Mobile Agent Based Open Cloud Computing

Federation and Task Management

[1]Suraj Kumar Thakur, [2]Mohit Kumar Maroti
[1]B. Tech - Information Technology, [2]B.Tech - Computer Science Engineering
SRM University, Kattankulathur, Tamil Nadu, India- 603203

***Abstract* -- "Cloud computing" is the next natural step in the evolution of on-demand information technology services and product.It is expected to provide quick, agile, stable, reliable services. Even though cloud has simplified cloudnetwork architecture, there is some complexity. Some of them are portability and interoperability between different Cloud Computing Services Providers. These problems handicap the widely deployment and quick development of cloudComputing. As a solution to these problems we put forward an inevitable approach "Open Cloud Computing Federation "for the wide use of cloud computing and to realize the value of it. In this paper we propose the MABOCCF mechanism. It will combinethe advantages ofMobile Agents andCloud Computing.ThusMABOCCFcan span over multiple heterogeneous Cloud Computing Platforms and realizes portability and interloper ability.In this paper we also present how cloud schedules user jobs and motivation for the combination of Mobile Agents and Cloud Computing.**

***Index Terms* - Cloud Computing, Simplified Network, Federation, Mobile Agent Portability, Interoperability**

I. INTRODUCTION

Cloud computing is the next natural step in the evolution of on demand information technology services and products .To a large extend cloud computing is based on virtualization resources. Some of them might have questions like "what is cloud computing, and what does it mean for my business?"Cloud computing platforms are growing very quickly. Let's explore the cloud computing infrastructure and its impact on critically important areas to IT, like security, infrastructure investments, business application development, and more.

To make cloud computing work, what you need are three things: thin clients (or clients with a thick-thin switch), grid computing, and utility computing. Grid computing links separate computers to make a large infrastructure, use idle resources. Utility computing is paying for what you use on shared servers like you pay for a public utility (such as electricity, gas, and so on).

Evolution of Cloud Computing

The evolution of sharing on the Cloud went through: Networking, network sharing, information sharing, resources sharing, and services sharing. The first stage of the Cloud was around networking, the TCP/IP abstraction. Their inter-networking with TCP/IP led to network sharing and the emergence of the Internet and its worldwide adoption. The second stage of the Cloud was around documents, the WWW data abstraction. The HTML format, the HTTP protocol, and the Mosaic browser were adopted by universities for document exchange and then worldwide for information sharing. Then, grid computing emerged with the creation of standards and software for remote resources sharing and collaboration, exclusively utilized for highly scalable High Performance Computing (HPC) jobs. The newest stage of the Cloud, cloud computing, has emerged to provide service sharing by abstracting infrastructure complexities of servers, applications, data, and heterogeneous platforms.

Problems in Cloud Computing

However, cloud computing technology is still in its infancy, although there have been a lots of cloud computing platforms which provide by Google, Amazon, IBM, Microsoft, Yahoo, Sun and Salesforce.com, and so forth, most of the customers are reluctant to transform their current datacenters and IT resources to the cloud computing platforms, because a number of unsolved technical problems still exist for these cloud platforms, and Open Cloud Computing Federation (abbreviated as OCCF later) may be an answer to many of these problems.

II. OCCF (OPEN CLOUD COMPUTING FEDERATION) AND IT'S ADVANTAGES

Many researchers from industries and academics had proposed the concept of Open Cloud Computing Federation. It incorporates multiple CCSP's (Cloud Computing Service Provider) service to provide a uniform resource interface for the user. We think that OCCF is the only way to get the cloud computingwidely used and realize the greatest value of it. Most of the problems of cloud computing technology today are attributed to the lack of portability (user's application can spans over multiple CCSPs) and interoperability (user's application can deploy on multiple CCSPs), with reference to some viewpoints on cloud computing in the whitepaper from University of California, Berkeley [5], we analyze theproblems caused by the two defects and the count measure to each of the problems after the adoption of OCCF

1. ***Limited scalability***. Most of the CCSP claim that they provide infinite scalability for the customer, actually, with the wide

use of cloud computing and the rapid growth of the users, none of the CCSPs can meet all the requirements of all the users, only an OCCF (open cloud computing federation) can provide a real infinite computing resources for the customer.

