The Research on Monitor Model of Middleware Based on Cloud Computing

Wenlong Feng and Mengxin Huang

ABSTRACT

According to multi-levels and concurrency features of cloud computing services, a monitor model based on middleware is proposed. The model analyzes cloud middleware architecture, types and functions under the cloud computing environment, defines load size and designs middleware load algorithms. It also proposes evaluation of middleware performance constructs the overall structure and function of middleware monitor model and ensures safe and stable operation of the middleware. Simulation results shows that the model can achieve real-time performance monitoring of middleware, and improve the reliability of cloud computing services.

INTRODUCTION

Most of the traditional services offered by a single service provider tend to focus on certain areas of the business and provide a single and limited function. All resource of traditional services required such as hardware design, data collection, data structure design, logic analysis, business combinations and display modes are independently provided by service providers, so there is a long development cycle, costly and higher technical threshold, and it is difficult to meet the complex requirements [1]. The cloud computing service possesses multi-levels and concurrency features. An integrated service often is composed of many middleware such as the data component, logical components and web services component.

Wenlong Feng, Mengxin Huang, Institute of Information, Hai Nan University, Haikou 570206, China
These middleware are provided and managed by different service providers, have heterogeneous and exotic features [2]. For a user, his concern is that a service can be reliably and normal operation. Therefore, service reliability is the core issue of cloud computing services. Middleware is the basis for cloud computing service reliability. Building middleware monitoring model, knowing changes in middleware performance in real time, guaranteeing reliable and safe operation of middleware is a great significance to improve the reliability of service.

THE SERVICE MIDDLEWARE UNDER CLOUD COMPUTING

The Middleware Features

Cloud computing service is a multi-level structure and composed of the client, middleware and database. Middleware are located between the client and the database, including web components, logical components and data components. It is the key to cloud computing services. Cloud computing service is shown in Figure 1.

WEB components directly faces the user, provides a platform for users to interact with the system in accordance with certain display mode. Generally it does not involve complex data processing, and the main function is to back the result of business components to the user [3] such as jsp, servlet, struts, etc. Business components with complex logic functions are a core component of the service. The development of business component follows certain norms and is cross-platform and portability. Common business components include EJB, SPRING. Data component such as HIBERNATE is to solve the traditional data connection, operation complexity and reduce the burden on developers.

Middleware performance evaluation parameters include response time, throughput and success rate. These parameters used by system are different because functions of middleware are different. For an on-line query system, what users are most concerned about is the query time. Therefore it often uses response time as middleware performance. For a large data processing system such as storage, it generally uses throughput as middleware performance [4].

The service middleware under cloud computing has the following characteristics:

1. Middleware has exotic features. Cloud computing services are distributed services, middleware services are often deployed on different nodes and run on different servers. For example, web server is installed in the node A, EJB server is installed in the node B, so it increases the difficulty of communication between middleware and reduces service reliability.

2. Middleware has different management body. Cloud computing services are composed of multiple services; these middleware often provided by different service providers, management and maintenance are also completed respectively. Because integrity of these service providers are different, middleware reliability provided by
provides is difference in the actual run-time. It makes difficult to service composition.

The Design of Middleware Load

The middleware load is related to the nature of the service in addition to the category of the middleware. For some simple query system, the amount of data involved is small, and it can be changed in a stable range, so the load can be expressed by query times. For some complex computing systems, the amount of computation and data are large, and the load can be expressed by throughput [5].

1. DEFINING MIDDLEWARE LOAD GRANULARITY

Middleware runs in the corresponding server and the server do management to the middleware such as activating, calling and releasing. Cloud computing service load granularity can be divided into two aspects: one is the load of server. It makes the server as a whole to describe the server busy state, such as thread pool usage, EJB object pool, database connection pool usage, etc. When the server load reaches a certain degree, the server processing ability will drops. The central manager carries on the resources to mobilize the work, and ensures that the server is in normal working condition. The advantage of using server level load design is to design simple, most of the servers have monitoring method, without the need to develop the programming process [6]. But the disadvantages are also obvious: because the server runs a lot of middleware, such as multiple EJB components, so it cannot determine in the end which application that its EJB component load is overload to causes the entire server overloaded, and it is difficult to determine the location of a specific middleware. The other is the middleware level load. It describes the middleware state from the middleware usage including the number of middleware calls and the number of middleware method calls, etc. [7]. This kind of load granularity is more detailed and more specific than server level granularity and it can accurately track, locate and judge the cause of service overload, so as to provide the basis for resource scheduling. Because most servers do not provide the monitoring function of the middleware, it requires developers programming through the interceptor to obtain the corresponding monitoring data. It increase development
cost. In this paper, it uses middleware level granularity and focus more on service reliability.

