagram of the new think tank evaluation system microservice is shown in Fig.2.

Fig.2 The overall framework of the think tank evaluation system

Under the micro service architecture, each small micro service is actually implemented by multiple small applications. And each small application has an independent front-end and server-side three-tier architecture, the server-side three-tier architecture is respectively: control layer, business layer, and data access layer. Thus, each service can manage its own data on its own, and different services can be developed using different technology stacks. In the architecture of micro service system, distributed governance and application gateway are adopted. The request first goes through the gateway, and the gateway forwards all the requests uniformly. As shown in Fig.3, the core of the Zuul Gateway is a servlet running inside Tomcat. Requests go to Zuul Servlet, and then the request is sent to zuulFilter Runner. ZuulFilter Runner manages all of its internal filters, which are divided into three levels. The system is internally divided into aggregate services and infrastructure services. Each service is first registered in the service registry and then forwarded by the gateway. The system can manage its own data flexibly and realize the communication between services, which improves the freedom of data communication of each small service. In the system, each user's access request is received through Nginx, and the load balancing algorithm is used to route and forward the data to the server with the lowest current workload. The server processes the data through a three-tier architecture, and finally returns the data to the front end in Json format, so that the user can see the required data on the front end interface.
4. **Score model**

The flow chart of the automatic grading model is as follows. The system will automatically identify the model according to id based on the contents of the declaration file. Where the basic requirements do not meet the declaration, direct no score. If attitude and discipline are involved, points will be deducted. The basic condition module is the fixed score. If there is a declaration, the score will be added, and the score will be added once at most. For the declaration within the range, set the maximum and minimum value within the range, and the range value can only be taken as integer. For example, the range value is 8-10 points, among which 10 points are certain, 9 points are approved, and 8 points are agreed. For the incremental score, the calculation rule is, for each increment, assign x points. The lowest score is 0 and the highest score is N. According to this automatic scoring model, each score value is calculated, and the total score value is summed up. The scoring flow chart is shown in Fig.4:

![Automated Scoring Model](image)

**Fig.4** Automatic scoring model

5. **System implementation**
The system uses Spring Cloud to build micro-service architecture, service governance uses Spring Cloud Eureka to achieve, and client load balancing is achieved with the Ribbon. The combination of the Ribbon and Eureka makes the system reliable and easy to expand. The system declarative REST call is implemented with Feign, which alleviates the risk of the system. The API gateway is implemented with Zuul, and the unified configuration management of the system is implemented with Spring Cloud Config. The underlying service architecture is developed using the Spring Boot framework. At the same time, in order to reduce the pressure of database access, NoSQL technology is adopted to put the data with more changes in the memory cache, so that the later request will have priority to read the data in the cache. Since the think tank score has four levels of indicators, the system uses a random algorithm to group all the projects, and each project needs multiple experts to score, and finally the average value is taken. The single-module scoring code of think is as follows:

```java
public BigDecimal computeAvg(String poid, List<XJSONObject> list_zb) throws Exception {
    BigDecimal score = new BigDecimal(0);
    for (XJSONObject zb : list_zb) {
        XJSONObject ascore = dataService.selectOne("SELECT SUM(ascore) AS ascore FROM zk_data WHERE poid=" + poid + "+ AND id IN (SELECT id FROM zk_zb WHERE _joincid LIKE "/" + zb.getString("id") + "/%")");
        if (ascore.get("ascore") != null) {
            score = score.add(ascore.getBigDecimal("ascore")).multiply(zb.getBigDecimal("mix")).multiply(new BigDecimal(0.01));
        }
    }
    dataService.updateData("UPDATE zk_unit SET ascore = " + score.floatValue() + " WHERE poid=" + poid + "/");
    return score;
}
```

6. Conclusion
Microservice architecture breaks down the traditional single system into several discrete services, so that each service can be developed and maintained independently. In this paper, a new think tank evaluation system is designed and implemented. From the perspective of business design, it meets the needs of local think tanks. The various micro services of the system realize different functions, the services can communicate with each other, and each service is deployed separately. The new think tank evaluation system greatly improves the user experience and the use efficiency of the system.

Acknowledgments
This work was supported by Development and Application research of Chongqing new think tank evaluation System 2019WT06

References
[1] Yuan Z, Yulei W, Geyong M, Chengqiang H and Ke P 2020 An Intelligent Anomaly Detection Scheme for Micro-Services Architectures with Temporal and Spatial Data Analysis IEEE Trans. Cogn. Commun. Netw. 6 pp548-561
[2] Kang H, Le M and Tao S 2016 Container and Microservice Driven Design for Cloud Infrastructure DevOps IEEE Int. Conf. Cloud Eng., IC2E pp202-211
[3] H. Sami and A. Mourad 2020 Dynamic On-Demand Fog Formation Offering On-the-Fly IoT Service Deployment IEEE Trans. Netw. Service Manag 17 pp1026-1039
[4] Ping W, Luobing D, Yueshen X, Wei L and Ningning J 2020 Clustering-Based Emotion Recognition Micro-Service Cloud Framework for Mobile Computing IEEE Access 8 pp49695-49704
[5] Luzheng B, Mingtao W, Cuie W and Yili L 2015 Development of a Driver Lateral Control Model by Integrating Neuromuscular Dynamics into the Queuing Network-Based Driver Model IEEE Trans. Intell. Transp. Syst. 16 pp2479-2486