A Discovery Method of Service Bottleneck for Distributed Service

Jie Wang[1], Tao Li[1,2(✉)], Hao Wang[1], and Tong Zhang[1]

[1] School of Computer Science and Technology,
Wuhan University of Science and Technology, Wuhan 430065, Hubei, China
{909901326,1593487967,zt1996816}@qq.com,
litaowust@163.com

[2] Hubei Province Key Laboratory of Intelligent Information Processing
and Real-Time Industrial System, Wuhan 430065, Hubei, China

Abstract. In order to deal with the large scale access to billions of users currently, large internet companies have adopted the distributed service of parallel processing to support it. The uncertainty of user behavior and software multi-tier architecture led to the distributed service behavior of uncertainty and complex dynamic combination between services, so it is difficult to detect the service bottleneck of distributed service. In this paper, we propose a service bottleneck discovery model of distributed service which is based on the two layer structure: the analysis based on the behavior attribute of the service and the relationship between the services.

Keywords: Distributed services · Bottlenecks · Relationship · Behavior attribute

1 Introduction

Big data, Mobile Internet, Mobile client application and Internet applications have gathered a huge amount of users. In order to timely response to the large crowd access and request, as well as provide efficient service quality, large Internet companies are adopting the distributed service system of concurrent processing as a solution.

The random behavior of massive users leads to the uncertainty of service behavior. This has brought a huge challenge to the whole service system. It can't find deep level issues by the direct influence factors such as hard disk resource, memory capacity, CPU utilization, service time, call number, network bandwidth, I/O and so on. The paper [1] introduced several methods of bottleneck discovery: the use of continuous time Markov chain to analyze the availability of resources, and through the average rate of service arrival and service rate of services to analyze the possible bottlenecks in workflow; analyze the behavior characteristics of multi level Internet application, propose a analysis model based on closed alignment network, each queue in the model corresponds to the different levels of the application, according to captured the behavior attribute of the model, then it can be used to predict the performance of the Internet and bottleneck positioning.

S. Guo et al. (Eds.): CollaborateCom 2015, LNICST 163, pp. 309–316, 2016.
DOI: 10.1007/978-3-319-28910-6_29

For different applications of distributed servers, of course, the factors that affect the quality of service are different. Some software can't simulate the real behavior of mass users by stressing testing, no fine grained analysis and deep excavation so that it can't predict and implement feedback in advance. The current research is based on load balancing to solve the problem of distributed server bottleneck, do a lot of redundant work in advance. In this paper, a new model for bottleneck discovery of distributed service is proposed. The model is designed for the two layer architecture: analysis of dynamic service behavior attributes and based on service dependencies, the validity of the model is verified in experiments and it has a certain value for the bottleneck discovery.

2 Theoretical Model and Analysis

At present, there is not a clear definition of the service bottleneck for distributed service. In order to make experiment and theory, we make a definition of service bottleneck: Under normal condition, the effect of service quality is caused by the mass increase of the service behavior, the dynamic combination of service behavior and the transfer of large-scale in short time. In this paper, a new model for service discovery of distributed service is proposed. The model is designed for the two layer architecture: analysis of dynamic service behavior attributes and based on service dependencies. When the two methods of analysis results are consistent, we believe the result is valid, that is this service is most likely to affect the quality of service in the entire service implementation process. This service has led to a large increase in time for the user-oriented services, which makes the resource utilization rate of the background server increased dramatically, even the service will run out.

2.1 Analysis of Dynamic Service Behavior Attributes

In the analysis of dynamic service behavior attributes, we are mainly to find a service bottleneck possible based on the correlation between data and the degree of data fitting in the verification of the uncertainty of distributed behavior. In the paper [2], a detailed description of the behavior of distributed services is that the dynamic behavior of the service is described by the **9** tuple of a service origin log. Those are Token, Invoking Service, Service Invoked, Location, Elapsed Time, Times Tamp, Input, Output and Status. In this paper, we use the method of service behavior collection in document [2]. The **9** tuple are the input data of the experimental model. The main work is recording statistics the counts of service-invoking, elapsed time of service- executing, invoked service and service invoked. Then use mathematical functions to explain the uncertainty of service behavior by optimization and processing of elapsed time and counts. In this paper, we first construct an uncertainty verification model of distributed service behavior to explain the real existence of bottlenecks. Then find the bottleneck or bottleneck area and lay a solid foundation for the follow-up to solve the bottleneck problem. Figure 1 as a model for uncertainty verification:

This experiment is based on the number of service calls and service time consuming statistical processing. In order to be closer to the actual operation of the distributed service, experiments are divided into two steps to verify the uncertainty

