Realization of Open Cloud Computing Federation Based on Mobile Agent

Zehua Zhang and Xuejie Zhang
School of Information Science and Engineering, Yunnan University, Kunming, P.R. China
{zehuazh, xjzhang}@ynu.edu.cn

Abstract—**Although cloud computing is generally recognized as a technology which will has a significant impact on IT in the future. However, Cloud computing is still in its infancy, currently, there is not a standard available for it, portability and interoperability is also impossible between different Cloud Computing Service Providers, therefore, handicaps the widely deploy and quick development of cloud computing, there is still a long distance to the fine scenery which theoretically depicted by cloud computing. We analyze the problems in the current state of the art, put forward that Open Cloud Computing Federation is an inevitable approach for the widely use of cloud computing and to realize the greatest value of it. Accordingly, we proposal the MABOCCF (Mobile Agent Based Open Cloud Computing Federation) mechanism in this paper, it combines the advantages of Mobile Agent and cloud computing to provide a realization for the Open Cloud Computing Federation, MABOCCF can span over multiple heterogeneous Cloud Computing platforms and realizes portability and interoperability, it can be a beginning of open cloud computing federation and a future part of cloud computing. We also present in this paper the rationalities and the motivations for the combination of Mobile Agent and Cloud Computing, finally, a prototype is given with a performance analysis.**

Keywords-cloud computing; federation; mobile agent; portability; interoperablility

I. Introduction

With the quick development and extensively use of IT technology, cloud computing become a hot topic of industry and academe as an emerging new computing mechanism, with a great hope that it will provide computing as the 5th utility (after water, electricity, gas, and telephony) to meet the everyday needs of the general community [1], and it may lead to a third reform to IT, follows the PC and Internet.

Cloud computing represents a distributing computing mechanism that by the utilize of the high speed network, data processing is moved from private PC or servers to the remote computer clusters (big data centers owned by the cloud service providers), any user has a potential super computer at hand and can access the data and get the computing capability at any time, from anywhere, you only need to pay for the resources which you have used, don't care about who provide the resources and in what way. Cloud computing is considered has the advantages including infinite scalability, lower the TCO (Total Cost of Ownership), reduce the upfront investment and risk for the user, system automation and promote the realization of green IT.

However, cloud computing technology is still in its infancy [2], 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.

A. *Open Cloud Computing Federation and it's advantages*

The conception of Open Cloud Computing Federation is proposed by many researchers from industry and academe [3, 4]; 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 computing be widely 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 white paper from University of California, Berkeley [5], we analyze the problems 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 widely 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 be migrate to another CCSP, once your CCSP breakdown, 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.

978-1-4244-4738-1/09/$25.00 ©2009 IEEE

3) Service provider lock-in. It's impossible for you to move your data and application from one CCSP to another for some special considerations, because of the absence of portability; consequently, you are locked to a certain CCSP. An OCCF will democratize the whole cloud computing market where the small scale competitors can enter and thus promote innovation and vitality.

4) Unable to deployment service over multiple CCSPs. Currently, application can't scale over multiple CCSPs since there is no interoperability between CCSPs. OCCF can make it possible for an organization to integrate different CCSP's service to provide more valuable service for the end user.

5) High TCO (Total Cost of Ownership) for user. One of cloud computing's value is brought by the scale economy which 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.

B. Related work of OCCF and problems

Currently, the reality is that the API of different cloud computing platform is private, each CCSP has poured a large amount of funds on their own cloud computing facilities, it's impossible for them to abandon current platform and adopt a new cloud computing standard. As a result, despite IBM and Microsoft's call for a cloud computing standard, and The Open Cloud Manifesto (it believes that cloud computing must be open, define many principles to ensure that the cloud is open, but not a practical action plan) [6] supported by many IT giants, however there is still not a common cloud computing standard, cloud computing services of different CCSP can not migrate and interoperate between each other, OCCF currently remains in concept and models.