2. ***Unreliable*** availability of a service. Actually, shutdown events happened recently with many of the of the CCSP's cloud computing service, including Amazon, Google and Microsoft. If you use only one CCSP's service and your application can't migrate to another CCSP, once your CCSP breaks down, your service will disappear in the air since it can't be found on the network any more. On the contrary, in OCCF, user's application can migrate to another CCSP in case of breakdown, and multiple CCSP's service can be used simultaneously, offer a good availability of a service.
3. ***Service provider lock-in***. It is impossible for you to move your data and application from one CCSP to another for some special considerations, because of the absence of the portability: consequently you are locked to a certain CCSP. An OCCF will demonstrate the whole Cloud Computing market where the small scale competitors can enter and thus promote invocation and vitality.
4. ***Unable to deploy service over the multiple CCSPs.*** Currently, application cannot scale over multiple CCSPs since there is no interoperability between CCSPs. OOCF can make it possible for the organization to integrate different CCSPs service to provide more valuable service for the end users.
5. ***High TCO (Total Cost of Ownership) for user.*** One of cloud computing's value is brought by the scale economy which is caused by the share and reuse of resources by lots of users, so the user needn't prepare large quantity of equipment for the peak demand, similarly, if the CCSPs want to get high utilization ratio of their facilities, and not prepare lots of extra expensive resource for the peak request, driven by cost, OCCF is certainly the best choice for the CCSP to provide a high cost performance service for the customer.

III. MABOCCF (MOBILE AGENTBASED OPENCLOUDCOMPUTING FEDERATION)

After more than a decade of study, mobile agent has got a lot of breakthroughs in many key technologies, but it suffers from that and it can't find an appropriate platform in a large scale network to run and manifest its expected advantages. Cloud computing technology provides a good chance for mobile agent to display its capability; we think cloud is the best platform for mobile agents so far.

Mobile agent based cloud computing is feasible because most of mobile agent systems are based on or support Java, such as Aglets and D' Agent, many cloud computing platforms currently provide virtual machines which support different kinds of OSs (operating systems) such as Linux and windows, since Java can "write once, run anywhere", so the mobile agents can run on the JVMs (Java Virtual Machine) which install on these OSs. We proposal a new mechanism call MABOCCF (Mobile Agent Based Open Cloud Computing Federation) which combine mobile agent with cloud computing to construct a cloud computing federation mechanism, this mechanism can realize portability and interoperability between different kind of cloud computing platforms.

MABOCCF Architecture and Components

In order to offer better compatibility, we introduce a Travelling Bag mechanism in which application code or user's tasks (both call as task for short) are encapsulated in a mobile agent, the mobile agent run on a MAP (Mobile Agent Place, one virtual machine can have more than one MAP) on the virtual machines which provide by the CCSPs (cloud computing service provider), mobile agent (carry the application code or user's tasks) can move from one MAP to another MAP, realize portability among deferential CCSPs even though they are heterogeneous. At the same time interoperability is realized by the negotiation and collaboration among agents by agent interoperability standards such as MASIF (Mobile Agent System Interoperability Facility) and FIPA (Foundation for Intelligent Physical Agents).

Because the realization of portability and interoperability, make it possible for the implementation of OCCF (Open Cloud Computing Federation), the architecture of MABOCCF (Mobile Agent Based Open Cloud Computing Federation) is shown in Fig 1. A JVM (Java Virtual Machine) and a MAP (Mobile Agent Place) is installed on every virtual machine in the CCR (Cloud Computing Region, a CCSP or an administrative domain in one CCSP), this process can be done automatically. A virtual machine or a physical machine is chosen to act as TS (Task Manager, there can be more than one TS in a CCR); it's the region access point of the CCR; a MAP registers itself to the TS and frequently renews its situation (the information of the mobile agents reside on it and the situation of the resources on it) to the TS. All newly created mobile agents are sent to the MAPs in the CCR, then the MAPs register the information of the mobile agents (reside in the MAP) to the Task Manager, or unregister it in case of the mobile agent left the MAP, MAP is also responsible for the backup and monitors the mobile agent on it. Furthermore, many services including resource indexing, authentication, security, billing, disaster recovery and fault tolerance are provided by the TS. The MAPs interact and interchange information (the information of the register mobile agent and the situation of its resources) with the TS frequently.

