ATLANTIS PRESS

Advances in Computer Science Research, volume 70

2nd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2017)

Application of Multi-tenant Service Customization Algorithm Based on Multi-target Ant Colony Algorithm in Cloud Platform Software as a Service

Weibo Li

Wuhan Textile University, Wuhan 430073, China

Keywords: Ant colony algorithm, Multi-tenant, SaaS, Service customization.

Abstract. Multi-tenant service customization is one of the core technologies to implement the SaaS multi-tenant software architecture, which can meet the changing needs of individual tenants. In order to broaden the intelligent application of ant colony algorithm in SaaS and improve the service quality and efficiency of SaaS platform, a multi - tenant service customization algorithm based on MapReduce and multi - target ant colony algorithm is proposed. From a number of business processes and mass services, the multi-tenant service customization algorithm sets the most suitable business process and optimized service portfolio for tenants. Multi-tenant service custom algorithm designed multi-target ant colony algorithm, and applied the MapReduce cloud computing technology. In a cloud computing environment, it runs in a distributed and parallel manner to optimize tasks. The results show that the multi - tenant service customization algorithm has a good convergence and expansibility in solving the multi - tenant personalized service customization problem. The algorithm has the ability to handle massive data and large-scale problems.

1. Introduction

Cloud computing technology is gradually changing the whole academia and industry [5]. The design and development of computer hardware and software are facing great challenges [3]. The National Institute of Standards and Technology divides the services in the current cloud computing environment into three levels: (1) Infrastructure as a Service (IaaS), such as IBM's blue cloud, Sun's cloud infrastructure platform (IaaS), and Amazon's elastic compute cloud (referred to as EC2). (2) Platform as a Service (PaaS), such as Microsoft's Azure platform, and Google's Google App Engine. (3) Software as a Service (SaaS), such as salesforce's customer relationship management services [2]. As a next generation computing model, cloud computing plays an important role in both scientific computing and business computing [1]. Software as a service application platform has promoted further development of the software as a service model. SaaS application model has gradually become an important way for small and medium enterprises and individuals to apply advanced technology [4]. Through the SaaS application platform, more and more units or individuals lease a variety of application services, so as to achieve their business needs. These application services have covered all areas of modern life [7].

The SaaS delivery model provides the application software to the user in the form of a service [8]. Users can reduce the cost of software application and enhance the flexibility of business change by renting software. Multi-tenant is one of the core strengths of the SaaS platform. The typical feature is "single instance multi-lease", that is, multiple tenants share a service provider of an application example, and the different tenant services are physically shared. For each tenant, this service instance seems to be only for their own services. SaaS platform has a large number of services, and tenants demand is very different. Each tenant is expected to have a suitable tenant application. As a result, service customization has become an important aspect of multi-tenant applications. Cloud computing is developed on the basis of parallel computing science, distributed computing science and grid computing science, and it can be said that it is the commercial realization of these computational science. The ant colony algorithm is proposed by the Italian scholar Dorigo M et al [6]. It is a kind of bionic swarm intelligence optimization algorithm, which has strong universality and robustness. Ant

colony algorithm has been successfully applied to solve NP problems of many combinatorial optimization problems, such as quadratic allocation of vehicle scheduling, communication network load, and integrated circuit design. Due to the fact that the performance of ant colony algorithm is very good, it has aroused widespread concern. Therefore, this paper studies the ant colony algorithm in cloud computing environment and its application in SaaS platform to solve the problem of multi-tenant service customization.

2. Sate of the art

At the end of 2007, IBM announced the cloud computing plan. Since then, the concept of cloud computing has emerged. At present, large international companies have set up their own cloud computing platform and provide a variety of cloud computing services.

IBM's cloud computing platform: IBM's blue cloud platform supports open standards and open source software [9]. The blue cloud computing platform infrastructure uses Xen and Power VM virtualization software, the Linux operating system, and the Apach Hadoop software. The use of virtual machines and large-scale data processing software Hadoop is a key feature of the "Blue Cloud" software platform. IBM has officially launched the "blue cloud" product based on the x86 chip server system.

Amazon's elastic compute cloud (referred to as EC2): Through the UI interface of EC2, the user operates various instances in the platform. Users only need to pay for their own use of the instance, and they do not need to build their own cloud computing platform. Through the use of leasing, tenants use a variety of cloud computing resources. This meets the needs of small and medium-sized software developers for cluster systems, thereby saving the cost of purchasing, managing and maintaining equipment.

Google's cloud computing platform: The platform serves Google search applications. Google's cloud computing architecture consists of four systems: Google File System (GFS) distributed file system, MapReduce programming mode, Bitable model simplified large-scale distributed database and Chubby distributed lock mechanism. The GFS distributed file system takes full account of the failure of a large number of nodes. It divides the file into small data segments, which ensures the fault tolerance by using policies such as multiple copies of stored data segments. MapReduce programming mode makes programming in a distributed system easier and simpler. Developers only need to write their own Map function and reduce function. They will be able to run their own programs in a distributed cloud computing platform, process large-scale data, complete tasks, and implement application requirements [10]. In this way, developers can focus more on the task itself, they do not need to pay too much attention to cloud computing platform reliability, security, scalability and other technical aspects. The cluster processing issues are handled by the cloud computing platform. Bitable is a weakly consistent large-scale database system that can handle large-scale structured and semi-structured data. Its application mainly includes Maps, Orkut, and RSS reader and so on. Chubby is not only a distributed and highly available data lock service, but also a very robust coarse-grained lock.