Reservoir is a European Union FP7 funded project aim to develop breakthrough system and service technologies that will serve as the infrastructure for Cloud Computing [7], B. Rochwerger et al. proposal the model and architecture of Reservoir in [3], they attempt to use grid interfaces and protocols to realize interoperability between the clouds or infrastructure providers, but their work is in the model stage, the author acknowledged that how to deal with the interoperability in grid is still a unsolved problem.

R. Ranjan et al. describe a Decentralized Overlay for Federation of Enterprise Clouds called Aneka-Federation, nevertheless, there is only one kind of enterprise cloud in the federation, portability and interoperability between different CCSPs isn't mentioned in the work.

OCCF lack a practical realization mechanism at present, since there isn't a cloud computing standard, portability and interoperability is impossible among CCSPs. We proposal the MABOCCF (Mobile Agent Based Open Cloud Computing Federation) mechanism in this paper, it can be a beginning to realize the OCCF and become a future part of cloud computing.

The rest of this paper is organized as follows: Section 2 describes the architecture of MABOCCF, the realization of this mechanism and the migration of the computing task. Section 3 analyses the performance of MABOCCF, describes the prototype for our experiment. Finally, a conclusion of the work is discussed in Section 4 along with the envisaged future work.

II. MABOCCF (Mobile Agent Based Open Cloud Computing Federation)

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

Mobile agent based cloud computing is feasible because most of mobile agent systems are based on or support Java, such as Aglets [11] and D' Agent [12], many cloud computing platforms currently provide virtue 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.

A. Architecture and components of MABOCCF

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) [13] and FIPA (Foundation for Intelligent Physical Agents) [14].

Because the realization of portability and interoperability, make it possible for the implemention 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 domains 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

