A Community Cloud Platform Model Supporting Cross-Domain Business Collaboration

Wei Guo, Xinjun Wang, and Lizhen Cui

***Abstract*—Cloud computing is becoming a new application model for small and medium-sized enterprises to share hardware and software resources on-demand over the Internet. Despite the flourish of the cloud computing, there are many challenges, including establishing a model to make the cross-domain business collaboration between enterprises possible. In this paper, we present the concept of the community cloud, in which, resources can be integrated as services, for the community cloud members to collaborate with each other flexibly and quickly. We design a community cloud platform (CCP) model to support cross-domain business collaboration. We subsequently specify the formal definition of the functional components in the community cloud platform and give a detail analysis of the business collaboration engine of the platform.**

***Index Terms*—Business collaboration, cloud computing, community cloud.**

I. Introduction

With the continuous development and popularization of information technology, the traditional information construction mode has been unable to meet the requirements of the information technology, due to the needs for the larger infrastructural investment, the longer application development cycle and the higher operating maintenance cost. Cloud computing [1] platform is becoming a new platform for enterprises. The mode of software as a service (SaaS) application, is being adopted by more and more enterprises, especially small and medium-sized enterprises, for it is easy to allow enterprises and individuals to demand pay-as-you-go, to expand the consumption by using a variety of hardware platforms, software, and other various resources, to avoid the maintenance costs of IT infrastructure and enormous capital investment.

Business collaborations between enterprises have become increasingly frequent. These businesses generally can't be completed within one department, but need more departments to do in collaboration. They are often inter-departmental, even cross-regional. We take the business in the social insurance area for example. By now, each municipality has basically established a social insurance cloud computing application platform within its own region, which provides social insurance related business services for the insured organizations and persons. The cross-regional business collaboration is not solved well enough, such as cross-regional laboring, cross-regional social insurance relationship transferring, cross-regional medical insurance payment services, etc. These business collaborations involved multiple regions or departments are in manual or semi-manual, obviously, it is obviously unsatisfactory to the public.

The following Fig. is the business process of a cross-regional social insurance relationship transferring. It means that one insurant is to transfer his social insurance relations in the city A to city B, and to pay social insurance fee in the city B. Under such circumstances he needs to follow the operations as shown in steps.

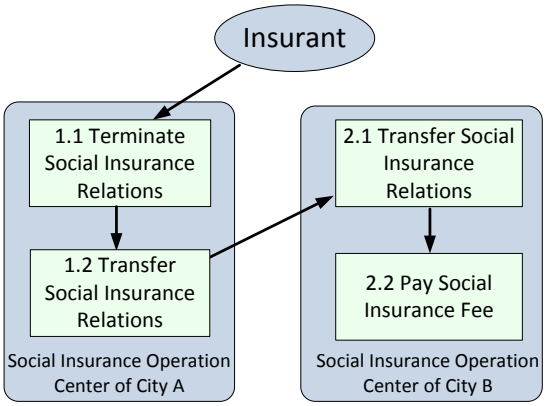

Fig. 1. An example of cross-regional social insurance relationship transferring.

In this mode of business processes, the insurant is doomed to have a poor experience, for cumbersome business operations. The insurant needs to go to city A and city B respectively to transact business, obviously, it deems to be inconvenient. We need to find a solution that allows the insurant to directly transact business in the social insurance operation center of city B, instead of going to city A. In other words, we should handle the step 2.2 directly, omitting the step 1.1, 1.2, and 2.1.

Rest of the paper is organized as follows. The next section discusses related works. The following section 3 is the structure and the definition of the community cloud platform model. Section 4 is the business collaboration engine framework. The last section describes a summary of this article and the next work outlook.

II. Related Work

Marinos et al, who have researched on the basis of the grid computing, the digital ecosystem, the autonomic computing, and the green computing, first propose the concept of the community cloud[2], [3], and they analyze the infrastructure of the community cloud. They consider that the community cloud is the infrastructure community network formed from a number of cloud providers, in order to provide the infrastructure to cloud providers.

Manuscript received July 17, 2012; revised September 12, 2012.

The authors are with the Shandong Provincial Key Laboratory of Software Engineering, School of Computer Science and Technology, Shandong University, China (e-mail: guowei@sdu.edu.cn.).

Chen Zhang et al design CloudWF [4], which is a scalable Hadoop-based cloud computing workflow system. They develop a simple workflow description language, to encode workflow block and to separate the dependencies between the blocks, just like independent executable components.

