

# *Development Method and Support Tools of Distributed System Based on Web Service*

**Hashash Mostafa\***

*Siksha O Anusandhan Deemed Univ, Inst Tech Educ & Res, Dept Elect & Elect Engr,  
Bhubaneswar 751030, Odisha, India*

*\*corresponding author*

**Keywords:** Web Service Application, Distributed System, Development Method, Support Tool

**Abstract:** With the advent of the Internet era, the system design concept continues to promote the development of software technology, and the distributed system development technology has made a qualitative leap, which improves the productivity of developers and provides convenience for producers. The purpose of this paper is to research distributed system development methods and supporting tools based on Web Services. The sub-process of Web Service component development and the deployment of Aspire Extension platform are analyzed. The supporting tools required by the distributed system of Web Service application development are presented. Taking the Haitao distributed system as an example, the implementation of the function design of the extension platform and each development process is verified, and through the performance analysis of whether the Haitao extension instance developed based on the extension platform is developed, the Haitao in two modes is analyzed. The difference between the browser and the browser memory of the extension is the starting point, and the performance improvement of the extension application by the extension development platform is verified and expounded.

## **1. Introduction**

The emergence of the distributed system based on Web Service can solve the high maintenance cost of the client/server system, especially when the client is in different regions, because it allows different users to obtain the same user experience. There are differences due to different intellectual property rights [1-2]. Furthermore, it can increase the scalability of the client/server structure [3]. From a reusability perspective, a distributed architecture based on web services is preferable to a server-side architecture. Distributed architecture can force software developers to write enterprise logic in the form of enterprise objects, allowing the operating system or other programs to reuse these enterprise objects, which is good for software developers and customer service, and can also

reduce system maintenance costs [4-5].

In fact, from the different signs of software development, distributed architecture has become an inevitable trend [6]. Kane M B introduced a distributed hydraulic control architecture with scalable computation and resilience to component failures in the network and physical domains. In this agent-based control system, an agent is defined as a set of co-located components in the cyber and physical domains that share sensations, actions, and/or goals. The proposed architecture consists of three types of agents: pumps, valves, and loads. These agents solve the convex constraint satisfaction problem, the steady-state mixed control problem, and the scalar bilinear model predictive control problem [7], respectively. Starikov A proposed the modern concept of Virtual Furniture Design Bureau (VFDB). In order to efficiently develop distributed projects in the multi-agent environment of VFDB, it is necessary to use a management system that ensures the coordination of the work of project participants, monitors the current status of the project, informs the project PARTICIPANTS VARIOUS SITUATIONS IN PROJECT WORK A feature of this control system is its invariance, i.e. independence from the design tools used in project work, and its possibility to dynamically adjust to the specific structure of the design process [8].

The Web services-based distributed system development approach in this paper plays a leading role in the development of service-based applications. The user program created based on this method has good flexibility, good reusability and good optimization. It is also based on open standards and supports dynamic integration between many types of users and applications. The partition processor algorithm is mainly used to allocate the number of processes to the data processing stream. However, when many applicants access the service, the server hosting the service remains a big challenge.

## **2. Research on Development Method and Support Tools of Distributed System Based on Web Service**

### **2.1. Distributed System**

The definition of a distributed system is mainly based on the characteristics of such systems, that is, a distributed system is a computer system with many interconnected resources that can work together to perform common tasks, relying on at least one centralized system. Terms like data and hardware. In the development process of distributed computing, for the software system of the application system, according to the number of layers between users, the application system can be divided into one-layer, multi-layer, three-layer or multi-layer software. Interface and background data [9-10].

In a two-tier system, if the business intelligence is placed on the client, the client layer is often very troublesome, and it will also bring a lot of trouble to routine maintenance and system function upgrades. If business intelligence is placed at the server level, it is often difficult to implement the complex logic of the application in the database system [11-12]. Therefore, in a three-layer or multi-layer approach, business logic is extracted separately, and one or more layers are created in the middle to form a true distributed application [13].

### **2.2. Web Service Component Development Sub-process**

The sub-process of Web Services component development is mainly represented by the Web Services component development life cycle [14]. The development life cycle consists of these four phases [15].