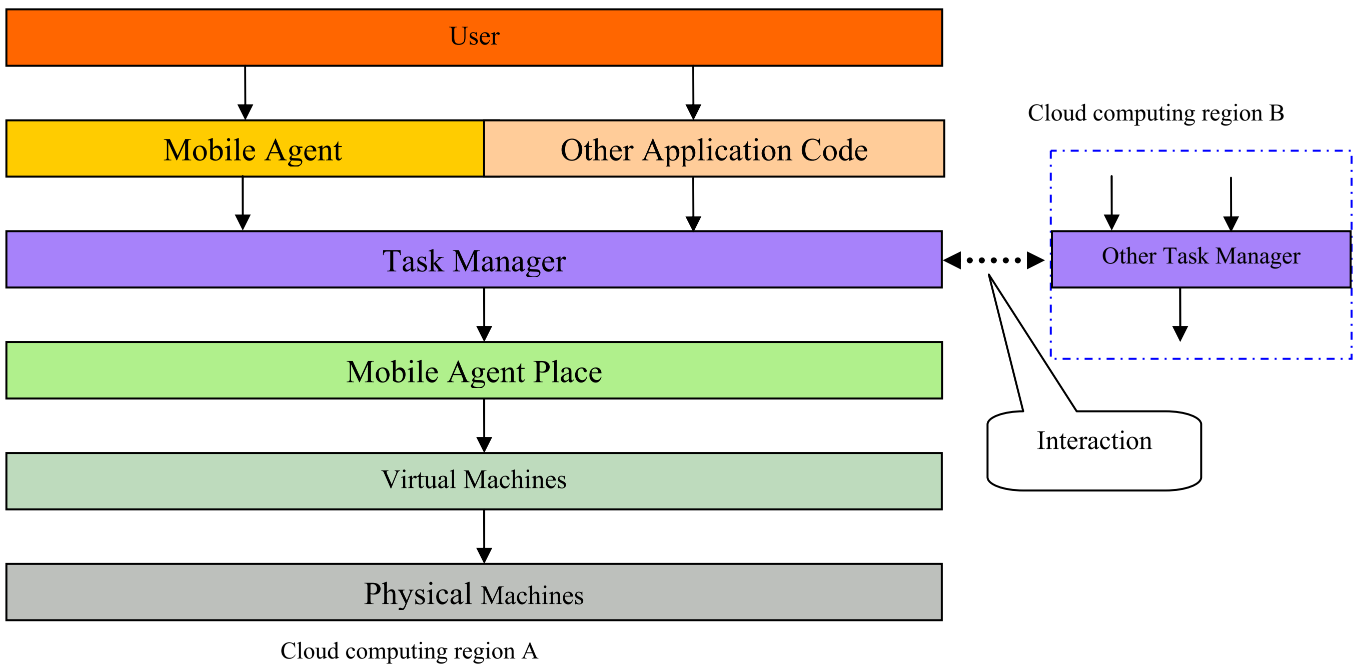


Figure 1. Architecture of Mobile Agent Based Open Cloud Computing Federation

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 also responsible for the backup and monitor 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 provides by the third party, the requirements of the resource (the demand for software, hardware and type of data resources) is wrote in the head of the mobile agent, the data structure of the mobile agent is shown as follows:

Type(is or not a Mobile Agent)	Requirements	Mobile Agent Code	Application Code or Tasks

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.

B. The MABOCCF mechanism

The mechanism of MABOCCF is shown in Fig.2, the user send a mobile agent to a TS, the TS read the head of the mobile agent to judge if it's a mobile agent or other kind of data packages, then the TS match the requirement 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 activate the mobile agent and execute the task including in the mobile agent, the assignment begin 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 other CCR 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 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 receive 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 deliver to the user by the TS.

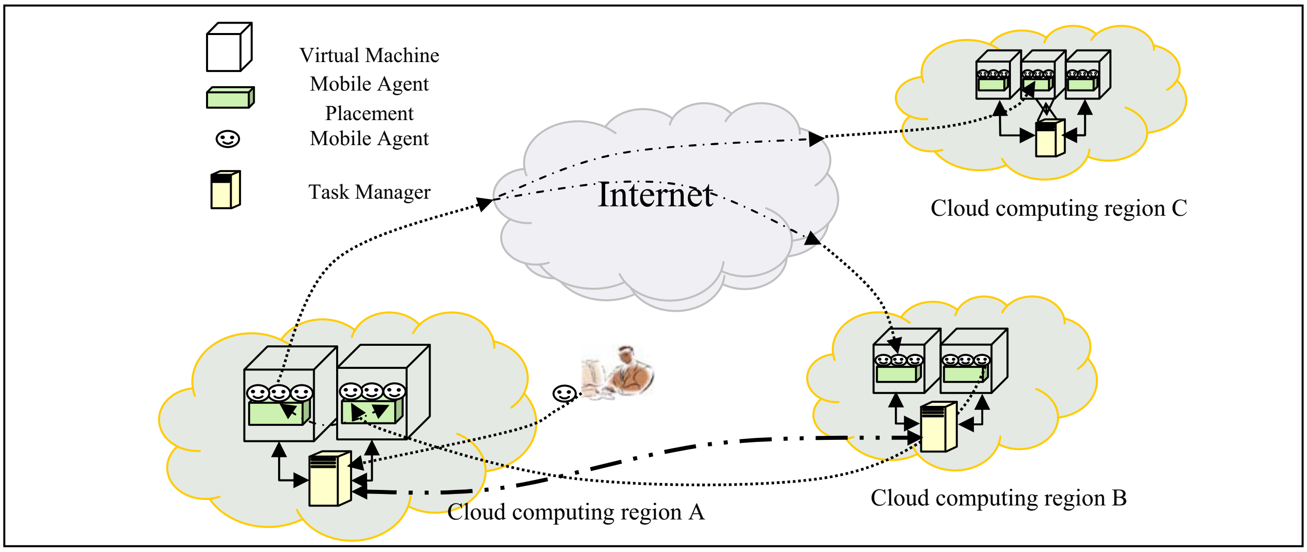


Figure 2. Mobile Agent Based Open Cloud Computing Federation mechanism

A more complex case is that the task is assigned to several mobile agents, and the mobile agents collaborate with 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.

III. The Performance Analysis of MABOCCF

MABOCCF is a realization mechanism of open cloud computing federation, thus possesses the advantages aforementioned in open cloud computing federation, moreover, the unique advantages brought by the combination of mobile agent and cloud computing are the really infinite scalability of computing resource, the ability to customize task migration by user or the third party, reducing the network load and good execution performance and more intelligent computing. We only compare MABOCCF with current cloud computing mechanisms that don't support portability between different CCSPs, since interoperability relate to particular applications.