[5] develops a service coordination scenario based on the distributed collaboration model, which is a kind of top-down approach. This scenario describes the centralized service collaboration processes, and then maps global service collaboration processes into each participating roles, forming a distributed collaboration sub-processes. Unfortunately, this scenario doesn't give the method of how to build a business process. [6] describes a bottom-up multi-organizational workflow collaborative approach — CoopFlow, which is divided into three steps: workflow abstraction and advertisement, workflow matching and interconnection, and workflow cooperation.

III. Community Cloud Platform Model

A. Platform Model

On the basis of the analysis of characteristics of the cross-domain business workflow and its data distribution, we develop and design the community cloud platform (CCP) model. The overall framework of the CCP model is shown in Fig. 2, including domain service container, service registry, metadata management, business collaboration engine, service management, load balancing management, exception management. These modules are not only able to independently complete the specific task, but also to communicate with each other and work together.

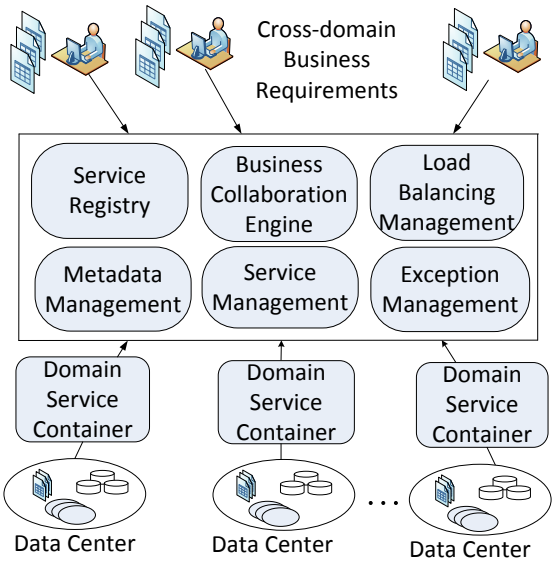

Fig. 2. The community cloud platform model.

The existing information systems of each organization are still running in the data center, as data center constitutes an autonomous domain, with internal business running in the autonomous domain. As for the cross-domain business requirements, we use the CCP model to complete the cross-domain business collaborations.

The platform resolves the data management issues in the CCP model environment on the base of self-management and self-tuning mechanism on the analysis data, by building a large number of node-based index structures, to deal with resource scheduling and load balancing. In additional, this platform establishes a unified transmission interchange format using by platform and domain services containers in the form of XML.

B. Platform Components

Next, we will interpret the components of the CCP model in detail:

Domain Service Container: The query business requirements for cross-domain will be pushed down from the platform to domain service containers deployed on data centers. Simultaneously, the operational business requirements for cross-domain will be scheduled by CCP, running in domain service containers.

Service Registry The cross-domain business requirements should be abstracted to CCP, to do services virtualization, service registration, service authorization, service release, etc, and to define protocol at the application level based on P2P basic discovery protocol and the heartbeat mechanism. These protocols include service node registration, response, broadcast, qualification, etc. The platform also defines the transfer interchange format between nodes.

Metadata Management: The CCP model uses the metadata management component to manage the metadata and to dynamically generate metadata in community cloud computing platform. By extracting, storing, updating, and managing metadata, the platform manages the data from the scattered data centers. The metadata's magnitude is very large, so they need stored more than one copy. Based on this, CCP model references the master-slaver deployment mode. The metadata of community cloud computing platform can be summarized into four categories: data center identification metadata, application-level metadata, index metadata, and business metadata.

Business Collaboration Engine: The cross-domain business collaborations in CCP model cover all aspects of the overall life cycle of business processes, including the individual requirements achieved, processes automatically generated, process instances scheduled, and users interacted. The business collaboration engine can achieve the unified scheduling of tasks, monitoring of services, and routing of messages, and can coordinate the domain service containers deployed on data centers to complete business collaboration. The execution of business collaboration processes is accomplished by a number of different members of the community cloud, in accordance with the order of the processes. The execution request is firstly captured by the management node of the community cloud member, which creates the collaboration service. After completing scheduling of tasks in the members, the remaining fragments of the process business will be delivered to the other domain service containers through business collaboration engine. At last, the entire cross-domain business is completed ultimately.

