Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2021.020407
ISSN 2790-0916 Vol. 2, Issue 4: 52-60

Multilayer Distributed System Software Architecture Based on Aspect Service and Web Service

Varuun Vermane*

Jimma Institute of Technology, Ethiopia

**corresponding author*

Keywords: Aspect Service, Web Service, Distributed System, Software Architecture

Abstract: With the development of network and computer technology, Web services and service-oriented architecture are emerging as technologies and architectures of distributed systems. Web service technology can integrate various loosely coupled applications, so that different operating systems, different programming languages, and different application platforms can be seamlessly connected and interoperable. The purpose of this paper is to introduce aspect-oriented thinking into Web services based on a multi-layer distributed system software architecture based on aspect services and web services. Aspect-oriented programming provides a mechanism to explicitly capture and modularize cross-cutting concerns, and uses AOP concepts to address cross-cutting concerns in web services, thereby improving the adaptability and reusability of web services. A multi-layer software application architecture based on AOP and Web services is proposed to make embedded systems easier to understand, maintain and expand. The test results show that the bandwidth is 1Gbps, and the peak throughput is 0.069MB/s, indicating that the access request can be well shunted and limited. The device upload record adopts an asynchronous method, simulating 2000 devices uploading records at the same time, and the average response time is 0.5 seconds.

1. Introduction

In the field of distributed applications, most of the traditional distributed application architectures start from their own needs and use a variety of different technologies to develop independent closed systems, which are incompatible and effectively compatible with each other. This makes the system difficult to expand, hindering its further development and application [1]. As the best distributed computing model at present, Web services have become the main strategy to solve many development problems. Facing the ever-changing Internet environment, the evolution of Web services is a major challenge in the life of Web services. In the system framework, it is a very

important goal to solve the evolution of Web services to ensure the correct operation of services and the constantly updated service requirements [2].

At present, the application research of multi-layer distributed system software architecture at home and abroad starts from many aspects. Bahri L explores the existing literature on privacy in web services during transactions, and in this paper conducts a comprehensive survey of the most relevant published proposals. We identified 20 works that address privacy-related issues in the consumption of web services. We classify them according to the methods they employ and compare them according to a proposed evaluation framework derived from the techniques employed and the requirements fulfilled [3]. Kovalets IV uses a web client-web server-cloud computing architecture, where the computation of the model is performed in the cloud infrastructure, while the client and server parts run on different computers. Through the developed service, it is possible to predict air pollution at every point in Ukraine, including more than 30 substances, including chlorine, ammonia, hydrogen sulfide, etc. Forecasts are made using data from the WRF-Ukrainian Numerical Weather Prediction System and visualized through a web interface [4]. Swetha NG proposes a new MCDM method, called Improved RIM (I-RIM), to overcome the existing flaws in RIM. This paper also proposes a novel framework that combines the power of graphics processing units (GPUs) and I-RIMs to improve the efficiency of the selection process. When parallelized using GPUs, the proposed I-RIM was found to outperform the parallelized MCDM technique used for research. The results also mean that I-RIM is more consistent and stable to the ranking process. It is also evident that the proposed framework incorporating I-RIM outperforms RIM in terms of execution time, mean reciprocal rank and Spearman correlation coefficient, which makes the framework more stable and reliable, making it suitable for real-time web service selection [5]. The existing research points out the direction for the application of the multi-layer distributed system software architecture based on aspect service and web service.

This paper explores the potential of AOP in dealing with cross-cutting concerns in Web services, and effectively separates the core concerns in Web services—normal Web services and cross-cutting concerns—aspect services. Since the weaving process occurs in SOAP messages Layers, Web Services and Aspect Services are not really woven, it facilitates the updating and reuse of Web services and the maintenance and extension of programs made from it. The distributed design model has become a widely used application model due to its scalability, reconfigurability, flexibility and reliability, as well as the concept of infrastructure and software integration. Based on the combination of the simplified J2EE multi-layer architecture and SOA, this paper studies the multi-layer software application architecture that introduces aspect services. The combination of the two will definitely bring forward new ideas for the development of our modern application system.

