Send Orders for Reprints to reprints@benthamscience.ae
The Open Cybernetics & Systemics Journal, 2015, 9, 433-442 433
Open Access

SaaS Resource Management Model and Architecture Research

Zhang Xiaodong[1,2*], Zhan Dechen[1] and Chu Dianhui[2]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, P.R. China; [2]School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai 264209, P.R. China

Abstract: Nowadays, many enterprises are confronted with a real problem in their operational processes, which is the mismatch between the active demands and resource. To solve this problem, traditional literatures employed Resource aggregation of Cloud Computing. However, most of these researches focused on computing resources but only few studied the application resources, therefore a complete solution has not yet been proposed. In this paper, a resource service model RSM, which is dynamic, customizable and fixable, is presented. This model can assist the enterprise for adding surplus resources to the cloud resource pool. In this study, an SaaS resource management architecture CARMA was built. The CARMA was used to solve the problems in sharing and management resource of enterprises. Meanwhile, resource optimization and selection were introduced to CARMA to reduce the resource waste and task delay. This method was applied in the transport service field, where practical work proved that the proposed model is feasible which achieves the expected results.

Keywords: Private resource management, Resource service model, SaaS resource management, Service equivalent.

1. INTRODUCTION

Cloud computing is a kind of service resource pool in which a lot of resources are aggregated. Tenants dynamically select a number of resources to serve themselves according to their own demands [1, 2]. Cloud computing mainly provides resources for computing, such as CPU, disk, memory, network, software, data and so on. The attribute of quality of service (QoS) is relatively simple. It generally uses time or cost as the parameters of QoS to establish the model which is based on performance and is economical [3]. The purpose is to seek the shortest running time (the optimal performance). Besides computing resources, other shared resources are also required in cloud computing based industrial application system. For example, manufacturing cloud includes the hard manufacturing resources, the soft manufacturing resources and the manufacturing capacity [4, 5]. The logistics cloud includes the hard distributing resources (such as transportation of vehicles, manpower, *etc*.), the soft distributing resources (such as data, software, knowledge, *etc*. in the process of transportation) and distribution capabilities (such as distribution, scheduling and integration, *etc*.) and so on. Enterprises have started incorporating some resources which can be shared to the cloud. On one hand, this can improve the utilization rate of resources, on the other hand, it can also widely promote cloud application. However, the types of resources in application system are various, which makes their access and management more difficult. In addition, there are some different constraints and evaluations between the user's demands for the QoS and QoS of cloud resources in different fields. The requirement of users is not only restricted to time and price, but usually includes TOPS (Time, Quality, Price, Service *etc*.). Therefore, scheduling based on performance or economics cannot meet the actual needs of the enterprise.

For a case of transport of enterprise, there is usually such a situation where resources provided by Resource Service Provider (RSP) are idle, but enterprises which need these resources can not find them. The cause is as follows: 1) Poor system of information exchange and lack of adequate information; 2) Lack of an effective solution, which meets the interests of both the supply and demand sides at the same time, due to numerous resources and tasks. The reason for these drawbacks is inadequate and faulty resource management mode. Considering these issues, the characteristics of cloud resources are discussed and researched in this study. Following this, the paper also constructed the resource model, management architecture and service mode and applied them to the related field.

2. RELATED WORK

2.1. Resources Service Mode and Optimization Schedule

Resource is usually integrated into cloud environment *via* the service mode. After combining cloud computing with network manufacturing, Li Bohu *et al.* [4, 5] presented a structural design of cloud manufacturing and introduced virtual resource layer for integrating resources. This architecture can access all kinds of resources by the cloud technology, achieve the servitization and virtualization of resources, and enhance the 'many to one' service mode of 'scattered resources concentrated use' to 'many to many' service mode. Liu Lilan *et al.* [6] studied the current networked manufacture and various patterns of manufacturing cloud, and proposed four levels of manufacturing cloud architecture which uses the manufacturing resources as the bottom level of cloud computing, emphasizes on encapsulating distributed

*Address correspondence to this author at the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, P.R. China; Tel: 13863151206; E-mail: z_xiaodong7134@163.com

1874-110X/15 2015 Bentham Open

Table 1. Analysis of resource's characteristics in cloud environment.

Resource Feature	Features Cause	The Cloud Resource Management Technology
Autonomy of Distribution	In different organizations and geographic areas, the owner masters full knowledge of resources, can control and manage resources	Provide resources integration model, through the interaction with resource management system of different domain or resource calendar provided by RSP to schedule resources
Heterogeneity	A wide range of different characteristics	Define standard resource management model and information interaction protocol [12]
The diversity of technology	Different organizations have different resources management, scheduling and maintenance policies to resources	Make standard mechanism of resources and users' demand expression [13] and build extensible resources framework
Dynamic	Resource's configuration, ability, and running state dynamically changes constantly in the process of operation	Have a certain adaptive ability and can handle the fault tolerance of failure
Non-real-time	Production cycle is long, the process is relatively independent	Establish resource calendar, monitoring resources, scheduling resource
Perception	By the Internet of things and RFID technologies and so on, the current information of the resources can be gotten real-time	Access to awareness technologies of the Internet of things and RFID
Collaborative	More complex task requires resources from different providers work together to complete	Understand security mechanism, the resources characteristics of different areas, do task tracking, form "many-to-many" service mode

resources in the form of services and manages them in a centralized manner. Casati F *et al.* [7] established a matching and search framework based on the task's functional requirements, and proposed an optimization algorithm based on intuitionistic fuzzy set resources. When a resource is evaluated, trading experience, decay with time and other non-functional services quality attributes are added. This kind of methods cannot effectively guarantee to display the service features comprehensively and is more oriented towards the theoretical study of service selection rather than practical and general. In order to further improve the condition for utilizing resources , scholars studied a large number of algorithms, such as service network of the simulated annealing algorithm [8], ant colony system optimization algorithm based on trust perception [9] and genetic algorithm based on multi-objects GODSS [10], *etc*. The resource aggregation service chain constructed by these methods is the optimal single solution satisfying the constraint conditions rather than providing acceptable multi-solutions, which cannot fully show the quality of services and motivate service providers to optimize service quality. For achieving network based resources, the most common method is to package the resource as a service. After that, many optimized selections and schedulings for services can be used for the resource optimization scheduling. Tao F *et al.* [11] proposed a group decision making fuzzy hierarchical analysis model based on service quality. The whole process of resource scheduling can be divided into resource search, resource scheduling based on the service quality and dynamic interaction and consultation with three stages, along with fuzzy analytical hierarchical process applied for scheduled resources to reach a group decision.