We construct a prototype call MABOCCF1.0 for MABOCCF on single computer; it simulated the performance of 10 CCSPs (cloud computing service providers) for MABOCCF and NMBOCCF (ordinary computing mechanisms that don't support portability between different CCSPs), the time of experimentation is 3000 unit times (a unit time is a execute step defined by the program), suppose there are N_{it} tasks are submitted to the CCSP i at time t, and N_{it} is a random value between 0~100, all tasks arrive at time t have the same execute time ET_t, ET_t is a random value between 1~10, W_{it}=N_{it}*ET_t is the arrive workload of CCSP i at time t, and all CCSPs have the same Execute capability *EC*, *EC* is set to 275 workload/unit time, the tasks arrive at each CCSP at any time are the same for MBOCCF and NMBOCCF, and the task manage cost factor *TF* is set to 0.03.

The USM (Average user satisfaction) and *URM*(Average utilization ratio) for MABOCCF can be calculated by the following equations:

$$W_t=\begin{cases} 0 & t=0 \\ W_{i-1}+\sum_{i=1}^{10} N_{i1}\times ET_1 & t=1 \\ W_{it-1}+\sum_{i=1}^{10} N_{it}\times ET_t-10EC & t\in[2,3000] \end{cases} \quad (1)$$

Where W_t is the total workload of the CCF (cloud computing federation) at time *t*, and the user satisfaction of the CCF at time *t* can be expressed as:

$$USM_t=\begin{cases} 1 & W_t<=10EC \\ 10EC/W_t & W_t>10EC \end{cases} \quad (2)$$

So, the total average user satisfaction is:

$$USM=\frac{\sum_{t=1}^{3000} USM_t}{3000}\times TF \quad (3)$$

With (1), we can get the facility utilization ratio at time t as:

$$URM_t=\begin{cases} 1 & W_t>=10EC \\ W_t/10EC & W_t<10EC \end{cases} \quad (4)$$

With (4) we can obtain the total facility utilization ratio:

$$URM=\frac{\sum_{t=1}^{3000} URM_t}{3000}\times TF \quad (5)$$

The *USM* (Average user satisfaction) and *URM* (Average utilization ratio) for NMABOCCF can be calculated by the following equations:

$$W_{it}=\begin{cases}0 & t=0\\ W_{i0}+N_{i1}\times ET_1 & t=1\\ W_{it-1}+N_{it}\times ET_t-EC & t\in[2,3000]\end{cases} \qquad (6)$$

Where W_{it} is the workload of CCSP *i* at time *t*, so the user satisfaction of users on CCSP *i* at time *t* can be expressed as:

$$USNM_{it}=\begin{cases}1 & W_t<=EC\\ EC/W_{it} & W_t>EC\end{cases} \qquad (7)$$

So, the total user satisfaction is:

$$USNM= TP\times\frac{\sum_{t=1}^{3000}\sum_{i=1}^{10}(USRM_{it})/10}{3000} \qquad (8)$$

With (6) we can obtain the facility utilization ratio as:

$$URNM_{it}=\begin{cases}1 & W_{it}>=EC\\ W_{it}/EC & W_{it}<EC\end{cases} \qquad (9)$$

The total facility utilization ratio is:

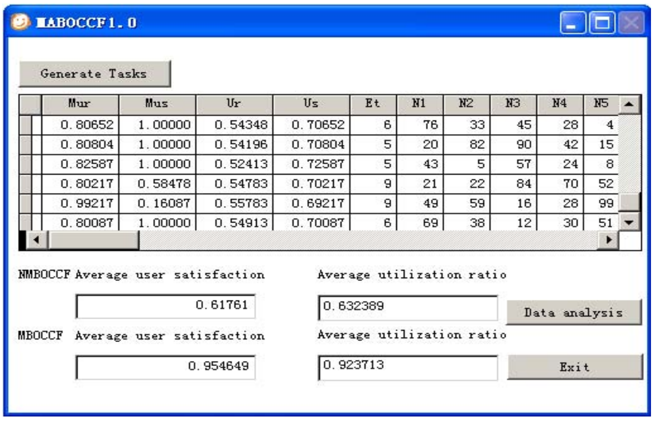
$$URNM=\frac{\sum_{t=1}^{3000}(\sum_{i=1}^{10}URNM_{it})/10}{3000} \qquad (10)$$

Experiment results with the aforementioned parameters are shown in Fig. 3.