2. Research on Multilayer Distributed System

2.1. Core Technologies of Web Services

(1) Data representation - XML

XML is an extensible markup language that provides a way to mark up content and add information about the purpose of the data. Like HTML, XML is derived from Standard Generalized Markup Language (SGML), which is the common language for all markup languages. SGML is a wireframe language, also known as a system specification for markup languages such as HTML. XML is also a wireframe language, a subset of SGML that defines web applications. Like SGML, you can also use XML to define a variety of markup languages to meet different needs, especially in terms of data representation, users can create new tags as needed, and the extensibility of XML is

here [6-7].

(2) Messaging - SOAP

At the heart of Web services, SOAP is a simple XML-based protocol for exchanging information in a shared environment. It provides a simple and lightweight method for the peer-to-peer exchange of structured and typed information using XML in a shared-rest environment. The SOAP message format is based on the XML standard, so SOAP can be used to communicate between different computer companies, different technology platforms, different languages, and different operating systems. Web services use SOAP as the standard communication protocol [8-9].

(3) Description - WSDL

Web Services Description Language (WSDL) is an XML implementation that defines a Web service description as a set of service access points through which clients can access information or request access to services. It depends on the work [10-11]. WSDL first describes the access service and the request/response messages used during the access, then associates them with a specific transport protocol and message format, and finally defines the specific access point that executes the service. Concrete activation service access points are aggregated into abstract Web services [12-13].

2.2. AOP Technology

AOP can be said to be an extension and improvement of OOP (Object Oriented Programming). According to the different functions of each module in the system, AOP divides the software system into two parts: core concerns and cross-cutting concerns. The main concern is the modules related to business logic in the program, and the other parts are cross-cutting concerns. Such as authorization verification, logging, transaction processing [14-15].

2.3. Distributed Web Application System

A distributed web application refers to a distributed computer created around a computer network (Internet/Intranet), which has a browser and a web server, using a standard web browser instead of a local client such as a client. Application is a computer application program developed rapidly with the development of Internet/Internet [16-17]. Whether it is a computer application system within government departments and related enterprises, or an application system that provides services through the Internet, Web-based distributed application systems play an increasingly important role and have become the mainstay of computer applications. By using distributed application systems on the Internet, users can access and use computer applications as long as they have a browser (browser), not only on the user's computer [18].

3. Design and Research of Multilayer Distributed System Based on Aspect Service and Web Service

3.1. Overall System Architecture

According to the composition level of the system and the detailed division and function of each level. The general architecture of the system will be given below. The architecture and operation process of the entire application system are shown in Figure 1.