The above studies generally believe that the resource can be encapsulated as a service, which can effectively shield heterogeneity, distribution and diverse characteristics of different resources, thus reducing the difficulty of selection and scheduling algorithm. In practice, however, if the related features of the resources are not considered, it usually blocks the optimal selection of resources, thus making it impossible to achieve the ideal effect. If those features are considered, the complexity of management is increased. Faced with this situation, this paper proposed an extensible architecture, in which RSP and RSD can dynamically select the resource characteristics, build resource model, obtain optimal selection and scheduling scheme of resources through relevant algorithms and simultaneously reduce management difficulty.

2.2. Analysis of Resource Features in the Cloud

The resources in cloud computing management from different RSPs are widely distributed, heterogeneous and dynamic. The cloud resource management does not have a complete control of resources, therefore, it cannot predict the state of resources. Thus, the heterogeneous resources greatly complicate the resource management. All of the above limitations make the management of resources and scheduling tasks difficult. Therefore, this paper summarized and analyzed the features of resource management in cloud computing and introduced the technologies of management in the cloud environment. All the results are shown in Table **1**.

3. RESOURCE SERVICE MODEL AND RESOURCE MANAGEMENT ARCHITECTURE

3.1. The Scene Description and Analysis

Firstly, the cloud resource service management should solve the problem of the service mode. For this purpose, it is required to analyze and design the resource management framework. The scene analysis is very useful to define the style of the system framework [14]. This paper examined the validity of the style of resource management framework based on the scenario shown in Fig. (**1**). In service man-

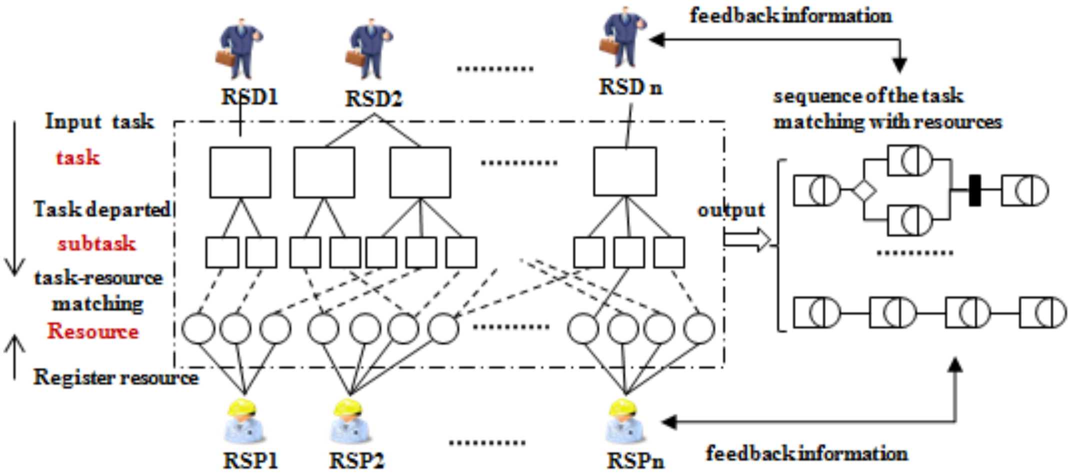

Fig. (1). Cloud resources service operation scene.

agement, the Cloud Platform supports the RSPs to register different kinds of resources in the cloud to realize 'the unified management of scattered resources'. At the time of sharing, RSP can also manage resources independently, which includes monitoring the running state of the resources, allocating resources by utilization rate and benefit maximization, and cooperating with other resources. Cloud resources, packaged into services, can be assembled into different processes and serve for different users, which is different from the network resource management mode. As a result, the mode turns from 'many to one' into 'many to many'. The system can receive feedback information received from different users and specifically formulate and improve the service strategy. RSD submits the tasks to the cloud and manages tasks independently. RSD can divide the tasks to control the workflow. According to the requirements, the tasks can be executed by the specific resources which are allocated independently by RSD, or can be assigned by the system automatically. RSD can monitor the running state of the tasks, investigate the optimized solution of the task and provide feedback of the problems related to the quality of service.

In most of the cases, the RSD does not consider the problem of discontinuity when running the tasks. When a resource encounters a problem, the system assigns a new resource to execute the task dynamically and timely provide feedback of the information to RSD. In other words, the cloud platform can provide indirect guarantee for establishing a reliable relationship between supply and demand by resource aggregation of the cloud platform.

3.2. Resource Service Model

As shown in Fig. (**1**), the resource management is divided into two parts: a part of it shows the functional management of resources based on their purpose, including capacity, consumption, operations, *etc*., and determines how to match the task. The other one shows the non-functional management of monitoring and scheduling resources, including scheduling, coordination, control, optimization, *etc*., by displaying how to implement the task based on the resource. This can better exploit the advantages of resources only if both the two functions are combined organically. The cloud resource management, therefore, can be able to aggregate different types of resources into the system by fully embodying the features of them. On one hand, it allows RSP to manage their resources independently; on the other hand, it also allows the system and RSD to schedule the resources according to the requirements of the task. The paper built a multi-tenant, customizable, and scalable SaaS resource model based on the scenario shown in Fig. (**1**). Resource management's goal is to make better use of resources. It is closely related to the activities of the tasks linked with the resources. The elements (object) and the relationship between the elements in the resource model are described by IDEF3, as shown in Fig. (**2**). In order to accurately describe the objects and the relationship between them in the resource model for convenient computing, this paper highlights the following definitions:

Definition 1: In Resource Meta Model RMM={G,P,T} , where G is the general resource type, each kind of resource has attributes. For example G = {resource identifier, and resource type, resource location ...}; P is the private resource type, incorporating private attributes and methods; T is represents the resource types; different resource types have different private resources.