At the user end, task is encapsulated in the mobile agent, or let the user custom the mobile agent which isprovided by the third party, the requirements of the resource (the demand for software, hardware and type of data resources) is written in the head of the mobile agent, the data structure of the mobile agent is shown below. Then, the mobile agent is sent to the TS in certain CCR (Cloud Computing Region), The TS (Task Manager) will send the mobile agent to a MAP (Mobile Agent Place), finally, the mobile agent will execute on a MAP to complete its mission.

IV. MECHANISM OF MOBILE AGENT BASED OPEN CLOUD COMPUTING FEDERATION

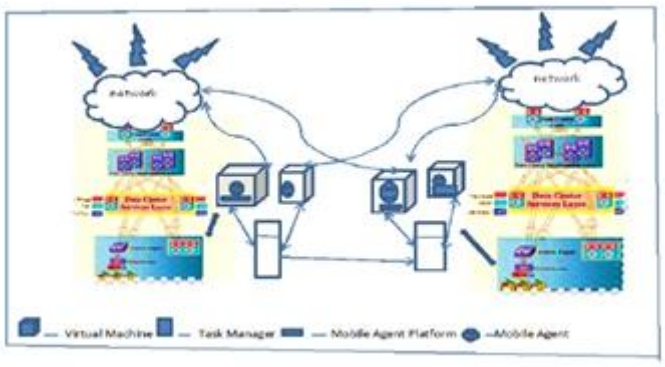

Figure 1 The MABOCCF mechanism

The mechanism of MABOCCF is shown in Fig.1, the user sends a mobile agent to a TS, the TS reads the head of the mobile agent to judge if it's a mobile agent or other kind of data packages, then the TS matches the requirements with its resource index to decide which MAP the mobile agent should be sent to, or assign a new virtual machine with a MAP for this mobile agent. Once the MAP receives the mobile agent, it activates the mobile agent and executes the task including in the mobile agent, the assignment begins to execute in the virtual machine. The mobile agent monitors the execution of the task and the situation of the resources in the MAP, decide whether to leave the MAP, or clone some new mobile agent and send them to other MAPs (in the same CCSP or in a different CCSP) to accomplish the task.

A mobile agent can carry the task and move among MAPs in a CCR, or MAPs distributing over many different CCRs, realize the computing portability. Mobile agents can negotiate and collaborate with each other through communication, to realize the interoperability among different CCSPs.

A TS can map other CCRs' resources (software, hardware, and data) to its CCR according to the local manage regulations (instituted by the goals of the TS, or the price, QOS and reputation of the resources, etc.), as if they are local resources, or send the mobile agent to an optimistic differentiated service job schedulingother CCRs when the resources of local CCR is scarce. Thus this mechanism can provide a really "infinite" resource pool for the user, realize a high scalability of the cloud computing resources.

In a simple case, user's task is assigned to one or many mobile agents, if there are many mobile agents, they don't interact with each other during the execution of the task, the TS receives the mobile agent and sends it to a MAP for execution, the mobile agent can migrate from one MAP to another during its lifecycle, the result is sent back directly to the user or delivered to the user by the TS.

A more complex case is that the task is assigned to several mobile agents, and the mobile agents collaborateswith one another to accomplish the task, these mobile agent can move among different MAPs while keep the collaboration.

The most complex case is that many tasks are assigned to many mobile agents; these mobile agents should negotiate, collaborate and even compete to meet the requirements of its owner, for example, in the case of e-business.

The move of mobile agent could be active or passive; the mobile agent is forced to move to another MAP when the virtual machine which it resides in has malfunction or performance degradation.

V. TASK MANAGEMENT OF CLOUD COMPUTING BASED ON MABOCCF:

Task management in cloud computing is one of the core and challenging issues. Job scheduling (JS) system is one of the core and challenging issues in a Cloud Computing system. Traditional job scheduling systems in Cloud (or Grid) computing only consider how to meet the QoS requirements for the resources users, they seldom consider how to make the maximum profits for the resource providers. Actually, a job scheduling system plays a very important role in how to meet Cloud computing users' job QoS requirements and use the Cloud resources efficiently in an economic way. Usually, from the Cloud computing resources users' sides (we use CCU stands for Cloud computing user), users always think which Cloud computing resource can meet their job QoS requirements for computing (such as the due time of job finishing, the computing capacity etc.), how much money they must pay for the Cloud Computing resources. While, from the Cloud Computing service providers' (we use CCSP stand for Cloud Computing Service Provider) side, the CCSP always thinks how they can gain the maximum profits by offering Cloud Computing resources, apart from meeting the CCU's job QoS requirements. To make these two ends meet, the job scheduling system must take efficiency and economic strategies for CCU's differentiated service QoSrequirements. Focus on this issue, this paper puts forward System for CCSP and CCU.