The experiment results manifests that with the move of the computing tasks among the CCSP, the average user satisfaction and the average utilization ratio is considerably enhanced 54.6% and 46.1%, respectively.

IV. Conclusion and the Future Work

Currently, there is not a common standard for cloud computing, and OCCF is difficult to realize. We proposal in this paper a Mobile Agent Based Open Cloud Computing Federation mechanism which aims to realize portability and interoperability between different Cloud Computing platforms, and has a good performance in user satisfaction and the facility utilization ratio.



	Mur	Mus	Ur	Us	Et	N1	N2	N3	N4	N5
	0.80652	1.00000	0.54348	0.70652	6	76	33	45	28	4
	0.80804	1.00000	0.54196	0.70804	5	20	82	90	42	15
	0.82587	1.00000	0.52413	0.72587	5	43	5	57	24	8
	0.80217	0.58478	0.54783	0.70217	9	21	22	84	70	52
	0.99217	0.16087	0.55783	0.69217	9	49	59	16	28	99
	0.80087	1.00000	0.54913	0.70087	6	69	38	12	30	51

NMBOCCF Average user satisfaction: 0.61761
Average utilization ratio: 0.632389

MBOCCF Average user satisfaction: 0.954649
Average utilization ratio: 0.923713

Figure 3. Experiment results for MABOCCF and NMBOCCF

In the future cloud computing environment, there will be numerous of applications consist of millions of modules, these applications serve for large quantities of users, and the requirements of user is dynamic, in such situations, how to get a high user satisfaction and a better facility utilization ratio is a really a complex problem, the viewpoint of the Complex Systems Theory may give some clues.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (NSFC) (No.60573104) and the Natural Science Foundation of Education Department of Yunnan Province (No.09Y0050).

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, 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, June 2009, pp. 599-616.

[2] W. Kim, "Cloud Computing: Today and Tomorrow," Journal of Object Technology, vol. 8, no. 1, January-February 2009, pp. 65-72.

[3] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. Llorente , et al. "The Reservoir model and architecture for open federated cloud computing," IBM Journal of Research and Development, Volume 53, April 2009, in press

[4] R.Ranjan, R.Buyya, "Decentralized Overlay for Federation of Enterprise Clouds," http://arxiv.org/ftp/arxiv/papers/0811/0811.2563.pdf.

[5] "Above the Clouds: A Berkeley View of Cloud Computing," http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.

[6] "Open Cloud Manifesto," http://www.opencloudmanifesto.org/Open%20Cloud%20Manifesto. pdf.

[7] http://www.reservoir-fp7.eu/.

[8] D. B. Lange, M. Oshima. Seven good reasons for mobile agents. Communications of the ACM, 1999, 42(3), pp. 88-89.

[9] G. Cabri, L. Leonardi, F. Zambonelli. Mobile Agent Technology: Current Trends And Perspectives. AICA' 98, 1998, The paper is only available online at http://www.agentgroup.unimo.it/MOON/papers/toBeDeleted/aica98.pdf.

[10] C.Spyrou1, G.Samaras, E.Pitoura and P.Evripidou, "Mobile agents for wireless computing: the convergence of wireless computational models with mobile-agent technologies Mobile Networks and Applications, Volume 9, October 2004, pp. 517-528.

[11] http://aglets.sourceforge.net/.

[12] http://agent.cs.dartmouth.edu/.

[13] D. Milojicic, M. Breugst, et al. MASIF: The OMG mobile agent system interoperability facility. Personal and Ubiquitous Computing, Volume 2, June 1998, pp.117-129.

[14] http://www.fipa.org/