Definition 2: Resource Private Type

$$P=\{(t,E,A,f,C)|e=f_e\,(E,C,t)\&a=f_a\,(A,C,t),e\in E,a\in A,t\in T,f_e,f_a\in f\}$$

The resource service capability attributes are represented as: E = $\{E_1, E_2, E_3,... E_n\}$, where E describes the resources with multiple dimensions. Private resource attribute A = $\{a_1, a_2,... , a_n\}$, implies that a resource can have many features. f_e, is the capacity constraints of the model, and f_a is the customized method; both of them are used to constrain model computing and expression.

The biggest difference between E and A is that A has only two values 'existence' or 'inexistence'. There is a need to point out the scope value of E, such that the vehicle's load is divided into values of 5 tons, 10 tons *etc*. Though both E and A are customizable, but they are relatively stable during the entire life cycle of resources.

According to definition 1 and 2, RSPs can customize resource information according to their own resource characteristics, register their resources into the system, and manage them. The main activity is the management task, the prop-

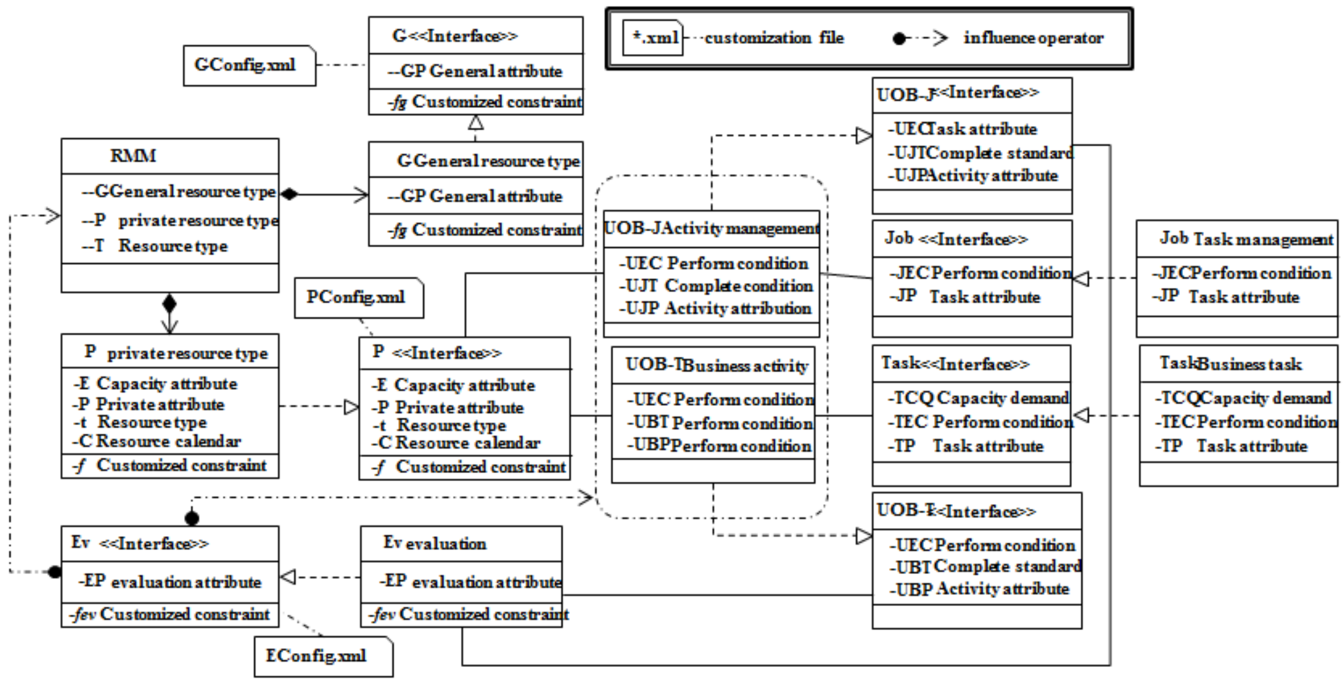

Fig. (2). Resource service model.

erties (JP) of which express the characteristics of the management task and these tasks are performed under a certain condition (JEC). Management task is the management of resources, which gathers the resources to form an activity (UOB-J), records the target (UJT) and processes (UJP) of the activity. RSD can select the resource fulfilling the demands (TCQ) from the resource pool to complete the task. The demand refers to the resource capacity. The condition for performing the task is described by TEC, and the task's description is defined as TP. The task is executed by the resource. Therefore, the tasks and resources produce a series of business activities (UOB-T).

Fig. (**2**) shows that the interaction between the components takes place through the interface, which simplifies constructing the following system architecture on the whole and improves the interaction. Reality and service provided by the components are decoupling, promoting independent resolvability. In order to explain the engineering principles of the design patterns of SaaS software, this paper introduced two symbols into the IDEF3; one is a component customization file (*. XML), and the other is the influence operator. Customization file makes different tenants (RSD and RSP) expression according to their own demands. The influence operator expresses the influence of the resource service quality evaluation (Ev) on the resources and resource selection. Ev is associated with the task (Task and Job). Data is generated dynamically from the two activities (UOB-T and UOB-J), which has an impact on the activities (such as the choice of resource). In addition, it also reflects the influence of the resource model on evolution (such as increasing or decreasing the relevant attributes or methods).

