

A Smart Service Computing Platform Helping Users Constructing and Combining their Own Web Services

YuWeng¹, PengGuo², Xiaodong Jia³

¹College of Information Engineering,

¹Minzu University of China, Beijing, Beijing 100081, China

²College of Information Engineering,

²Minzu University of China, ² Beijing, Beijing 100081, China

³College of Information Engineering,

³Minzu University of China, Beijing, Beijing 100081, China

¹dr_wengyu@126.com, ²1125082967@qq.com, ³jxdshimon@gmail.com

Abstract

Under the circumstances of user's varied requirement, lots of web services are available and the number is increasing rapidly. In order to change service consumers into service providers with their own data and give them autonomous right, we designed a smart service computing platform which provides basic environment and infrastructure. As a platform, we cannot only make full use of our data resources through providing customized web service to target people, but also let users construct their own web services with their own data. Furthermore, combining the services and improving our efficiency of software development is our another main purpose. Therefore, in this paper, we propose a smart service computing platform to help users to publish their own web services and then combine the services into new ones with new functions.

Keywords: *publish web service, web service, cloud computing, service computing, SOA, cloud platform*

1. Introduction

In this big data [1-3] era, we individuals have been as service consumers using online resources, but we rarely consider how to become a service provider using our own data. Let alone combine the services we constructed into new ones and do service computing. Not because we do not want to do so, but there is no suitable method to help us.

When we are faced with new software requirements, we still have to start from the most basic part—coding until all the modules completed, however, most parts of these modules are the same, especially in software in an organization or a company. If we have such a platform, which not only make our software development not from coding, but from the combining well developed modules, and greatly improve the efficiency of software development, that will be a great thing. Generally, when we want to publish a service [4], the first thought is to use existing interfaces and available online data.

In order to implement the above functions, we designed a smart service computing platform—SSCP, when users use it to develop software, they only need to combine the services that the software needed simply. Deployment and execution of the software are by controlled by the SSCP.

Our platform is divided into two main models: service construction, service regeneration. Moreover, each model has its own components or layers those help us understand.

In this paper, we propose a platform helping users to publish their own web services[5,6] especially using their own data based on their own needs, then combining

the services published into new services and deploying them efficiently. The rest of the article is structured as follows: part 2 introduces several typical related works about web service construction and service regeneration. Part 3 describes the structure of the platform and how it works. The description includes two main models as said above: service construction, service regeneration. And then we give an experiment and verify the feasibility and efficiency of the platform by analyzing the experimental results in part 4. Lastly, some works about improving the model are talked about.

2. Related Works

Publishing web service is an important step of a web service's life cycle, Demian Antony D'Mello and his colleagues proposed A Service Broker Based Architecture for Effective Discovery and Automatic Publishing of Internet Web Services to improve the efficiency of discovery and auto-publishing [7]. In that paper, they mentioned publish, find and bind operations among the service provider, service discovery and service requester. Then a series of activities are presented to complete automatic web service publishing. But there rarely mentioned that help individual users or organizations to publish their own web services using their own data. And that's the difference between us. We mainly focused on helping users to become service provider rather than consumer especially using data source in their hands.

During the design process of service regeneration model, previous results have given us a lot of help, and this model is divided into mainly two parts: cloud computing and service-oriented architecture.

In cloud computing, we learned a lot from it, and we used some open source software of cloud computing. Rajkumar Buyya *et al.* [8] detailed the overall architecture of the cloud and introduced the future research direction of cloud computing. Michael Armbrust, *et al.* [9] gave a definition of the cloud computing, they also described the difference between cloud computing and general computing, pointed out the advantages of cloud computing. Finally, they elaborated the opportunities and challenges which cloud computing is facing. Anton Beloglazov [10], who designed a Data Center architecture that uses a new virtual machine allocation algorithm, with that algorithm Data Center's energy consumption is reduced. Athanasios V. Vasilakos [11], who summarized the different Data Center performance evaluation model and done some research in Live VM migration and the VM performance management.