**Build:** The build phase of the life cycle consists of developing and testing the web service implementation [16].

**Deployment:** A deployment manager publishes definitions of service interfaces and service implementations to a service requester or service registry (Web Component Services Library) and deploys the web service to a runtime environment (usually a web application server) [17].

**Running:** During the running phase, Web Services can be invoked.

**Administration:** The administration phase includes the ongoing administration and operation of the Web Services application [18].

### 2.3. Aspire Extension Extension Platform Deployment

Based on the Aspire Extension adaptive browser extension high-performance extension development platform, the environment construction for extension development is mainly divided into three steps: Node local service deployment, Web extension application development directory construction, and IDE development environment installation.

(1) Node local service deployment. When the third chapter introduces the design and implementation of the back-end service of the adaptive browser extension development platform, the creation process of the Node service is described in detail. Web application developers, when developing extensions based on Aspire Extension, the first thing to do is to download the Aspire Extension extension development platform toolkit. In order to enable the Node background service to provide more powerful and convenient services, we also need to introduce third-party module packages such as grunt through the NPM package manager to execute npminstall Node will automatically introduce third-party packages according to the module directory in the dependencies in the package.json document. Modules, or developers can add third-party modules with arbitrary functions through npminstall+module identifier.

(2) The construction of the Web extension development directory. After the Node local service is deployed, take development under the Windows platform as an example, open the Windows command line window, and switch to the Node local service resource directory, (for example, execute the command: cddesktop/Aspire Extension to switch to The resource directory stored on the desktop), execute the nodecreate.js command to create the development directory.

(3) Install the IDE development environment. The development environment here is Web Storm (developers can also choose their own familiar IDE environment). WebStorm is a lightweight and powerful Javascript IDE developed by JetBrains, which is used for web application development. It provides a powerful client-side development environment and a Node.js-based server-side environment. The installation of Webstorm is very simple. Go to the JetBrains official website, download the latest version, and unzip it to any location.

## 3. Investigation and Research on Web Service-based Distributed System Development Methods and Support Tools

### 3.1. Distributed System Support Tools for Web Service Application Development

Any software engineering method, especially the component-based software development method, is inseparable from the support of tools and environments. The development process and development method of enterprise application system based on Web Service architecture also inevitably needs the support of a series of tools and information bases. These tools and information bases are integrated according to the ideas of data integration, control integration, expression

integration and process integration., the software engineering environment of distributed system can be formed. The distributed system environment that supports the development of enterprise application systems based on Web Service architecture should have the following basic capabilities:

- (1) Support requirements modeling;
- (2) Support framework modeling and component assembly;
- (3) Support the automatic generation of documents;
- (4) Support the whole life cycle management of Web Service;
- (5) Support basic management activities such as project management, configuration and change management, and quality management.

### 3.2. Componentized Development

A component-based development approach based on modular systems, structured design, and sound-based technology development. In component-based software development, the application system is composed of some general and specific standard components, which can be obtained through procurement, customization or self-development. Specific steps are as follows:

Custom manufacturing development is done through architectural requirement declarations, which can be understood as custom components.

Purchasing common core components or business components according to the requirements of the architecture is the development of common components.

Extract, modify and package fully reusable components from existing systems or component libraries.

### 3.3. System Stream Data Processing Task Assignment

After the system clarifies the current resource status, the system mainly allocates the number of processes for the current stream data processing task through the Assign Processors algorithm. The Assign Processors algorithm mainly passes:

$$k_i = \lambda_i / \mu_i + 1 (i = 1, 2, 3 \dots n) \quad (1)$$

That is, the arrival rate of the current node data is divided by the processing rate of the current node and added by one to calculate the number of processes required by each node. And compare the maximum number of resources  $K_{max}$  that the current system can provide, if:

$$\sum_1^n k_i > K_{max} \quad (2)$$

It means that the currently required resources exceed the system load and an exception is thrown.