Definition 3: Let R be the resource set which meets the demands of the resource model RM in the cloud resource pool; T be the types of resources and E be the service capacities. The kinds of resources required to complete the task are represented as $R_q=\{r_1,r_2,\ldots,r_i,\ldots,r_n\}$, where the type of resource r_i is represented as t_i, and the capacities required are represented as $E_{qi}=\{e_{i1},e_{i2},\ldots,e_{ik}\}$. Relational operation $\delta_{T=t_i}(R)$ is used for the selection of resource type and the process of filtering the set by service capacity is defined as $\delta_{F_i}(\delta_{T=t_i}(R))$, where

$F_i=(e_{j1}\geq e_{i1})\cap(e_{j2}\geq e_{i12})\cap\ldots\cap(e_{jk}\geq e_{ik})$,

$e_{j1},e_{j2}\ldots e_{jk}\in E_j, E_j\in E$. The resources can be utilized from the cloud resource pool to complete the task:

$$R_p=\bigcup_{i=1}^{n}\delta_{F_i}(\delta_{T=t_i}(R)) \text{ if and only if } R_p\geq R_q \tag{1}$$

Definition 3 is derived from definitions 1 and 2, which highlights one of the conditions to perform the activity UOB-T.UEC, where $\delta_{F_i}(\delta_{T=t_i}(R))\subseteq f_e, \bigcup_{i=1}^{n}\delta_{F_i}(\delta_{T=t_i}(R))\subseteq f_j$. It can be seen by definition 3 that the selection of resource requires matching many capacities of a resource. Thus, the type and capacity conditions cannot be used to filter the resources concurrently in general. Because different types of resources have different service attributes, it is difficult to carry out computing concurrently. This is carried out by the resource model, since different types of resources lead to different models. Thus, it has been proved that the resource model described by definitions 1 and 2 has certain flexibility.

Formula (1) shows that the supply of resource is not exactly the same as the demand of the task. The amount of supply resource must be greater than the amount of demand resource; otherwise the task cannot be completed. Even the condition $R_p \geq R_q$ was met, formula (1) still has the following problems:

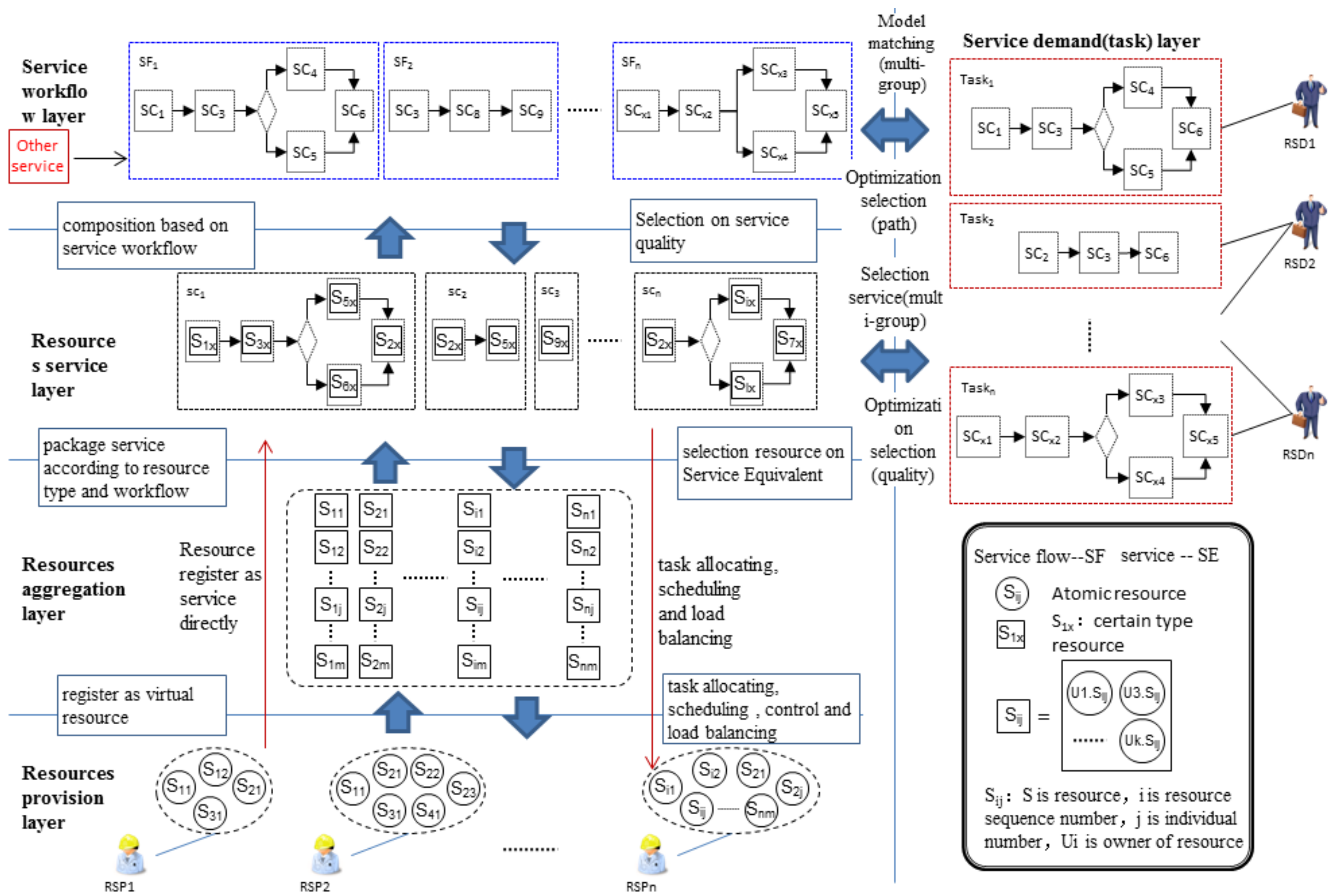

Fig. (3). Resource service model.

(1) The demand and supply of the same resources are not considered in different periods, *i.e.* there is no resource calendar constraint. Therefore, the selection may not complete the task. The result provided by the formula shows a necessary condition to meet the task requirements but does not show all possible conditions.

(2) The quality, efficiency and cost of resources are not considered to perform a task therefore, the result of selection may not be optimal.

(3) It is suitable for a single resource selection, but the efficiency is not high while selecting multiple resources.