In the aspect service-oriented architecture, Yi Wei and M. Brian Blake [12] described the characteristics of cloud computing and SOA, and they pointed that they can be combined. Mike P. Papazoglou *et al.* [13], who authoritatively described the SOA and its basic framework, detailed the structure and composition of service computing. Sasko Ristov *et al.* [14] designed a SOA-based cloud computing platform, and used several strategies to deal with the problem of load balancing.

3. Platform Description

Smart Service Computing Platform mainly has two model: service construction model and service regeneration model, just as Figure 1 shows.

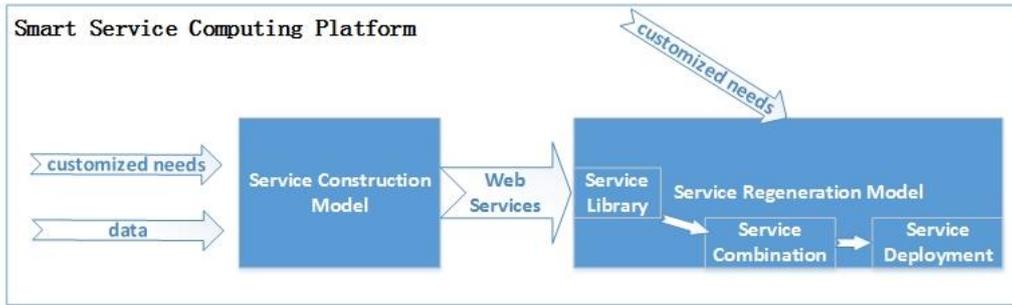


Figure 1. Structure of Smart Cloud Computing Platform

3.1. A Service Construction Model

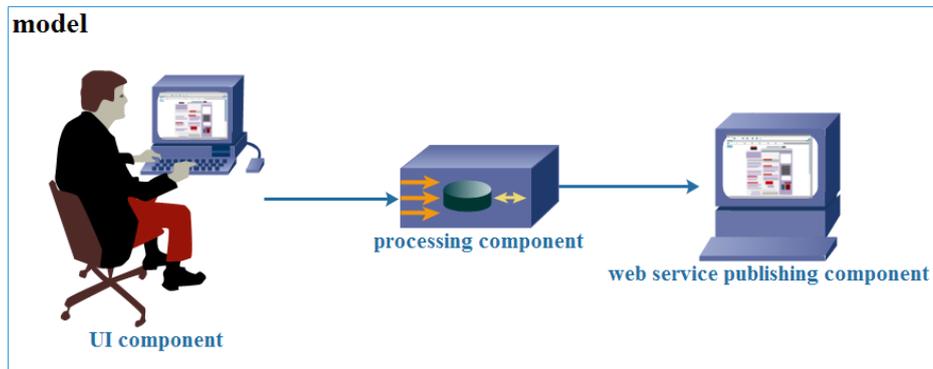


Figure 2. Structure of the Model

Service construction model we proposed mainly includes three components[15]: UI component, processing component, web service publishing component. Specific details of each component will be described later.

3.1.1. UI Component: UI component is just as its name, it provides UI[16][17] which mainly used to match data columns and displayed items and obtain data, web service information, customized information from user. And it provides the appropriate web pages to user. User can only see the simple web pages provided by the model and the background complex processing procedure is shielded. The web pages consist of input box and button for adding more entries. All the operations from user are completed by these simple controls, such as constructing a data table.

The data input by user is the source for the final publishing pages. Then the description of web service and some key words will be involved in the web service information. Besides, some other personalized demands, such as data processing method, are contained in the customized information.