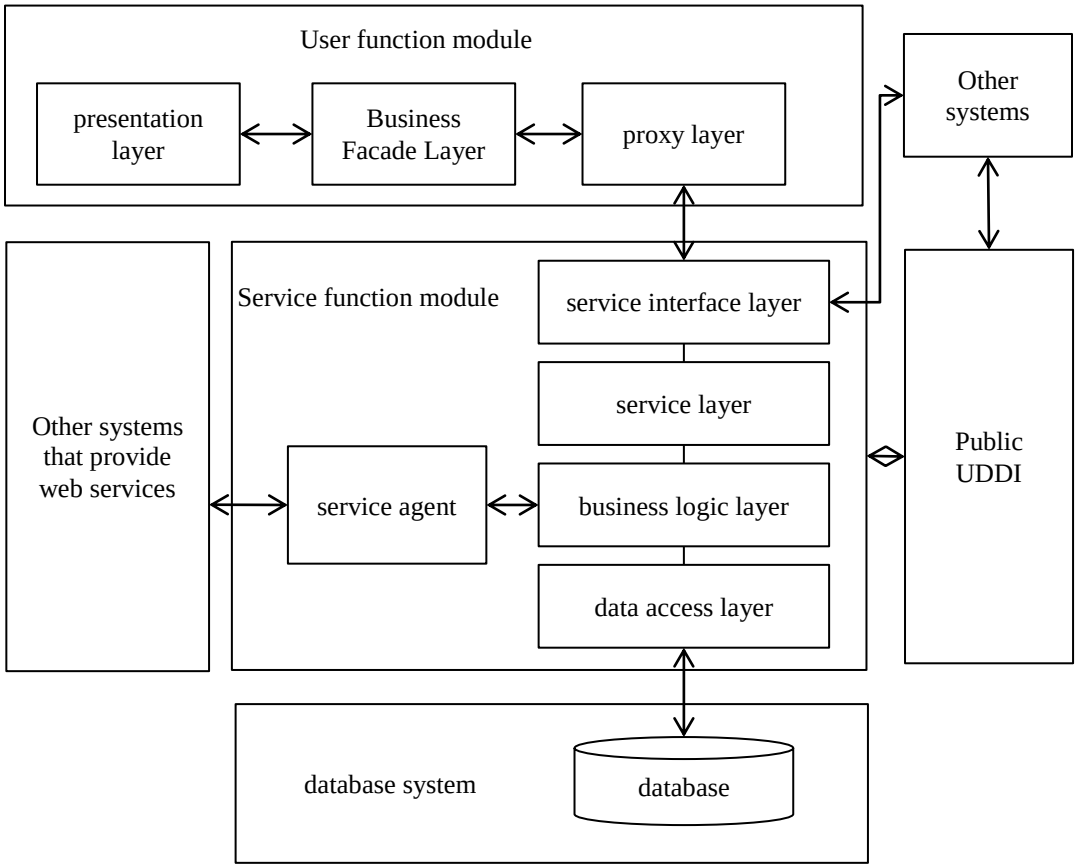

Figure 1. Overall architecture diagram of distributed system

It can be seen from the figure that the workflow of the entire application system is: the service proxy layer in the user module creates an instance of the Web service proxy class defined in the WSDL document, and calls the methods in the proxy class. WSDL document descriptions can be obtained from service modules or UDDI requests. NET platform serializes the parameters of the web page application into SOAP messages, sends them to the service interface of the service module through the network, and resumes the transmission through ISAPI in ASP.NET. The ISAPI extension accepts SOAP messages and outputs XML. Create an instance of the class implementing the web service and call the web service method of the service layer, passing the parsed XML as a parameter. Implement methods in web services and call business logic layer methods. The online system invites data providers to access data levels and process transactions. If the output and output parameters are set to return, they will be returned in XML. The business appearance layer of the user function module receives the result set and output parameters through the proxy class, and calls the display control of the presentation layer to display the output.

3.2. Advantages of Architecture

(1) From the client's point of view

Multiple clients and even external systems (other systems or web services) can access business intelligence through referenced standard web service interfaces.

(2) From the perspective of system development

The business intelligence of the system can be achieved entirely through services or service packages provided by third parties. In extreme cases, application developers only need to develop a client interface, and then remotely connect to Web business logic services such as standard modules provided by service providers to quickly build a business application system. Professional service providers are committed to developing business logic service components (common Web services and component services) to provide business logic services for application developers. Different app developers can combine these strategies to start their own businesses.

3.3. Design of Fitness Function and Processing of Fitness

This paper designs a dynamic load balancing algorithm for the server. Therefore, the balancing algorithm should make calculations based on the actual load of each server when allocating requests. For different applications, the importance of each factor is different, so the form of weighting can be used to highlight the important factors. Expressed as:

$$curLoad_i = k_1 * p_{cpu}\% + k_2 * p_{memory}\% + k_3 * p_{io}\% + k_4 * p_{cpuQueue}\% \quad (1)$$

where curLoadi represents the current load of the i-th server, and k1 represents the weight corresponding to the CPU utilization.

In the case of allocating requests according to a chromosome, the load size of each server is calculated, and then the average load of all servers is calculated, and the deviation of each server relative to the average load is further calculated. The smaller the deviation, the more even the distribution, the larger the deviation, the uneven distribution, which is not a good distribution scheme. So we get the following fitness function:

$$Fitness = 1 - \frac{\sqrt{\sum_{i=1}^{N}(load_i - \frac{\sum_{i=1}^{N} load_i}{N})^2}}{\sum_{i=1}^{N} load_i} \quad (2)$$

The principle of this formula is to first subtract the average load from the load of each server to obtain the load deviation, and then calculate the load deviation rate. The smaller the load deviation rate, the more balanced the load distribution and the better the performance. Therefore, the expression obtained by subtracting the load deviation rate from 1 can be used as a fitness function. Therefore, the larger its value, the smaller the load deviation rate and the better the performance.