If the schedule of task i is c_i, it is easy to obtain the following corollary:

Corollary 1:

$$R_p = \bigcup_{i=1}^{n} \delta_{F_i}(\delta_{T=t_i \cap C=c_i}(R)) \text{ and there must be } R_p \geqslant R_q \qquad (2)$$

The evaluation of the service quality of resource is mainly carried out from the dynamic data of activities UOB-T and UOB-J. Following is the definition :

Definition 4: Let the QoS of resource R-QoS={Q1, Q2,…, Qn} be multi-dimensional. All the evaluation values are obtained from the following formula:

$$R\text{-}QoS = f_{ev}(\sum_{i=1}^{n} UOB - J_i.UJP) + f_{ev}(\sum_{i=1}^{n} UOB - T_i.UBP) \qquad (3)$$

Where, i is the task sequence.

Definition 4 points out that the resource QoS is obtained from the analysis of historical data that highlights the resources which undertake the tasks. It is an objective evaluation method of the resource QoS. Thus evaluation with similar time efficiency, service, cost, reliability, availability *etc.* of each resource can be obtained by this model. When the task is assigned, the resource selection is optimized with formula (2) after filtering. At the same time, it also monitors the QoS. It can help the system find imperfections and correct them in real time which is an important part of the extensible mechanism.

As the above definition shows, although the general resource model is almost the same, different types of resources can have different private resources. The capacity attributes of private resource type reflect the service features of resources. Different resources have different service capacities , which support the operation of different resources. Capacity and private attributes are the static descriptions of resources, which are the basis of the resource selection. With the resource calendar, various tasks are completed according to different business processes. Moreover, all kinds of UOB-T data are produced with a dynamic description of resources. The Ev obtains feedback data through the UOB-T interface to analyze the aspects of the business, from the perspective of optimization scheduling resource by RSD. The UOB-J reflects the autonomy of the resource distribution. RSP

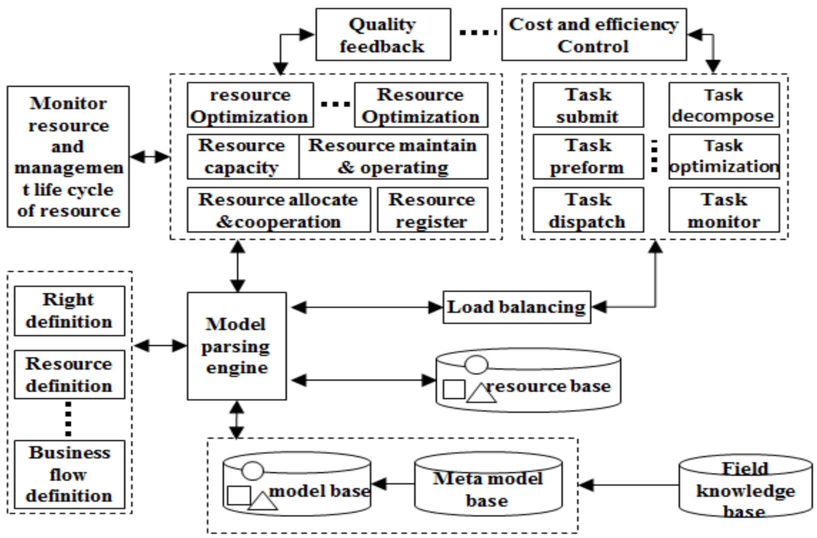

Fig. (4). CARMA.

maintains and manages resources from its own interests, and determines whether to accept the scheduling, and transmit information to Ev management to analyze the perspective of the optimization scheduling resource from RSP through the UOB-T interface.

3.3. Resource Management Architecture

The resource model can extend to Cloud Application Resource Management Architecture (CARMA), as shown in Fig. (**3**). This architecture can approximately be divided into several modules, such as resource management, task management, system maintenance, model parsing engine and load balancers, resource base, mate-model base, model base, domain knowledge base and so on. This paper does not discuss all the modules in detail but only describes some critical aspects due to limited space. The model parsing engine f mentioned in the RMM is the most critical part of the CARMA. The domain experts make all kinds of meta-resource models and save the models on the provisions to the knowledge based domain. The model parsing engine displays these models on RSP(visibility), where the RSP registers and manages the relevant resources, which are parsed and executed by the model's parsing engine. Furthermore, this engine schedules the resources that are chosen and optimized in task assignment and load balancing. The model parsing engine shields the heterogeneity among different resources and different management technologies, in order to carry out customized management of multi-tenant resources and improve the resource aggregation. The model parsing engine can also realize the virtualization resources under the control of resource calendar. The four bases (resource base, mate-model base, model base, domain knowledge base) can enhance the scalability of the system and improve resource aggregation, suggesting that every task can have multiple implementations (reliability). Based on this, the optimization of selection is achieved. CARMA aggregation can make the resource capacity continuous in a wide scope (for example, the load of a truck changes continuously from 0.5 tons to 100 tons), which can simplify the matching algorithm and improve the executable sequence probability of the task matching with resources (availability).

In order to support CARMA, this paper describes the Resource Managing Layers Architecture (RMLA) in detail, which is shown in Fig. (**4**). RMLA is divided into five layers: resource provision, resource aggregation, resource service, service workflow and service demand (task)in an hierarchical fashion.

(1) Provision of resources: It is the layer that executes the tasks, involving physical resources provided by the RSP.

(2) Resources aggregation: The RSP registers resources into the system based on the management strategy of RSP and the integration rules of CARMA. The resources, redistributed by type, make a free and scalable layer of aggregated resources, which can reduce the interference between the selection of services and the difficulty of designing and realizing the relevant algorithm. From the definition of meta-resource models, it is known that registering resources can support large amount of resources without the limitations of region, RSP and RSD's demands, which guarantees enough resource supply during the task assignment process [15].

(3) Resources service: It is the layer of resource combination. It integrates one or more resources with their service characteristics. For example, the collaborative couplings to the forklifts load 1 ton and 90cm up, the trucks height 70cm and load 10 tons and the truck drivers become a group of transport service. It also supports the RSP to package resources as a service, by directly accessing the system through two different system resource layers.