3. Methods

3.1 Multi - tenant service customization process

In order to clarify the relationship between the various parts of the multi-tenant service customization, we give a multi-tenant service customization process, as shown in Figure 1. First of all, according to the functional requirements of the tenants and abstract services in the abstract service layer, we use the business process component tools to analyze and model, so as to form a number of business process models. Second, according to the non-functional requirements of the tenants, the QoS of the specific services and the multiple business process models of the tenants, we use the multi-tenant service customization algorithm to construct a set of non-inferior specific service portfolios, so as to form a service customization for the tenants. This is an executable specific service portfolio. It can

perform the specific service portfolio to complete the tenant application. Finally, the tenants choose the most satisfying specific service portfolio based on their preferences, and other specific service portfolios will be enabled as an alternative to prepare for the occurrence of the tenant service.

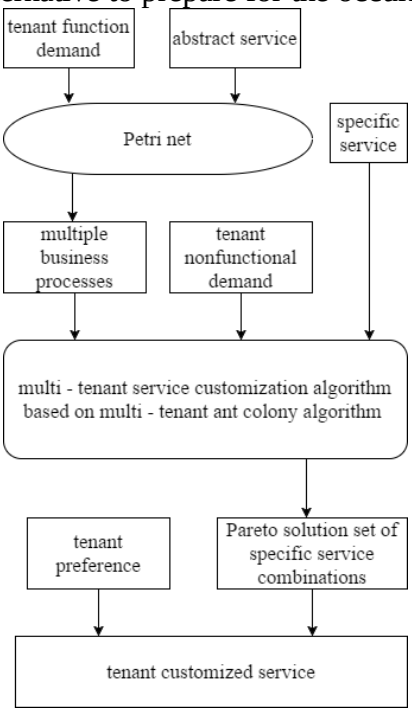

Fig. 1 Multi - tenant service customization process

3.2 MapReduce programming mode

MapReduce is a software architecture. The operation model of MapReduce contains m Map and r Reduce. Each Map handles the different parts of the original data, and the maps are independent of each other to realize full parallelization. In the output of the Map function, the result of the same key is merged into a node. Then, they are then processed by the same Reduce. Each Map processed by Reduce does not cross each other and can be executed in parallel. After simply assembling, all the results produced by Reduce form a complete result set to output. MapReduce is used for distributed parallel computing and processing of large-scale mass data.

The limitations of MapReduce are: (1) Poor flexibility. The MapReduce architecture breaks down the processing of the problem into a Map function and a Reduce function. However, for many problems, it is difficult to abstract or decompose it into Map functions and reduce functions. (2)When implementing iterative algorithms, MapReduce is less efficient. (3) When performing multiple data sets, MapReduce is less efficient.

3.3 SaaS

SaaS is targeted at cloud computing end users. It provides users with Internet-based software services. Typical SaaS service applications include Salesforce CRM and Google Apps and so on. SaaS application model has gradually become an important way for small and medium enterprises and individuals to apply advanced technology. SaaS service model makes the cloud computing service providers and users to achieve a win-win situation. Similar to infrastructure (such as power grids), cloud computing relies on sharing resources to achieve economies of scale. Cloud computing service providers integrate a large number of resources available to a large number of users, and ensure the confidentiality and security of user data through effective technology. Users do not need to buy hardware and software equipment and hire specialized IT staff. They rent software services according to actual needs. According to their own needs, users adjust the use of resources in a timely manner. When resource demand increases, tenants only need to increase the amount of resource rent. When demand decreases, the tenant returns the leased resources, thereby reducing the cost of hire. Service providers can reallocate these returned resources to other users, or even adjust the rent in line with the overall demand. For example, in the period of large leasing of resources (such as day), the rent is

relatively high. In periods of small leasing of resources (such as night), the rent is relatively low. In this way, through the adjustment of rental prices, SaaS can promote changes in user needs, so as to achieve the balance of resources and full use.

4. Comparison of MSCMA algorithm before and after optimization

The experiment is designed to compare and analyze the results of optimization and nonoptimization schemes. Suppose that the MSCMA with good solution preserves the diversity of the solution is the algorithm 1, denoted as A1. Suppose that the algorithm for preserving diversity is not used for algorithm 2, denoted as A2. Suppose that the algorithm without the use of the good solution is algorithm 3, denoted as A3. The two algorithms that are not used for optimization are the algorithm 4, denoted as A4.

The experimental steps are: the specific service size m is 10. We carry out experiment to the algorithm A1 ~ A4, respectively. The algorithms A3 ~ A4 are updated for the algorithm pheromone. The results are shown in Table 1 and Figure 2.