4. Analysis and Research of Multilayer Distributed System Based on Aspect Service and Web Service

4.1. Throughput Analysis

The throughput of the entire system increases rapidly at the beginning as the number of users increases, and then gradually becomes stable, indicating that the caching scheme of the system plays a very good role, because data access follows the law of twenty-eight, and a large number of requests are generally concentrated in a small number of parts. In terms of data, caching can speed up subsequent repeated accesses to improve system performance. The broadband and throughput

results are shown in Table 1.

Table 1. Broadband and throughput results

Time	Broadband (Gbps)	Throughput (MB/s)
00:00	1	0
02:00	1	0.012
04:00	1	0.069
06:00	1	0
08:00	1	0.03

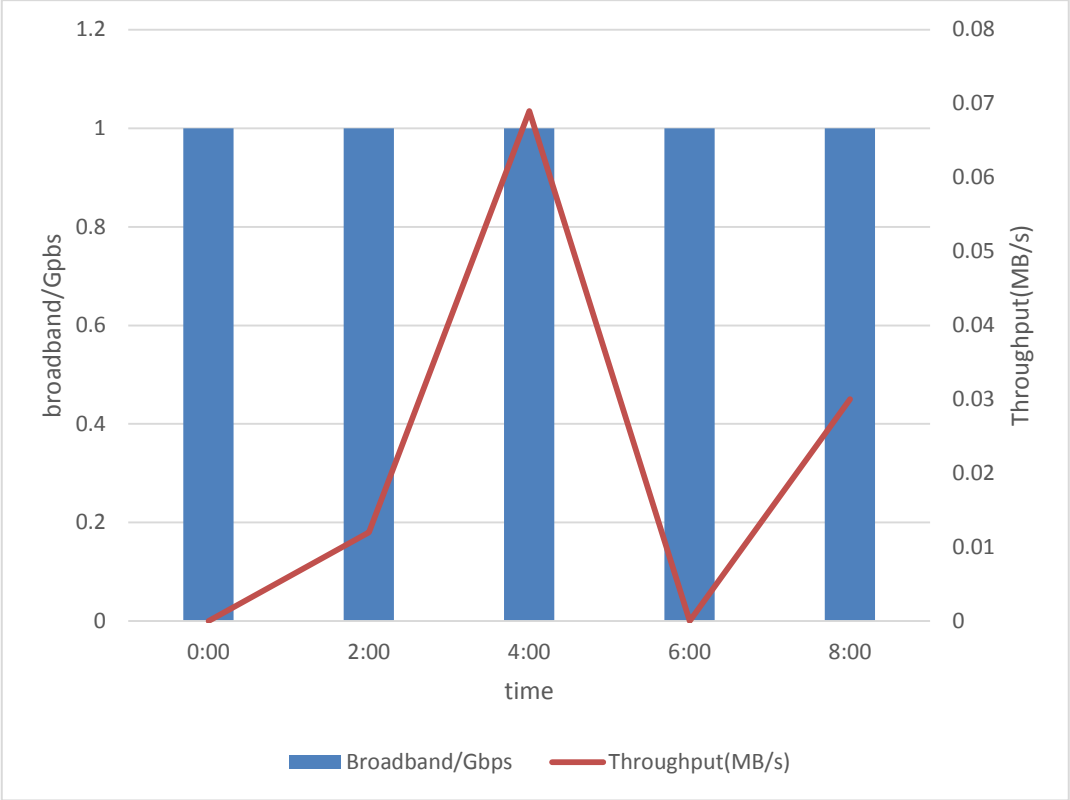

Figure 2. Broadband vs. throughput results comparison

As shown in Figure 2, the bandwidth is 1Gbps, and the peak throughput is 0.069MB/s, which is far from reaching the limit of the network, indicating that the load balancing scheme plays an obvious role, and it is a good way to limit access requests. Therefore, the system does not have to worry about network-level problems.