Service Management: The CCP model offers self-assembling services, services pushdown, collaboration of services MapReduce, services catalog management, etc. This new community cloud computing model allows service providers to participate in collaborative free to join and quit, at the same time, more rigorous modeling of the traditional process applications, task-based management activities are

weakened or eliminated. Cross-domain business collaborations in community cloud computing environment need to be quickly searched and found to ensure the timeliness of the services.

Load Balancing Management: The CCP model has the ability to provide real-time load balancing. Firstly, the platform provides the reliable collaboration services, for the CCP model to play the advantages of multi-organization collaborations. The business collaboration strategy in the community cloud has the ability to solve the distributed resource load balancing, and to give full play to the community cloud computing model, service execution location is completely transparent to the users. In other words, the users can enjoy the highest quality services anytime, anywhere. Secondly, the CCP model enabled load balancing can maximize the use of system resources for implementation, which is of great significance to enhance overall system performance.

Exception Management: The CCP model consists of multiple data nodes, using favorable data layout strategies to deal effectively with single point of failure. In the community cloud computing environment, the CCP model can handle the exceptions, such as service status changed, service nodes failure, etc.

C. Platform Definition

We can formally define the CCP model as follows:

Definition 1: The community cloud platform model is defined as a 6-tuple :

CCP = (DC,DSC,SM, MDM, BCE, TS),

Among them,DC is the autonomy of the cloud organization domain. DSC is the set of domain service containers deployed on the autonomous domains. SM=(S-Register, S-Logout, S-Start, S-Stop, S-Search)，SM is the service manager, which is the management tool for CCP services, consisted of service registering (S-Register), service logouting (S-Logout), service starting (S-Start), service stopping (S-Stop),and service searching (S-Search). This tool can help the CCP to filter malicious services, and to protect the services in the CCP. MDM is the metadata management, which adopts the manner of the distributed redundant storage, to solve the problem of the metadata server a single point of failure, to ensure high availability of metadata. The performance of the CCP will not be weaker after frequent operations, for the CCP having used Chord [7] technology to achieve a fully distributed metadata management. BCE is the business collaboration engine. TS is the task scheduler of the CPP, which is responsible for load balancing, and handles the abnormal transactions, by properly scheduling the tasks in the CCP. The task scheduler is a platform for the implementation of the tasks, which is a key link for tasks and threads, whose main duty is responsible for assigning tasks to the appropriate threads to deal with.

Definition 2: The service in the community cloud platform model is defined as a 4-tuple:

Service = (Type, BusinessLogic, Data, Rule)

The Type refers to the type of service. When the CCPdefines the constraints of business collaborations, it can finalize what type of service to collaborate using the Type. The BusinessLogic denotes the business logic of service. The Data refers to the data which the service operates. Data = (Table, Item), the Table indicates the data table, and the Item denotes data fields. The Rule denotes the constraints of the business collaborations.

Definition 3: The process in the community cloud platform model is defined as a 3-tuple:

BP = (SS, BPL, PT)

BP denotes business processes based on business collaborations. SS is the service set which is selected by the business processes, while the formal definition of the service has been given in Definition 2. The BPL means the business process logics, and PT denotes the business process templates. BPL = (BA, SA, DE), BA denotes the basic activities of the business processes, such as service calling and variant assigning. SA denotes the structural activities of the business process, such as loop and conditional choice. DE denotes the relationship of dependency on business processes, which can specify the order of execution of the business processes. BA = (IN, OUT), IN represents the input of the business, and OUT the output of the business.

IV. Business Collaboration Engine

A. Node Deployment

The CCP deploys multiple service nodes and at least one management node. The service node is the abstract representation of the network resources, which is used to execute the services of the members of the community cloud, such as web services of the members of the community cloud and servers. The service node is independently managed and maintained by the members of the community cloud, and it is the final executor of collaborative services. It achieves the service requirements and deals with the input data to return the implementation results. The management node is the network node which is responsible for assigning the tasks to the service nodes. By setting up such node within each member of the community cloud, the inner services in the members of the community cloud can be connected with the upper network of the CCP model as a whole, to participate in the implementation of the CCP model. This can ensure that the resources of the members of the community cloud are managed independently, and can ensure the efficient execution of collaborative business.

The CCP deployment structure consists of three components: the service catalog (SC) on the service directory node, the task execution interface (TEI) on the service node and the business collaboration engine (BCE) on the management node.

B. Business Collaboration Engine Architecture

Business collaboration engine is a complex framework, using hierarchical architecture design to reduce the degree of coupling of the internal components, as far as possible to separate engine core from other business functions, so that each layer of the engine have explicit functions and responsibilities.