(4) Service workflow: The packaged resource service based on business flow is a more perfect service which includes non-resource service. It is equivalent to knowledge based services.

(5) Service demand (task): The users can divide a task into several subtasks in this layer. On one hand, it can select relevant workflow model from the service workflow layer by model matching. On the other hand, it can also select resource service based on the needs of the tasks.

The selection of resource service is a bottom-up process which involves five layers: task decomposition → workflow/service selection → service binding → resource binding → resource performing. The optimization of service workflow layer includes the optimization of service path and the optimization of service layer ultimately lead to the optimization of selecting resources.

4. RESOURCE SELECTION ON QOS AND SERVICE CAPACITY MATCHING

RMLA not only deals with resource management, but also shows the relation and connection between tasks and resource, including matching models from the service requirement (task) layer to the service process layer, selections from the service requirement layer to the resource service layer, optimal combinations of resources and tasks, the task assignment, resource scheduling and related load balance. This section discusses service management in the field of logistics as an example, and explains the foundation of implementing the above processes which include, matching of tasks and resource service capability along with resource optimization based on service quality.

4.1. Resource Optimization Based on QoS

In RMLA, after filtration based on the matching capacity of resource service by formula (2), many resources satisfying service quality can be elected, but not all of the resources can finally serve as service providers; only some of them are needed for the task. Therefore, optimal group is required to be selected from the resources. Thus, the factors influencing the service quality are needed be determined and multidimensional service quality model is built, as stated in definition 4. Every parameter in the service quality model is derived from the evaluation results of the service usage by resource service evaluation model using formula (3).

For example, a five-dimensional service quality model including time Qti(r, ti), cost Qco (r, co), reliability Qrel(r, rel), availability Qav(r, av) and reputation Qrep(r, rep) is represented as QoS ={Qti (r, ti), Qco (r, co), Qrel(r, rel), Qav(r, av), Qrep(s, rep)}. Formula (3) is used to obtain information from UOB-T and UOB-J in order to provide the basis for resource selection.

In order to determine the computing standard and simplify the computing process, the ideal service equivalent is required; the definition of which is given as follows

Definition 5: Service Equivalent is a quantization standard for service capability, which is the ratio of the capability of certain type of resources in the industry's standard units to the minimal resource capacity in this field, represented as λ.

For example, the loading capacity for a lorry is 10 ton, and for the minimal lorry it is 5 ton. The service equivalent for the lorry should be λ_w=10/5=2. If the ship with a certain type of cargo can load up to a minimum of 20 20GP containers, the service weight of the ship should be $\lambda_{container}$ = $(40\times20)/(20\times20)$. In general, introducing ideal service equivalents can simplify the complexity of calculating service capabilities.

Similar to service equivalent, ordinal utility function is often used for service election in micro-economics, which can provide weak ordinal relations in numerical value and promote rational consumption of services. Constructive Ordinal Utility Function has been used in this study as a scale of service resource to obtain the optimal solution of service resource selection.

Definition 6: In Resource Optimization Model Based On QoS (ROMBOQ), R = R={r_1 ,r_2 ,…, r_n} is the set of candidate service resources obtained by formula (2). Q=$[q_{ij}]_{m\times n}$ is the QoS decision matrix, where q_{ij}=$Q_i(r_j,i)$ is the resource r_j ($r_j\in$R) whose value is obtained from QoS attributes q_i ($q_i\in$QoS), by formula (3) under data normalization. In constructive ordinal utility function $f(r_i)=\sum_{j=1}^{n} w_j q_{ij}$, if and only if $f(r_i)\ge f(r_j)$, the service that resource ri provides is better than r_j; where w_j is the weight of the attribute of service resource, i,j=1,2,...,m.

According ROMBOQ, the following decision model can be obtained:

$$S_{opt}=\min(f(r_1),f(r_2),...,f(r_m))=\sum_{i=1}^{m}\sum_{j=1}^{n}(q_j^*-q_{ij})^2 w^2 \quad (4)$$

$$s.t\begin{cases} \sum_{j=1}^{n} w_j = 1 & (5)\\ w_j\ge 0, j=1,2,...,n & (6)\\ E_q(r,\lambda_{ti})=\sum_{i=1}^{m}E_p(r_i,r_{ti}) & (7)\end{cases}$$

Where $q_j^*=\max(q_{1j},q_{2j},...,q_{mj})$ is the ideal value of attribute qj in the decision matrix. The objective function Sopt restricts the ideal value in the minimal variance with attribute qj for other candidate services. Therefore, the ordinal service resources R satisfy the condition: $\forall r_i,r_j\in R$. If and only if $f(r_i)\ge f(r_j)$, the resource ri is better than rj. E_q denotes the capability requirement and E_q denotes the capability supply. Formulae (4), (5) and (6) show that the resource weight wj is determined by standard deviation. Formula (7) represents that the optimal resource must revolve around the business of services, which not only selects one resource, but there is a strong possibility of a group of resources which can accomplish a certain task will be provided.

4.2. Evaluation of Resource Service Quality

In the whole Cloud, registered resources in different locations form a huge and complex network by their related business. It is important to monitor, estimate and modify the network which is pretty complex. It can help observe the situation of resource service, find the advantage and disadvantage of the algorithm and cope up with the problems to modify the algorithm on time. The evaluation is divided into 3 aspects: the consumption of resource individual service capability; the consumption of the whole resource service capability and the load balancing of resources.