4.2. Analysis of Test Results

The results of the average transaction response time and the number of users are shown in Table

2.

Table 2. Transaction average response time and number of users results

Time	Amount of users	Response time
00:00	2	0.05
02:00	4	0.1
04:00	6	1.95
06:00	8	0.7
08:00	10	1.5

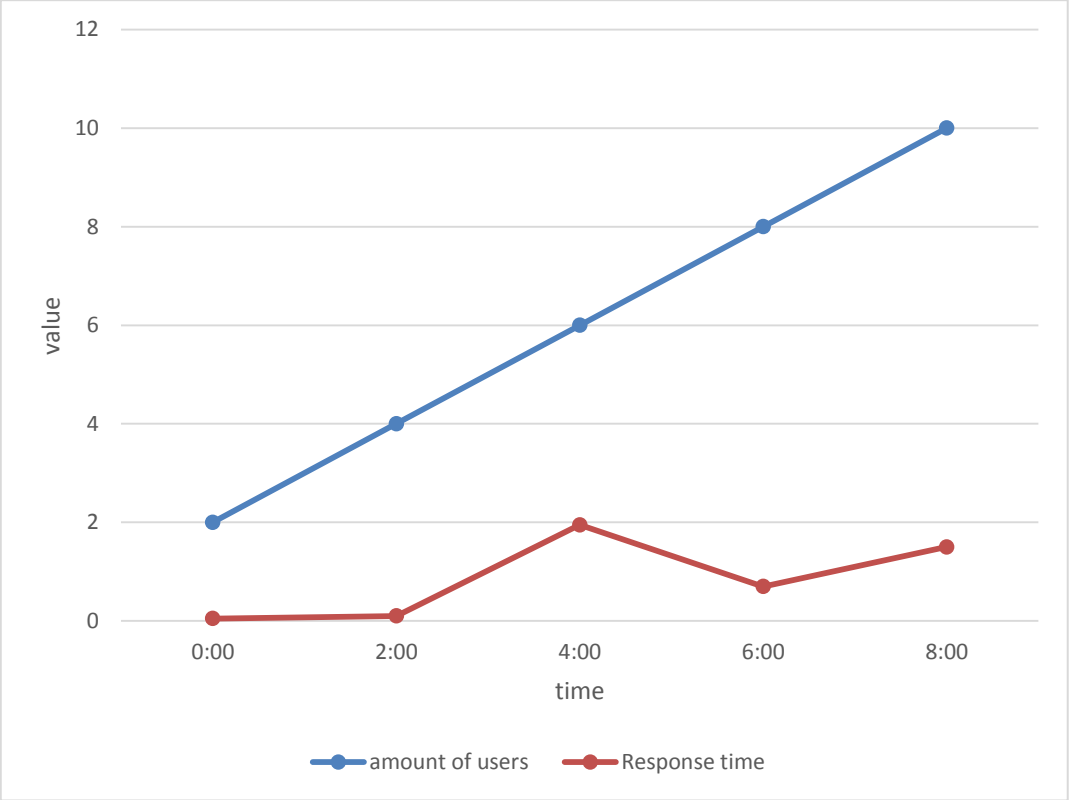

Figure 3. The relationship between the average transaction response time and the number of users

As shown in Figure 3, when 7,500 people access concurrently, the system will not crash, but the concurrency of 7,500 people will cause the following situation: about 45% of the query operations will have errors, or the expected results cannot be returned within the specified time. Some users may wait 1.5 hours with no results. If the concurrency is less than 7500, the response time of most users is shown in Figure 3: the graph can only reflect the experience of most users, and individual users may still have access errors. The ratio of error users will increase with the increase of the number of concurrent users, indicating that the system reaches the performance bottleneck when 7500 concurrent users access. The device upload record adopts the asynchronous mode, simulating 2000 devices uploading records at the same time, the average response time is 0.5 seconds, the asynchronous mode can greatly improve the interaction performance between the device and the

system.