2. THE MIDDLEWARE LOAD ALGORITHM

In this paper, we use the two layer algorithm: the busy degree of middleware and the method of middleware. The busy degree of middleware reflects the call of a middleware in the middleware server, and the busy degree of middleware method reflects the invocation of a method in a middleware.

(1) The middleware busy degree algorithm

The number of middleware deployed in the server is N, and middleware A is one of them. During the study time, the number of calls to all middleware in the server is W, the number of calls to A is Z. The call probability of A is:

$$p_A = \frac{Z}{W}$$

(1)

The busy degree of middleware is also relation with the number of all middleware in the server in addition to the call probability. If there is no logical relationship between these middleware, the busy degree of A is:

$$N_A = p_A \times \frac{N-1}{N} \quad N>1$$

(2)

By the formula (2), when the number of middleware is certain, the higher the A calls, the more weight of A, and the influence of the busy degree of server is larger.

(2) The middleware method busy degree algorithm

The number of method of the middleware A is m, and Ai is the method i. During the study time, the number of calls to all methods in A is s, and the number of the method Ai is r. The call probability of the method i is:

$$p_i = \frac{r}{s}$$

(3)

The busy degree of the method Ai is:

$$m_i = p_i \times \frac{m-1}{m} \quad m>1$$

(4)

Considering the busy degree of middleware and middleware method, using phasor model, the load of the middleware can be expressed as:

$$Load_A = (N_A, m_1, m_2, m_3, ..., m_m)$$
The Design of Monitoring Model

The service middleware monitoring model has two functions: first of all, it monitors the middleware, timely reflects the load changes, and masters the real-time performance of the middleware. Secondly, when the middleware load is too heavy, causes its performance reduced, it is able to timely alarm, prompts the server to optimize the resources, and adjusts the relevant configuration. The monitoring model consists of four modules: data acquisition module, load algorithm module, performance evaluation module and alarm module.

The data acquisition module: it collects information of middleware such as middleware name, caller name, the number of middleware instance, the number of calls to middleware and middleware method.

The load algorithm module: according to information from the data acquisition module of the middleware, calls load algorithm and gets the specific middleware load.

The performance evaluation module: according to information of the specific middleware and load conditions, calculates middleware performance indicators such as response time and throughput.

The alarm module: When the performance of the middleware is significantly decreased and cannot meet the SLA requirements, the alarm module can make alarm [8].

The monitor model structure is shown in Figure 2:

![Figure 2. The monitor model structure.](image)

**SIMULATION RESULTS**

The middleware monitoring system proposed by this paper has been applied in the smart tourism platform based on cloud computing environment developed by Hainan University. The platform based on cloud computing technology; data components are from domestic tourism companies, and business components such as route customization, personalized recommendation are developed by the Hainan University. The monitoring system is simulated on the platform, and the simulation is carried out to verify the monitoring function of the monitoring system. Simulation results are as follows:
1. According to the user access to the platform, the middleware load can be calculated in time.
2. The performance indicators such as response time and throughput can be monitored in real time.
3. The alarm module function is normal, can send out alarm in time.

CONCLUSION

The middleware monitoring model overcomes the shortcomings of the traditional monitoring system that monitoring granularity is too large. It considers the characteristics of multi levels in the cloud computing environment, and focus on middleware load and performance monitoring. The model introduces the two layer load granularity concept, can accurately calculate the load of middleware, the load of method, and link the middleware load to the performance index. The model makes the relationship between the alarm module and resource scheduling module by using the threshold and anti-jamming strategy, and provides basis for application optimization.

ACKNOWLEDGEMENTS

This work was financially supported by the Foundation projects: the natural science foundation of Hainan Province (project number: 614232) and the industry-university-research special fund of Hainan Province (project number: CXY20140002).

REFERENCES