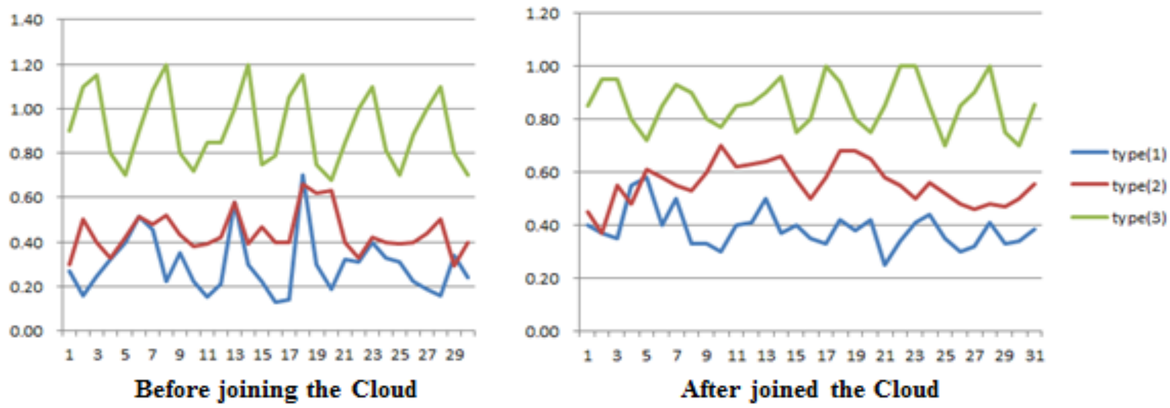

Fig. (5). A longitudinal comparison.

(1) Resource utilization rate: it reflects the task saturation degree for resource in the network, obtained by the following formula:

$$CoS_p = (\sum_{j=1}^{n}(\delta_j \frac{C_{qj-x}(r_i,\lambda_x)}{C_{pj-x}(r_i,\lambda_x)}) \times \frac{t_i}{T} \quad (8)$$

Where $C_{qj-x}(r_i,\lambda_x)$ is the service capability in use and $C_{pj-x}(r_i,\lambda_x)$ is the maximum service capability provided by the resource j. δ_j determines the importance of the service capability in the service process. For example, the regulation rules that the weight of the baggage in airline should not be more than 45kg with the sum of height, width and length less than or equal to 203cm, in reality, the maximum of which will be the limitation and δ_j should be 0 or 1. Formula (8) shows the ratio of service resource supply to the service resource consumption in a certain period. Higher ratio implies higher loading and higher utilization rate of resources . CoS_p is referred as the saturation of resource load. T is the work time designated by RSP, t_i is the practical work done time in T. For instance, if an RSP designates the working time period from 8 am to 6 pm to the resource i while the actual working time is 4 hours, the condition will be ti/T=4/10. It is evident that every RSP sets different working time periods for different resources mentioned in the resource calendars discussed in section 3.2, which shows SaaS characteristic.

(2) Load balancing of resource reflects the equilibrium resource load in the network, which can be obtained from the individual resource service capability, the formula of which is as follows:

$$avgl = (\sum_{i=1}^{n} CoS_{pi}) / n \quad (9)$$

$$B = \sqrt{\sum_{i=1}^{n}(CoS_i - avgl)^2 / n} \quad (10)$$

As shown in formula (9) and (10), the smaller value of B suggests increased load balancing of the resource. It is improper to use absolute resource load as the standard, because the resource load cannot be balanced when the resource has high service equivalent loads. For example, if a 5-ton lorry carries 2-ton cargo and at the same time, a 2-ton lorry also carries 2-ton cargo, it is an unbalanced tasking with serious resource waste. Therefore, individual service saturation is used to calculate the relative load .

5. EXPERIMENTAL METHOD AND RESULTS ANALYSIS

In order to test the validity of the model, BirisCloud of Harbin Institute of Technology was used in this experiment as the base platform, where CARMA was built, RMLA was realized and various resource models were built. After this, three representative enterprises were compared: (1) petty individual-owned transport agent with not more than 5 vehicles of not more than two types; (2) small transport company with 10 to 20 vehicles of not more than 5 types; (3) large transport enterprise with more than 80 vehicles of any type. The emphases of resource management in the 3 types of enterprises are different. Thus, it was permitted that the enterprises can customize and manage the attributes of the general resource type and private resource type, convenient for the cooperation and comparison of enterprises' performances. The capability attributes, UOB-T and UOB-j were set by the industry standard. In this perspective, building RMLA helps the individual management of enterprises in terms of resources and improves the management mode. Eventually, it also enhances the competitive power of enterprises in the market. Data filtering is carried out when all the RSP, RSD and related industries are involved in the platform and passto the stable phase. According to the estimation model mentioned in section 4.2, the experiment and analysis of the results were carried out based on two aspects.

5.1. Resource Utilization

By formula (8), a longitudinal comparison of the 3 types of enterprises was made, as shown in Fig. (**5**). Before joining the Cloud, there were lowest load saturation and stability for the type (1). Otherwise, the load of type (3) was the most stable. The line chart in the left shown in Fig. (**5**) shows that there were large quantities of tasks in type (1), but they could not cooperate with type(3) which lacked the tasks. This verified the problem mentioned in section 1. The right side of Fig. (**6**) shows that the business of every type of enterprise was improved. The state of individual-owned agents was

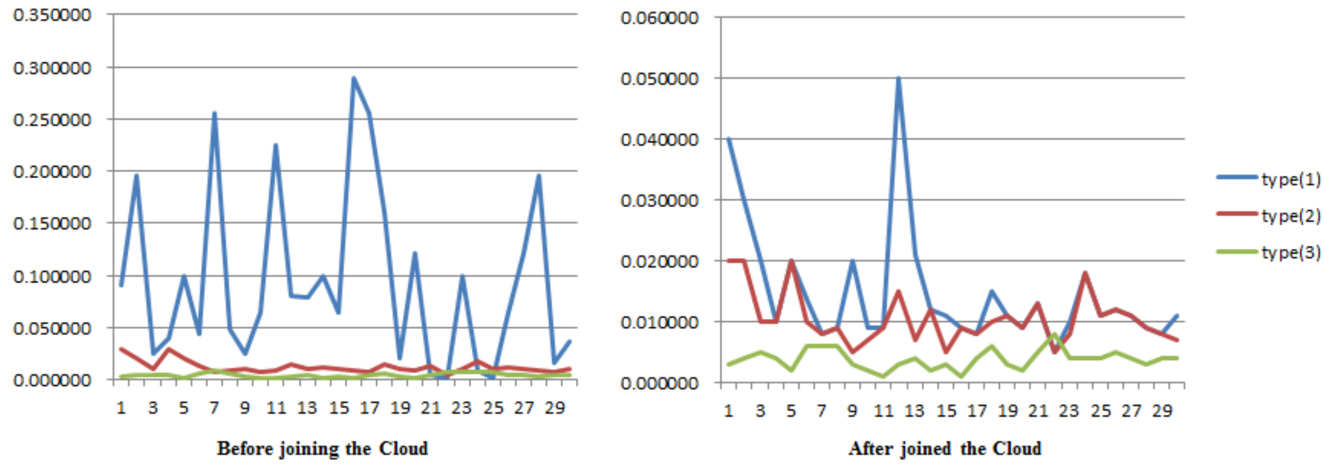

Fig. (6). B of resource load.