5.Conclusion

This paper is based on the multi-layer application architecture of aspect service and web service. After the conceptual design of aspect service, the multi-layer application architecture based on aspect service and web service combines SOA and flexibility J2EE multi-layer architecture is simple. It improves the degree of software reuse., which is convenient for parallel software development, reduces development time and complexity, makes the application system easier to understand, and has good scalability and stability. After the introduction of the Facet service, the Web service call process model After the introduction of the Facet service, the service provider provides server-side services through four objects: Aspect Service, Web Service, Aspect Service Document, and Weaver. Based on the research and analysis of the traditional web service invocation model, a web service invocation process model with the introduction of aspect service is proposed.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Polska O V, Kudermetov R K, Shkarupylo V V. An Approach Web Service Selection By Quality Criteria Based On Sensitivity Analysis Of Mcdm Methods. Radio Electronics Computer Science Control, 2021(2):133-143.

[2] Tedyyana A, Fauzi M, Ratnawati F. Revamp Keamanan Web Service Milik PT XYZ Menggunakan REST API. Digital Zone Jurnal Teknologi Informasi dan Komunikasi, 2021, 12(1):1-10.

[3] Bahri L, Carminati B, Ferrari E. Privacy in Web Service Transactions: A Tale of More than a Decade of Work. IEEE Transactions on Services Computing, 2018, 11(2):448-465.

[4] Kovalets I V, Maistrenko S Y, Khalchenkov A, et al. Adaptation of the Web-Service of Air Pollution Forecasting for Operation within Cloud Computing Platform of the Ukrainian National Grid Infrastructure. Science and Innovation, 2021, 17(1):78-88.

[5] Swetha N G, Karpagam G R. GPU enabled Improved Reference Ideal Method (I-RIM) for Web Service Selection. International Journal of Information Technology & Decision Making, 2021, 21(03):855-884.

[6] Feddaoui I, Felhi F, Algarni F, et al. QoS-Based Collaborative Filtering for Web Service Mining. International Journal of Web Portals, 2021, 13(1):40-61.

[7] Meshcheryakova A A. Distance Learning In Informatics Lessons Using The Online Test Pad

Web Service. Informatics in School, 2020, 1(7):47-56.

[8] Lakshman A. Web Service Recommendation Method of Hybrid Item-Memory based Collaborative Filtering for Scalability of Data. Journal of Advanced Research in Dynamical and Control Systems, 2020, 12(7):714-720.

[9] Atamuratov R K. The Effectiveness Of Timeline Interactive Web Service In Delivering Lessons On "History Of Personal Computers". Theoretical & Applied Science, 2020, 92(12):109-111.

[10] Aguilar O, Lasso E, Corrales J C. Web Service Adaptation for the Customization of Early Alerts in Agriculture. Revista Ingenier ás Universidad de Medell ń, 2020, 19(37):2248-4094.

[11] Susanto A, Inggi R. Implementation of Web Service for Integrated Data Students on Information System Academic and Library. International Journal of Scientific Research in Science and Technology, 2020, 5(1):623-627.

[12] Ghani I, Wan W K, Mustafa A. Comparative Evaluation of State-Of-The-Art Approaches in Web Service Testing. Test Engineering and Management, 2020, 82(Janurary-Feburary):12010-12017.

[13] Yaqin M A, Riza A, Anshori M, et al. Pengukuran Metrik Kompleksitas Web Service Sekolah. Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika), 2020, 5(1):147-153.

[14] Bader W I. Shared Web Service Architecture for Websites with Related Mobile Applications. International Journal of Computer Applications, 2020, 175(16):27-35.

[15] Cyganova A, Senotova S. Mathcad Remote Calculation Web Service. Scientific Papers Collection of the Angarsk State Technical University, 2020, 2020(1):43-46.

[16] Yaqin M A, Bachriwindi A, Zaman M, et al. Pengembangan Web Service Sistem Informasi Boarding School. Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika), 2020, 5(1):36-49.

[17] Aruna S. A Web service recommendation system based on QoS attribute based collaborative filtering. Webology, 2020, 17(1):1-9.

[18] Silva A, Ma H, Mei Y, et al. A Survey of Evolutionary Computation for Web Service Composition: A Technical Perspective. IEEE Transactions on Emerging Topics in Computational Intelligence, 2020, PP(99):1-17.