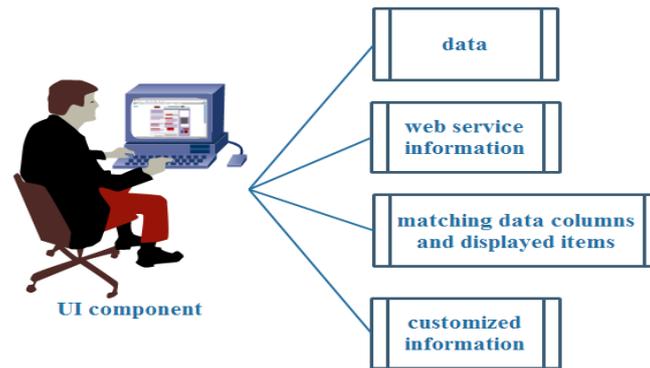


Figure 3. Structure of the UI Component

3.1.2. Processing Component: Processing component works according to the data and information from the UI component. Firstly, processing component stores the data in the database according to user's need, by the way, we use HBase in the experiment. Then according to the matching between data columns and the displayed items, processing component extract the data from database for the customized result. Finally, the inspector of the model will check the feasibility. If there are no errors, web service publishing component will be invoked.

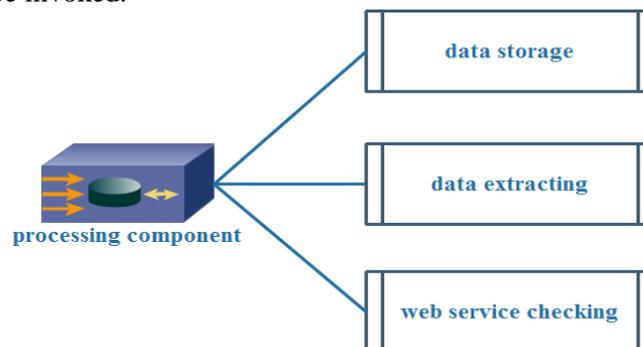


Figure 4. Structure of the Processing Component

3.1.3. Web Service Publishing Component: Web service publishing component is used to show the results of data processing and give the web service URL to target people. In addition, the customized result can be displayed in different ways ,such as displayed in a map based on its position or written back to the database. It all depends on user's customization.

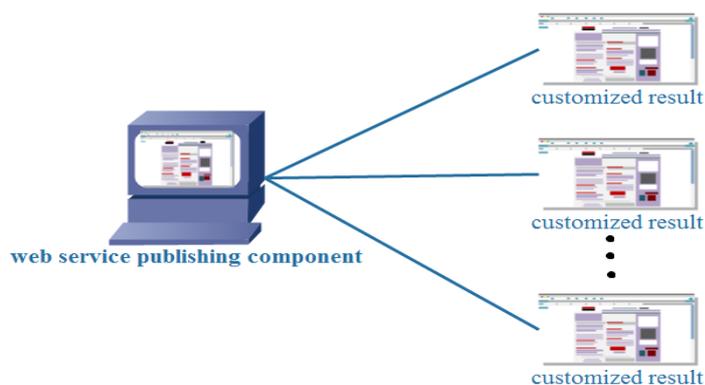


Figure 5. Structure of the Web Service Publishing Component

3.2 Service Regeneration Model

Referenced to architecture of cloud computing, service regeneration model is a four-layered service computing platform, from top to bottom, they are: service representation layer, service computing layer, the virtualization layer and the hardware layer. Above the service representation layer are users and software, they are not a part of SSCP. Figure 6 is the architecture of this model.