## 4. System Example of Distributed System Development Method Based on Web Service

### 4.1. System Analysis

The system adopts a three-tier distributed computing architecture, and the first-tier client is the interface between the user and the entire system. Client applications may be limited to common browser software, such as Microsoft's IE, etc. The program has interactive features that allow users to enter information and submit in the background, submitting performance requests. The secondary web server will start the corresponding process to respond to the request, and can generate a string

of HTML/DHTML code, process the result and return it to the client's browser. If the request from the client can access the data, then the web server must also complete this process with the tertiary data server. The structural frame is shown in Figure 1:

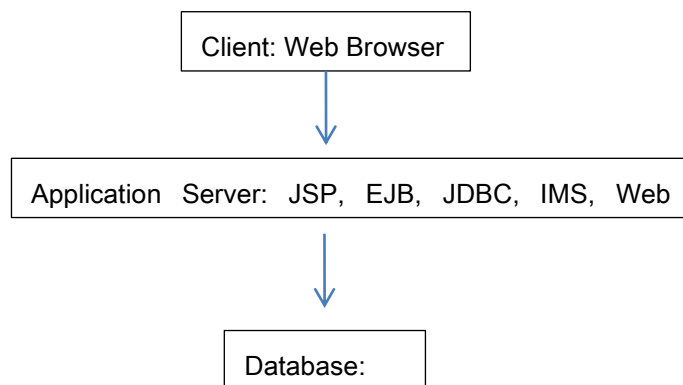


Figure 1. Three-tier distributed computing architecture

#### 4.2. Web Extension Front-end Performance Optimization

When testing and analyzing, we provide six functions such as real-time exchange rate, historical price, size change, express information, discount information, and operation prompts according to the Haitao extension, and divide the extension into six function points., the Haitao extension counts the changes in memory usage, as shown in Table 1.

Table 1. Changes in memory usage

Extension type	Real time exchange rates	Historical price	Size change	Express information	Discount information	Operation tips
Normal extension	12875	15317	12796	16842	16437	15428
Platform extension	8530	7975	7428	10214	7952	13274

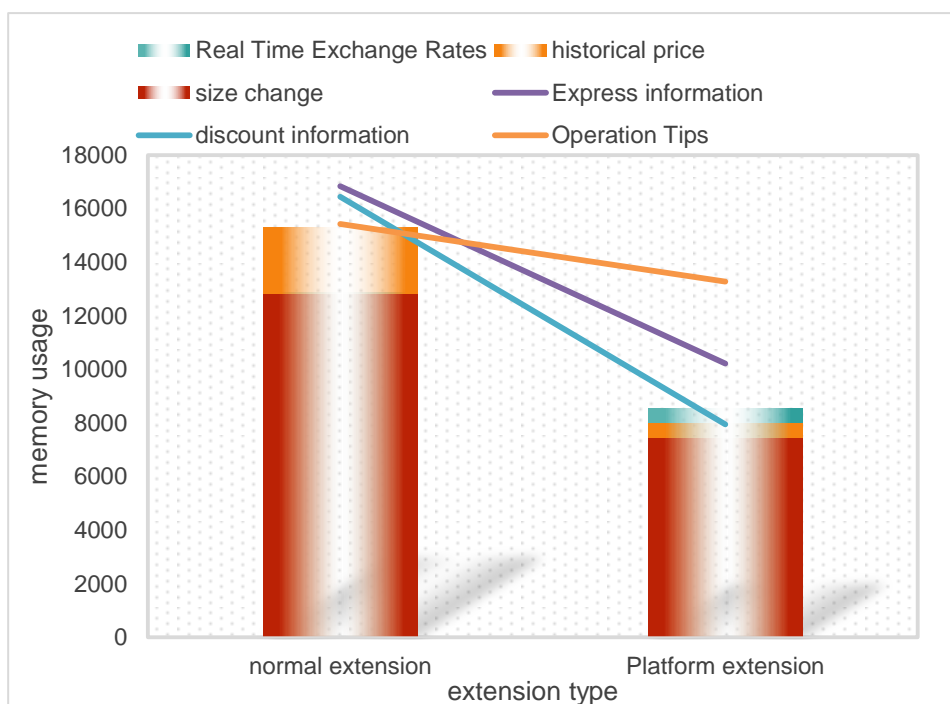


Figure 2. Six function point memory

Function 1 to function 6 in Figure 2 respectively represent the real-time exchange rate of Haitao Assistant, historical price, size change, express information, preferential information, and operation prompts for six function function points. Type-1 means that when the development system is not developed based on the Aspire Extension extension, when the Haitao extension is running inside the browser, as the user interacts with the extension function, the extension's memory space occupation of the browser changes. Type-2 means that when the development system is developed based on the Aspire Extension extension, when the Haitao extension runs inside the browser, as the user interacts with the extension function, the amount of memory space occupied by the extension on the browser changes.