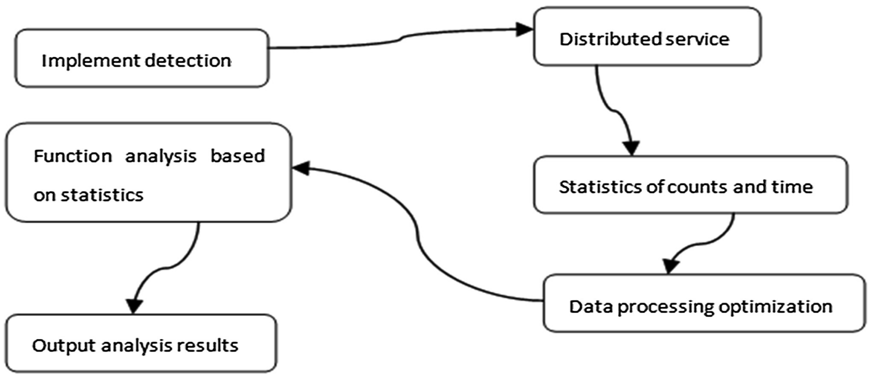
Implement detection
Distributed service
Function analysis based on statistics
Statistics of counts and time
Data processing optimization
Output analysis results

Fig. 1. Behavior uncertainty verification model of distributed services

behavior of distributed service: number of static service calls and dynamic random allocation service counts. Different results for different data processing, we introduce a concept of relative distance between data for these two variables of counts and time [3]. We deal with the data between different units by the "**relative distance**". The last we can use the **N** function to characterize relationship of counts and elapsed time by uniform amplification or reduction.

We illustrate the behavior of the distributed service by two results: correlation and function fitting degree between counts and time. Process two different units data, then analyses the results of several experimental data, we get a conclusion the correlation of their data **T** is close to **1**, the higher the correlation is. The **+1** indicates a complete positive correlation, the **–1** said total negative correlation. The experiment found the correlation between the two data is not consistent, there are complete positive correlation and complete negative correlation in two cases, so the behavior of the service is dynamic and its behavior is not determined. The fitting degree of function is developed from the data dependence, two data must be guaranteed to be relevant, the fitting function is effective. When the fitting function **R** was close to **1**, the representation of the two data can be replaced by the function expressions and the effect is remarkable. When the fitting function level was close to **0**, characterization of the two data can't be expressed by function.

The main purpose of this experiment is to analyze the service dynamic behavior attribute, then to find the service bottleneck of distributed services. In the experiment, we used the data correlation and the function fitting degree to analyze and discovery service bottleneck points. When a service is invoked by a large of users, the behavior are completely contrary to normal state or has special behavior, we think the service is a service bottleneck. The following is the algorithm process of dynamic service behavior property analysis.

2.2 Analysis of Dynamic Service Dependency

Define a series of cloud services as **S = {S1, S2, …, Sn}**, invoking between services such as **S1** invoke **S2**, remember as **S1->S2**, make **S1->S2** intitule path or service dependency. But it is not a complete path only as a part of the path, the full path to the implementation of the service is that **Sj** is no longer dependent on other services. We sort of total counts and total time for complete paths, find maximum complete path **L1** of total counts and longest complete path **L2** of total time. When **L1** and **L2** are consistent, we believe that the bottleneck is in this path. That is to say that the maximum complete path of total counts and longest complete path of total time are likely to be the **"bottleneck path"**. The behavior of distributed service is not deterministic, so we know when services execute at different times, different number of user access and different resource allocation schemes etc., the number of **"bottlenecks path"** is dynamically changing. When the **"bottleneck path"** is more and connected to each other, we think that the service bottleneck of distributed service constitutes a **"bottleneck area"**. The **"bottleneck area"** is also a way to find the service bottleneck and the solution to the service bottleneck. In this experiment, because of the specific service relationship and service's quantity is not very many, yet not use of **"bottleneck area"** to discover and solve problems.

After analyzing the **"bottleneck path"**, counts and time of each service in this path are analyzed. We make a sort of counts and time respectively by the proportion of sum time and sum counts for various services, then find the maximum value of the two sort, the service is likely to be a **"bottleneck service"**. Make data analysis combine with the first step experiment, we can find out which service is the bottleneck by comparing the correlation of data and the degree of function fitting. The following is the algorithm flow of service dependency:

3 Experimental Results and Analysis

3.1 Experimental Data Sources

In this experiment, the research object is login and elective service of students. Figure 2 shows the service flow. In order to simulate the real environment of student selection services, the Eclipse generated the real data scaling of a certain multiple by random to achieve the effect of the real environment.