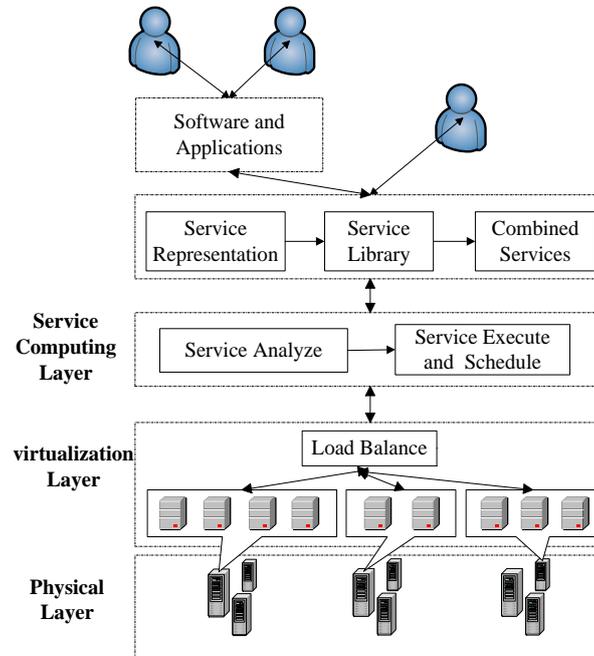


Figure 6. Architecture of Service Regeneration Model

This figure shows the different logical layers of service regeneration model

3.2.1. Services Representation Layer: Service representation layer is the uppermost layer of SSCP, and it is the interface of users and the software. SSCP is a service-oriented computing framework; therefore, service representation is essential in the SSCP. Like WDSL, we also use XML files to represent services, we called these files "service represent file". The service represent file describes the basic information of services, among which the data exchange interface between services and service combination information are the most important. Service data exchange interface defines what data is exchanged between the two services, and how to exchange data. There are three kinds of ways to exchange data, and they are networking, memory exchange and file read. Service combination information defined connection between services, and services will provide greater functionality through connection. Service composition enables service forming a graph, in which services are nodes, and the connection between services is the edge of the graph.

3.2.2. Service computing Layer: The services are executed in service computing layer. The service computing layer consists of two parts, and they are service scheduling systems and services analysis module. Service analysis module will first analyze the services and then assign the appropriate number of virtual machines for them. Because the resource needs of different services are different. After the service analysis, service scheduling system will schedule the execution of services.

3.2.3. Virtualization Layer: Virtualization layer contains all the virtual machines—VMs, we put heterogeneous virtual machine in SSCP, and these virtual machines running different software, such as Hadoop MapReduce and Apache Spark. The virtualization layer also contains a monitoring program to monitor the performance data of virtual machines and physical machines, which will be sent to a virtual machine scheduling program. This program is used to schedule virtual machines in order to achieve the load balancing of VMs.

3.2.4. Physical Layer: Physical layer contains all the physical machines and network equipment, and it is the foundation of the entire SSCP.

3.2.5. Software Development: After introducing the structure of SSCP, we will describe how to develop software in the SSCP. We offer a basic service library that contains all the services developed by us and other users in SSCP, and we provide the function of service querying, which can help users query their desired services by the metadata of services. After the user has got these basic services, they can develop software by modifying service represent files of these services. The mainly part users need to modify is the data exchange interface of services and service combination parameters. These files will be submitted to the service computing layer then. In service computing layer, service computing analysis module will first analyze the services, after that the service scheduling system will control the execution of them.

4. Experiment

We verify whether the model can work successfully or not according to the customized result in Figure 4. In this experiment, we used the existing cloud platform which can provide cloud storage and show data in a map using Web GIS[18][19].

Step 1:

According to their needs, user construct a data table and input data required when publishing a web service. As we talked above, input box and button which is easy for ordinary users play roles as adding more entries used during this procedure. After this, the data will be uploaded to database automatically. Then user describes his own web service and give some key words about it. Finally, customized information needs to be given. For example, we choose map using Web GIS as the way showing data processed in this experiment.

Step 2:

The processing component will deal with the data in database according to the demands from user. And all the demands are from UI component. And this step is invisible for user. Data will be stored in database through this component automatically. Then a part of the whole data will be extracted and match the items in customized result. And the result displayed in map as we chose previously which is only seen by inspector of the model.

Step 3:

If no errors are found in step 2, the web service publishing component will be invoked to publish the web service through customized result which is a map in this experiment.

Then we will continue the experiment to verify the service regeneration model. Because two or more services are needed, we choose two typical services into this experiment and introduce the experiment of developing a statistical application with service regeneration model. During the next steps, we first process data that stored in HBase with the help of Hadoop MapReduce. After that we display the results with a chart. We have some data stored in HBase, and now we want make a statistics for the data of construction field. One statistic is: Because of the geographical distribution of different

minorities, their houses also a great difference. In order to find the difference and links between these houses we want to classify these data according to the types of them.

Step 4:

We want to develop a service that reading data from HBase. This service is a common read service, and the most important part is to deal with the in and out parameters. In order to improve the scalability of our platform, we adopted the Key-Value form parameters to transfer data, and these parameters are configurable. Among the in parameters, besides the basic information of HBase, we add name of database table, column family and column name. Out parameters used Key-Value format too. Of course, we have done some preparations, like data cleaning.