The architecture of the CCP business collaboration engine is shown in Fig. 3.

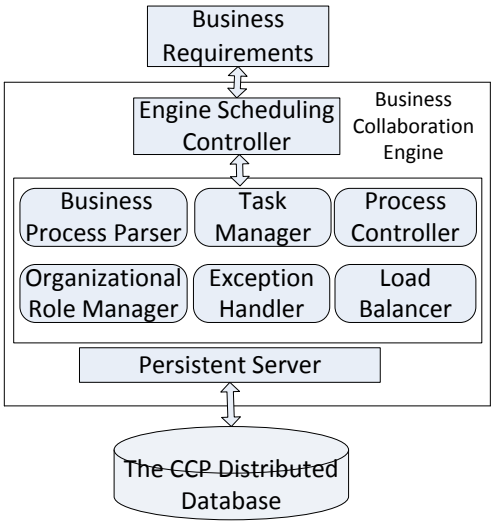

Fig. 3. The architecture of the ccp business collaboration engine.

The kernel of business collaboration engine is the engine scheduling control center, which needs to implement the routing capabilities of the engine, to ensure the safe running of the node. The scheduling center needs to access the function modules, such as process parser, task management, workflow control, organizational role management, exception handling, etc, to implement the data storage of the engine periphery.

The design of the business collaboration engine considers the following aspects mainly:

Business Process Parsing: The business collaboration engine provides the environment to maintain the collaborative operations of the business processes, and the running of the business processes depends on the process definition, therefore the engine core must parse the business process definition, and establish a mapping between the business process definition and business process instance.

Business Process Scheduling: While the business collaboration needs to provide routing control functions, the scheduling rules and routes of the business processes are controlled by the current business process instance. A business process corresponds to a business process definition, and then the business collaboration engine should explain the scheduling rules to ensure that the complex collaborative business processes can be handled, locating the next flow of the business processes, and scheduling the business processes continuously.

Business Process Execution: The CCP is a platform centric of the business collaboration processes, and all business operations conducted in accordance with the certain scheduling rules. As for a business process, each logical unit represents actual business logic. When the business process runs to a node, a set of enforcement mechanisms are needed to determine how to implement task in the node. During the lifetime of the business process instance, the state is an important control data, with changing the state of current business process instance to control the work state of the platform. The responsibility of the business collaboration engine is to control the state transition in accordance with the defined workflow scheduling rules.

V. Summary and Outlook

Cloud computing platform is becoming a new platform for enterprises, and the establishment of the community cloud application model and related systems provides a platform for cross-domain business collaboration between enterprises. Resources within the community cloud can integrate flexibly in the form of services, and the members of the community cloud can collaborate with each other quickly. This paper designs a community cloud computing platform model supporting cross-domain business collaboration, and discuss the components of the platform in detail. In the real community cloud computing environment, there are more complex issues need to be considered, such as how to effectively solve the free features of the service node and the credibility of nodes.

This work has been supported by the National Natural Sci ence Foundation of China under Grant No. 61003253; the N atural Science Foundation of Shandong Province of China u nder Grant No. ZR2010FM031, ZR2010FQ010, the Key Te chnology R&D Program of Shandong Province under Grant No.2010GGX10107, 2010GGX10114.

References

[1] 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, no.6, pp. 599-616, 2009.

[2] A. Marinos and G. Briscoe, "Community cloud computing," *Lecture Notes in Computer Science*, vol. 5931/2009, pp. 472-484, 2009.

[3] A. Marinos and G. Briscoe, "Digital ecosystems in the clouds: Towards community cloud computing," *3rd IEEE International Conference on Digital Ecosystems and Technologies* (DEST 2009), New York, USA, pp. 103-108, 2009

[4] C. Zhang and H. D. Sterck, "Cloud WF: A computational workflow system for clouds based on hadoop," *CloudCom 2009*, LNCS 5931, pp. 393–404, 2009.

[5] X.-Q. Qiao, J. Wei, and T. Huang, "Service collaboration approach based on decentralized mediation model," *Journal of Software*, vol. 20, no. 6, June 2009, pp. 1470-1486.

[6] S. Tata and K. Klai, "Nomane Ould Ahmed M' bareck. Coop flow: A bottom-up approach to workflow cooperation for short-term virtual enterprises," *IEEE Transactions on Services Computing*, vol. 1, no. 4, October-December 2008.

[7] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *SIGCOMM'01*, August 27-31, 2001, San Diego, California, USA.