Table 1. The number of iterations on getting paretos

N	3	5	7	9	10	11	12	13	14
A1 algebra	2	/	/	/	3	/	8	12	13
A2 algebra	7	10	12	16	17	19	22	30	31
A3 algebra	13	16	34	35	39	36	32	89	90
A4 algebra	16	26	48	99	141	151	215	263	264

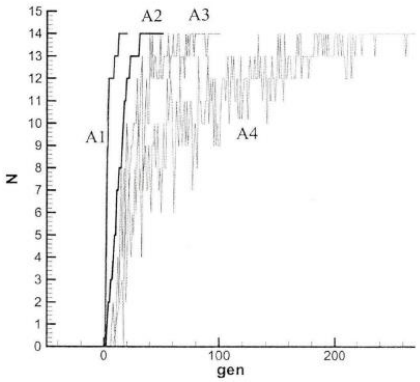

Fig. 2 solving process comparison of algorithms

In Table 1, N is the number of non-inferior solutions. A1 algebra ~ A4 algebra indicates the maximum value of the running algebra when the algorithm obtains N (N=1~13) non-inferior solutions. The value of N = 14 is the convergence algebra of algorithms A1 ~ A4. The space of A1 (such as N = 5) is because the algorithm A1 in the second generation to get three non-inferior solution and in the third generation to get 10 non-inferior solution. Therefore, there is no 5 (N = 5) solution evolution process. Figure 2 shows the change process of the non-inferior solution of each algorithm with the algebraic. From Table 1 and Figure 2, we can see that the algorithms A1 and A2 are fast and convergent and stable by using the optimal solution strategy. The number of non-inferior solutions of the algorithm A3 and A4 fluctuates greatly with the algebra because the strategy is not used. Therefore, in Table 1, there is a phenomenon that the values of N = 11 and N = 12 in A3 are smaller than N = 10 (that is, the least algebra 32 of the 12 non-inferior solutions of the algorithm A3 is smaller than the least algebra 39 of 10). Due to the adoption of diversity preserving strategy, the solution result of the algorithm A1 is better than the algorithm A2. The results show that the performance of the optimized MSCMA algorithm is greatly improved.

5. Conclusion

This paper studies the ant colony algorithm based on MapReduce to solve the multi - tenant service customization problem in SaaS. First of all, this paper analyzes the related concepts, theories, key technologies and implementation methods of multi-tenant service customization, and gives the

customization process of multi-tenant service. Then, the ant colony algorithm based on MapReduce is applied to solve the multi - tenant service customization problem, and a multi - tenant ant colony algorithm based multi - tenant service customization algorithm (MSCMA) is proposed. MSCMA uses excellent solution preserving strategy and diversity preserving strategy to optimize the algorithm. The application of cloud computing technology makes the algorithm run parallel and distributed in the cloud platform. It improves the algorithm's ability to deal with massive data and large scale problems. Experiments were carried out to compare the optimal strategies and achieved good results. The results show that the intelligent application of MSCMA algorithm in SaaS has a good practical value and some theoretical significance.

References

[1]. Almorsy, Mohamed, John Grundy, and Amani S. Ibrahim. "Adaptable, model-driven security engineering for SaaS cloud-based applications." Automated Software Engineering 21.2 (2014): 187-224.

[2]. Espadas, Javier, et al. "A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures." Future Generation Computer Systems 29.1 (2013): 273-286.

[3]. Jumagaliyev, Assylbek, Jonathan Nicholas David Whittle, and Yehia Said Shahat Ahmed Elkhatib. "Evolving multi-tenant SaaS cloud applications using model-driven engineering." (2016): 60-64.

[4]. Kaur, Parmeet, and Shikha Mehta. "Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm." Journal of Parallel and Distributed Computing 101 (2017): 41-50.

[5]. Kaur, Parmeet, and Shikha Mehta. "Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm." Journal of Parallel and Distributed Computing 101 (2017): 41-50.

[6]. Kapur, Ritu, and Maitreyee Dutta. "Review of various Load Balancing and Green Computing Techniques in Cloud." the Proc. of 7th International Conference on Advances in Computing Sciences, Software Solutions, E-Learning, Information and Communication Technology (ACSEICT-2015), Journal of Basic and Applied Engineering Research (JBAER). Vol. 2. No. 2. 2015.

[7]. Moens, Hendrik, et al. "Cost-effective feature placement of customizable multi-tenant applications in the cloud." Journal of network and systems management 22.4 (2014): 517-558.

[8]. Mishra, Sambit Kumar, et al. "Metaheuristic Approaches to Task Consolidation Problem in the Cloud." Resource Management and Efficiency in Cloud Computing Environments (2016): 168.

[9]. Rimal, Bhaskar Prasad, and Martin Maier. "Workflow Scheduling in Multi-Tenant Cloud Computing Environments." IEEE Transactions on Parallel and Distributed Systems 28.1 (2017): 290-304.

[10]. Walraven, Stefan, et al. "Efficient customization of multi-tenant Software-as-a-Service applications with service lines." Journal of Systems and Software 91 (2014): 48-62.