observed to be similar to that of small companies and the rate of the load saturation that increased in type (2) became significantly larger than that of type (1), related to the reputation and market recognition.

5.2. Load Balancing of Resource

Due to load saturation used as a parameter, the value of B is always lower than one. Therefore, small change derived from the variance may reflect a great load difference. For example, when the average saturation was 0.3 and the task afforded by a vehicle was 0 in a certain day and the perturbation was only 0.09. Absolutely, this result also showed low utilization of resource . Results were obtained by service tracking and data analysis, as shown in Fig. (**6**) . Due to the standard scheduling policy for resources, employers and tasks in the type (2) and (3) enterprises, the load was far more balanced than that in the type 1. After joining the Cloud, the load balancing of type (1) improved a lot(paying attention to different numerical ranges between the left and the right in Fig. (**6**). It also indicated improved management mode of medium and small scale enterprises.

CONCLUSION

This paper focused on the characteristics of resource management in Cloud and built the resource model. Based on this , a resource management architecture was constructed. Meanwhile, this paper proposed the evaluation method of resource management. Following this, the solution was implemented by BirisCloud platform the advantages of which were reflected during the application process such as: 1) Openness and extendibility. It mainly reflected the ability of customizing and adding new resources, resource service capability and quality with no change in the model architecture and maintaining the heterogeneity of the resources; 2) The solution combined self-management and unified scheduling and was able to monitor and dynamically schedule the resource and balance load, assure highly efficient operation and safety maintenance, and reduce the risky business of enterprise. 3) It also significantly improved the resource utilization. The solution adjusted the supply and demand relations between service consumers and service providers, improved the quality of service and promoted the cooperation among enterprises, thus, raising the competitive power of enterprises. The practical work proved that the proposed models and architecture are feasible which can achieve the desired results .

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

The national science and technology supports funding of the project (2012BAF12B16), the national natural science foundation of China (61273038), the scientific and technological torch-plan project of Shandong Province (2010GZX20126).

REFERENCES

[1] P. Mell, and T. Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards and Technology: US, 2011.

[2] F. Doelitzscher, A. Sulistio, C. Reich, H. Kuijs, and D. Wolf, "Private cloud for collaboration and e-learning services: from IaaS to SaaS," *Computing*, vol.91, no.1, pp. 23-42, 2011.

[3] H. Jin, H. H. Chen, Z. P. Lu, and X. M. Ning, "Q-SAC: towards QoS optimized service automatic composition" In: *Proceedings of the 5th IEEE/ACM International Symposium on Cluster, Computer and the Grid (CCGRID)*, vol. 2, pp. 623-630, 2005.

[4] P. Li, L. Zhang, S. Wang, F. Tao, J. Cao, X. Jiang, X. Song, and X. Chai, "Cloud manufacturing: a new service-oriented networked manufacturing model," *Computer Integrated Manufacturing Systems*, vol. 16, no. 1, pp. 1-7, 2010.

[5] P. Li, L. Zhang, L. Ren, D. Chai, F. Tao, Y. Luo, Y. Wang, and C. Yin, "Further discussion on cloud manufacturing," *Computer Integrated Manufacturing Systems*, vol. 17, no. 3, pp. 449-457, 2011.

[6] L. Liu, T. Yu, and Z. Shi, "Research on QoS-based resource scheduling in manufacturing grid," *Computer Integrated Manufacturing Systems*, vol. 11, no. 4, pp. 475-480, 2005.

[7] F. Casati, S. Ilnicki, L.J. Jin, V. Krishnamoorthy, and M.C. Shan, "*eFlow: A Platform for Developing and Managing Composition e-Services*," Technical Report, HPL-2000-36, HP Laboratories Palo Alto, 2000.

[8] S. L. Liu, Y. X. Liu, F. Zhang, G. F. Tang, and N. Jing, "A dynamic Web services selection algorithm with QoS global optimal in Web services composition," *Journal of Software*, vol. 18, no. 3, pp. 646-656, 2007.

[9] T. Yu, Y. Zhang, and K.J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," *ACM Transactions on the Web*, vol. 1, no. 1, p. 6, 2007.

[10] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, *A Resource Management Architecture for Metacomputing Systems*, *Job Scheduling Strategies for Parallel Processing*. Springer: Berlin, Heidelberg, pp. 62-82, 1998.

[11] F. Tao, Y. Hu, D. Zhao, and Z. Zhou, "Study on resource service match and search in manufacturing grid system," *The International Journal of Advanced Manufacturing Technology*, vol. 43, no. 3-4, pp. 379-399, 2009.

[12] R. Raman, M. Livny, and M. Solomon, "Matchmaking: distributed resource management for high throughput computing, High performance distributed computing", In: *Proceedings The 7th International Symposium on IEEE*, pp. 140-146, 1998.

[13] R. Buyya, "*Economic-Based Distributed Resource Management and Scheduling for Grid Computing*," arXiv preprint cs/0204048, 2002.

[14] R. T. Fielding, *Architectural Styles and the Design of Network-Based Software Architectures*, University of California: Irvine, 2000.

[15] X. Xu, "From cloud computing to cloud manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 1, pp. 75-86, 2012.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014