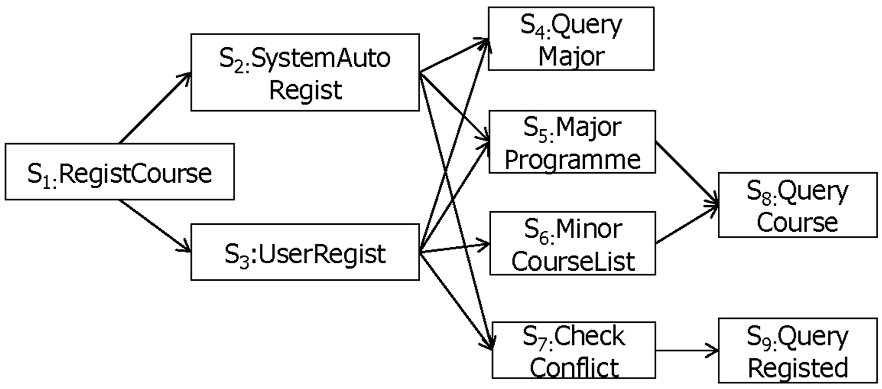

Fig. 2. The service flow chart of login and elective in the course for students

3.2 Experimental Procedures and Results Analysis

These nine nodes were recorded as services **S1–S9**, we take the service **S1** for example. Take six different invoking counts (10, 15, 20, 25, 30, 35 Unit: ten thousand). It is difficult to find the relationship between counts and time when data volume is relatively small in the experiment and the error is relatively large.

3.2.1 Behavior Attribute Analysis Based on Dynamic Service Behavior

A Counts of Static Service Invoked. Figure 3 is the data correlation between count and time of individual services. Figure 4 is function fitting degree, their horizontal coordinates are **S1–S9** these nine services, the longitudinal coordinates are the data correlation degree and the function fitting degree.

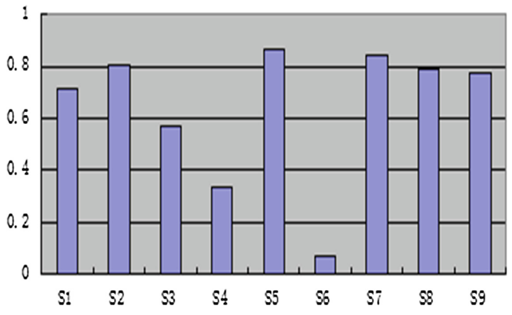

Fig. 3. The data correlation between count and time

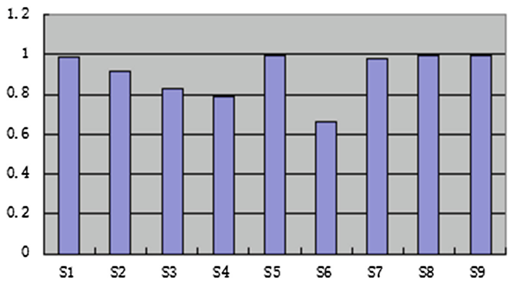

Fig. 4. Function fitting degree

B Dynamic Random Allocation Service Invoked Count. Figure 5 is the data correlation between count and time of individual services. Figure 6 is function fitting degree. Their horizontal coordinates are **S1–S9** these nine services, the longitudinal coordinates are the data correlation degree and the function fitting degree.

C Experimental Summary. In comparison to the experimental data chart, we find that the transformation of **S1–S9** is obviously different in both static and dynamic two

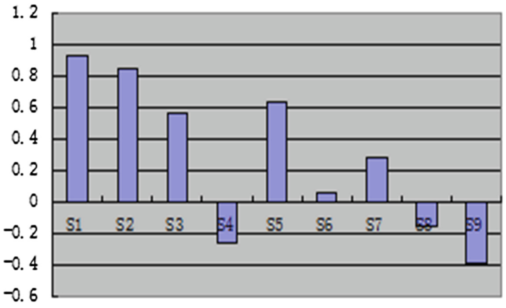

Fig. 5. The data correlation

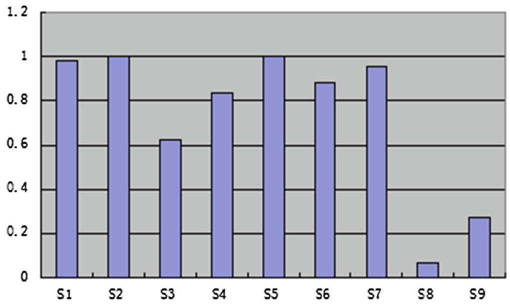

Fig. 6. Function fitting degree

different cases, the count of the invoked service in particular. In static service invocation experiments, we find counts and time to be positive correlation, but in the dynamic service invocation experiment, count and time of service **S4, S8, S9** to negative correlation. We think that as the most basic service in large-scale dynamic call situation is very likely to appear unstable situation, that becomes the bottleneck of the cloud services. In function fitting process, we do it by polynomial regression analysis, when the order number is four, function fitting is very well, under static and dynamic two cases, there is a significant difference of consuming function between the number of individual service. In summary, we think the service behavior of distributed service is dynamically changed and behavior is uncertain in different circumstances, so we must do a good job on the distributed service bottleneck detection, and research solutions in real life.