Through comparison, it is not difficult to find that Aspire Extension introduces AMD module dependency loading and MVC program architecture into Web application development through AngularJS, RequireJS and other front-end development frameworks, thereby helping developers develop faster and better high-performance Web applications Extended application.

## 5. Conclusion

The theoretical research results of this paper are mainly reflected in the definition of distributed system architecture based on Web Service architecture, and the development process of Haitao system based on Web Service architecture. A preliminary attempt at research. On the basis of sufficient practice, this paper summarizes the main contents of the distributed Web Service application system development support environment; these works are an indispensable and effective part of the methodological research of this paper. The method used in this paper is informal, and the formal constraints have the advantages of ensuring the verifiability of the system, and the work in this area needs to be strengthened in future research. Therefore, the aspect-based software development method, as a new brand of software development, needs to be extended to

many fields such as methods, technologies, tools, etc. to meet the needs of development, and there is still a lot of work to be done.

### 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] Weisenburger P, Khler M, Salvaneschi G. *Distributed system development with ScalaLoci. Proceedings of the ACM on Programming Languages*, 2018, 2(OOPSLA):1-30. <https://doi.org/10.1145/3276499>
- [2] Salih H S, Egorov S Y. *Development Of A Monitoring System For Scheduled Works At Distributed Facilities. Vestnik Tambovskogo gosudarstvennogo tehnikeskogo universiteta*, 2020, 26(1):056-063.
- [3] MC Pérez-Pirela, JP Garc ú-Sandoval, Camacho O. *Development of a Simplified Model for a Distributed-Parameter Heat Exchange System for Thermodynamic Principles-Based Control Purposes. IFAC-PapersOnLine*, 2018, 51( 13):396-401. <https://doi.org/10.1016/j.ifacol.2018.07.311>
- [4] Kim D W, Park H J. *Development of Distributed Control System for Interpreter-based Educational Robot. Journal of The Korean Society of Manufacturing Technology Engineers*, 2020, 29(3):235-242.
- [5] Mishra R, Banerjee U, Sekhar T, et al. *Development and implementation of control of stand-alone PMSG-based distributed energy system with variation in input and output parameters. Electric Power Applications, IET*, 2019, 13(10):1497-1506. <https://doi.org/10.1049/iet-epa.2018.5882>
- [6] Altsybeyev V, Kozynchenko V. *Development of the distributed information system for the cooperative work under the design and optimization of charged particle accelerators. Cybernetics and Physics*, 2019(Volume 8, 2019, Number 4):195-198.
- [7] Kane M B, Lynch J P, Scruggs J. *Development of a Scalable Distributed Model Predictive Control System for Hydronic Networks with Bilinear and Hybrid Dynamics. Journal of Computing in Civil Engineering*, 2018, 32(5):04018038.1-04018038.10.
- [8] Starikov A, Meshkov D. *Multi-Agent Approach To The Development Of Management System Of The Process Of The Distributed Furniture Design. Actual directions of scientific researches of the XXI century theory and practice*, 2020, 8(1):238-243.
- [9] M Dülger. *Development Of The State Machine For The Distributed Elevator Control System Implementing Controller Area Network (Can). Global Journal of Engineering Science and Research Management*, 2019, 2(6):39-51.

- [10] Evdokimenkov V N, Kozorez D A, Krasilshchikov M N. *Development of pre-flight planning algorithms for the functional-program prototype of a distributed intellectual control system of unmanned flying vehicle groups*. *INCAS BULLETIN*, 2019, 11(S):75-88.
- [11] 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. <https://doi.org/10.1109/TSC.2017.2711019>
- [12] 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. <https://doi.org/10.15407/scine17.01.078>
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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. <https://doi.org/10.4018/IJWP.2021010103>
- [17] Meshcheryakova A A. *Distance Learning In Informatics Lessons Using The Online Test Pad Web Service*. *Informatics in School*, 2020, 1(7):47-56.
- [18] 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. <https://doi.org/10.5373/JARDCS/V12I7/20202054>