VI. PRIORITY SCHEDULING SYSTEM IN CLOUDCOMPUTING ENVIRONMENT

From a systemic viewpoint of a Cloud Computing environment, we can take a Cloud Computing environment as a very powerful server. This server will handle the CCU's jobs (Fig 2). For each CCU may have different QoS requirements, usually, CCU's jobs have different priorities to be processed. So we can classify the jobs priorities into several classes.

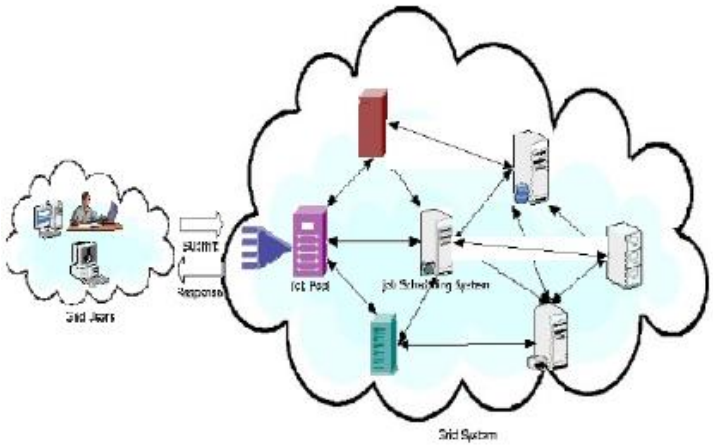

Figure 2 An Illustration for Cloud Job Scheduling

VII. CONCLUSION

Cloud Computing has become one of the hottest topic in the recent years. The simplified cloud computing network architecture aims to achieve the capability of security, agility, consolidation and energy efficiency. We think that OCCF is the only way to get the Cloud Computing be widely used and realize great value of it. We also propose new mechanism MABOCCF to realize portability and interpretability between different kinds of cloud platform and by introducing priority scheduling in task management we can maximize the profit for cloud computing service providers and also meet the CCU's job requirements.

REFERENCES

[1] A. Feldmann, "Internet Clean-Slate Design: What and Why?" Computer Communication Review, ACM SIGCOMM, vol. 37, no. 3, pp. 59-64, 2007.
[2] AKARI Architecture Design Project, "New Generation Network Architecture—AKARI Conceptual Design (Ver. 2.0)" 2007.
[3] ISO/IEC JTC 1/SC 6, "Working Draft Technical Report Text for JTC 1/SC 6 Future Network: Problem Statement and Requirements," 2009.
[4] The OpenFlowSwitch Consortium. http://www.openflowswitch.org.
[5] C. Jelger, C. Tschudin, S. Schmid, and G. Leduc, "Basic Abstractions for an Autonomic Network Architecture," in Proceedings of AOC'07, June 2007, Helsinki, Finland.
[6] D. Farinacci, V. Fuller, and D. Oran. "Locator/id separation protocol (LISP). Internet Draft," January 2007. Available online at: www.ietf.org/internet-drafts/draft- farinacci-lisp-00.txt.
[7] V. P. Kafle, K. Nakauchi, and M. Inoue, "Generic identifiers for ID/locator split internetworking," Proc. ITU-T Kaleidoscope event, Innovations in NGN - Future Network and Services, Geneva, Switzerland, May 2008.
[8] General requirements for ID/locator separation in NGN (Y.2015), ITU-T, January, 2009.
[9] AutoI http://ist-autoi.eu/autoi/index.php Yu-Hunag Chu, Min-Chi Tseng, Yao-Ting Chen, Yu-ChiehChou, Yan-Ren Chen, "A Novel Design for Future On-Demand Service and Security", IEEE ICCT 2