3.2.2 Analysis of Search Ranking Based on Path

The complete path of this experiment are **S1->S2->S4, S1->S2->S5->S9, S1->S2->S7->S8, S1->S3->S4, S1->S3->S5->S9, S1->S2->S6->S9, S1->S3->S7->S8**. Figure 7 is total count for each complete path chart, Fig. 8 is total time for each complete path chart.

From the chart, we can see when service S1 is invoked, the full path of the highest for the total count invoked is **L1 = S1->S3->S5->S9**, the full path of the longest for the

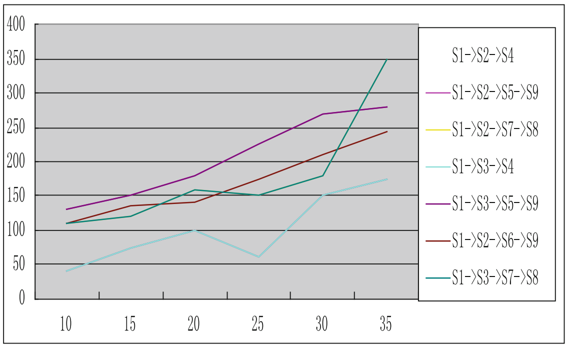

Fig. 7. Total count for each complete path

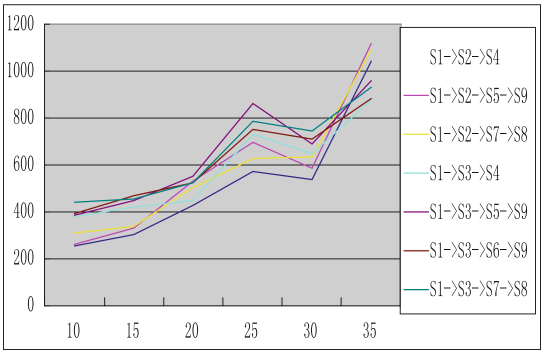

Fig. 8. Total time for each complete path

total time is **L2 = S1->S3->S5->S9**. That is **L1 = L2**, that is also the path **L1** and **L2** overlap. This also validates our hypothesis: if the maximum path and the longest path are the same path in service execution, we think this path is the **"bottleneck path"** which we are finding, the corresponding service bottleneck points are also in this path. After the contrast of the data again, we found out the maximum count of services is the service **S9** and the longest time of services is the service **S1** at the service **S1** transferring. We believe that service bottlenecks are likely to occur in service **S1** and service **S9** when users are dynamically invoked this service cosmically in the implementation of distributed services. At the same time, we find that service **"bottleneck path"** more than one when the amount of data is growing, but they are always related with the most basic and the bottom of the service. When the service bottleneck occurs, it is likely to be transferred from the dependency relationship between services, that is the **"bottleneck transmit effect"**.

4 Conclusion

Service bottleneck of distributed service is a problem of uncertainty, we can predict the possible emergence of a service bottleneck by using a graph based path search method. Improve the detection efficiency of the bottleneck by the invoking relationship between the services and make a relatively reasonable definition of "bottleneck problem". There are still some problems in the experiment: the service dependent relationship is not enough complexity and the test data is not enough, the method still needs to be strengthened. We will meet the requirements of the distributed service bottleneck detection efficiency in future research and study.

Acknowledgement. Jie Wang, Tao Li, Tong Zhang and Hao Wang are partially supported by the National Natural Science Foundation of China (Grant Nos. 61273225). Humanities and Social Sciences Foundation of Education Ministry of Hubei province (Grant Nos.2012D111).

References

1. He, Y., Shen, H.: A web service composition performance bottleneck locating strategy based on stochastic Petri net. J. Comput. Sci. (2013)
2. Li, T., Liu, L., Zhang, X., Xu, K., Yang, C.: ProvenanceLens: service provenance management in the cloun. In: Proceedings of 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2014), pp. 3–5 (2014)
3. Hu, Z., Liu, X.: A practical data fusion algorithm. Autom. Instr. **2**(1), 1–3 (2005)
4. Zhang, X.: Java Program Design and Development, pp. 1–245. Electronics Industry Press, Beijing (2010)
5. Wu, W., Yan, Y.: Data Structure (C language), pp. 1–230. Tsinghua University Press, Beijing (2011)
6. Yang, H.: Analysis of large scale network traffic bottlenecks. National Defense Science and Technology University, pp. 43–48 (2007)
7. Department of mathematics, Tongji University. Advanced Mathematics. Higher Education Press (1996)