Step 5:

We want to develop a service that uses Hadoop MapReduce for data classification. To connect with the former service, the in parameters of this service must be same as the out parameters of HBase read service. In the part of classification, we use Hadoop MapReduce to overwrite the Kmeans algorithm to make it suitable for distributed data processing.

Step 6:

We develop a service to display the statistics result. This service receives data and then displays it in the predetermined format. The most important part of this service is data interface too, so we have to modify the in parameters.

Step 7:

We combine these services together and submit them to the SSCP. SSCP will finish the deployment and execution. Figure 6 shows the results of the application.

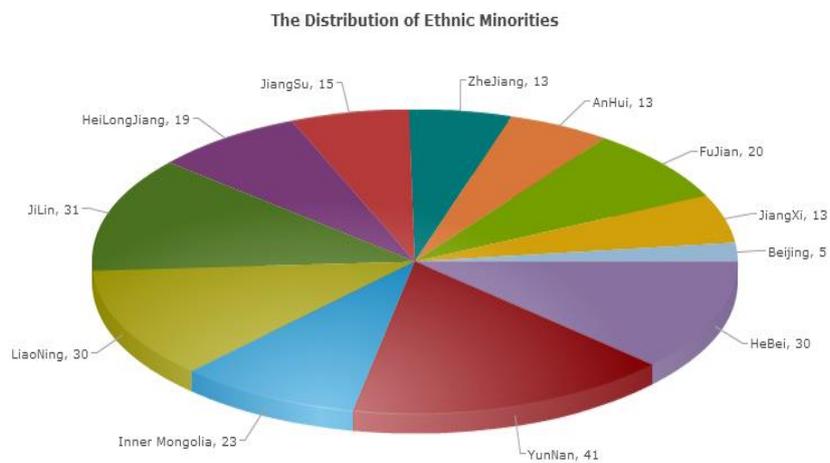


Figure 7. A Statistical Application Developed by SSCP

We classify ethnic minority according to the types of their constructions, and the result is little different from the existing ethnic classification. Figure 7 shows the statistics on the number of ethnic minorities in different provinces.

5. Conclusion

Following the step 1, step 2 and step 3, we got satisfactory results in the map with no error and the response time is acceptable. In other words, we have got a web service. After that the platform will add this service into the service library. Following step4 to step7, to improve the efficiency of software development within our group, reduce development

and running costs of software, we designed SSCP, and introduced the specific functions and composition of each part. SSCP provide simple service representation and software combined method. Experiment results showed that SSCP really simplifies the software development process.

But there are still some deficiencies in the SSCP, for example, service analysis is not smart enough and it did not fully optimize the use of virtual machine resources during the execution of the service. These issues will be addressed in our future research and we will also add some new technology.

Acknowledgment

At the design process of the SSCP, the rest of our group also helped us a lot, they are: Li Chengjie, Cheng Wenyi, Zhong Ling, Wang Ruiqi, Huang Ziwei and Bai Wencheng, thanks them here.

References

- [1] H. Hu, Y. Wen, T-S Chua, X. Li, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial", . Access, IEEE Volume: 2 DOI:10.1109/ACCESS.2014.2332453 Publication Year: 2014 , (2014), pp. 652 – 687.
- [2] X. Wu, X. Zhu, G-Q Wu, "Data mining with big data", Wei Ding Knowledge and Data Engineering, IEEE Transactions on Volume: 26 , Issue: 1 DOI: 10.1109/TKDE.2013.109 Publication Year : 2014 , (2013), pp. 97 – 107.
- [3] P Malik, "Governing Big Data: Principles and practices", IBM Journal of Research and Development Volume: 57 , Issue: 3/4 DOI:10.1147/JRD.2013.2241359 Publication Year: 2013 , (2013), pp. 1:1 - 1:13
- [4] K Ebner, T. Buhnen, N Urbach, "Think Big with Big Data: Identifying Suitable Big Data Strategies in Corporate Environments", System Sciences (HICSS), 2014 47th Hawaii International Conference on DOI: 10.1109/HICSS.2014.466 Publication Year: 2014 , (2014), pp. 3748 – 3757.
- [5] Y. Bai, "Practical Database Programming with Visual Basic", ASP.NET Web Services.NET DOI: 10.1002/9781118249833.ch9 , (2012), pp. 657 - 859 .
- [6] S Sowmya Kamath, A. Ananthanarayana, "A bottom-up approach towards achieving semantic web services", V.S. India Conference (INDICON), 2014 Annual IEEE DOI:10.1109/INDICON.2014.7030399 Publication Year: 2014 , (2014), pp. 1 – 6.
- [7] D. Antony, D. Mello1, S. Salian2, Sahana3, N. Shett, C. Rodrigues5 and C.D'Souza6, "Service crawler based effective and dynamic discovery mechanism for Web Services available over the Internet", 2nd IEEE International Conference on Parallel, Distributed and Grid Computing (2012).
- [8] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, pp. 599-616, 6// (2009).
- [9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, *et al.*, "A view of cloud computing," Communications of the ACM, vol. 53, pp. 50-58, (2010).
- [10] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, vol. 28, pp. 755-768, 5// (2012)
- [11] X. Fei, L. Fangming, J. Hai, and A. V. Vasilakos, "Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey", State of the Art, and Future Directions," Proceedings of the IEEE, vol. 102, (2014), pp. 11-31,
- [12] W. Yi and M. B. Blake, "Service-Oriented Computing and Cloud Computing: Challenges and Opportunities," Internet Computing, IEEE, vol. 14, pp. 72-75.
- [13] M. P. Papazoglou, "Service-oriented computing: concepts, characteristics and directions," in Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on, 2003, (2003), pp. 3-12.
- [14] S. Ristov, M. Gusev, G. Armenski, K. Bozinoski, and G. Velkoski, "Architecture and organization of e-Assessment cloud solution," in Global Engineering Education Conference (EDUCON), 2013 IEEE, 2013, (2013), pp. 736-743.
- [15] Q Zhang, X. Wang, K. Xie, F. Yang, "Research on Component Quality Model Computing", Measurement, Control and Sensor Network (CMCSN), 2012 International Conference on DOI: 10.1109/CMCSN.2012.85 Publication Year: 2012 , (2012), pp. 348 – 351.
- [16] H. Han, P. Gao, K Oyama, "Retrieval, description and security: Towards the large-scale UI component-based reuse and integration", Information Reuse and Integration (IRI), 2011 IEEE International Conference on DOI: 10.1109/IRI.2011.6009545 Publication Year: 2011 , (2011),

- pp. 193 – 199.
- [17] M. Zhao, Y. Gao, C. Liu, “Research and Achievement of UI Patterns and Presentation Layer Framework Computational Intelligence and Communication Networks (CICN)”, 2012 Fourth International Conference on DOI: 10.1109/CICN.2012.175 (2012).
- [18] Z Jovanovic, M. Tomic, S. Randic, Realization of Web GIS platform using open source technologies. Telecommunications Forum (TELFOR), 2012 20th DOI: 10.1109/TELFOR.2012.6419501 Publication Year: 2012, (2012), pp. 1488 – 1491.
- [19] H. Guo, Y. Xue, M. Zhang, “Comprehensive service system for digital community design based on web GIS”, Electrical & Electronics Engineering (EESYM), 2012 IEEE Symposium on DOI: 10.1109/EESym.2012.6258634 Publication Year: 2012, (2012), pp. 239 – 242.

Authors



Yu Weng. Yu Weng is associate professor of College of Information Engineering, Minzu University of China. His current research interests are distributed computing and service computing.



Peng Guo. Peng Guo is a full-time graduate student in College of Information Engineering, Minzu University of China. And he majors in computer science and technology. His current research interests are cloud computing and AI.



Xiaodong Jia. Xiaodong Jia is currently working toward the Master degree in computer science and technology at Minzu University of China. His current research interests are cloud computing and big data.

