IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
ISSN: 2278-2834, ISBN: 2278-8735. Volume 4, Issue 3 (Nov-Dec. 2012), PP 25-31
www.iosrjournals.org

Cloud deployment model and cost analysis in Multicloud

Subash Babu Pandiri[1,] Sangeeta Kulkarni[2]
[1, 2](*Electronics and Telecommunication, K.J.Somaiya college of engineering/ Mumbai University, India)*

Abstract : *Cloud computing is a revolution in the way it offers computational capability. The information technology organizations do not need to oversize the infrastructure anymore, potentially reducing the cost of deploying a service. The two main objective of this paper is to study a cloud deployment model which involves user, broker, federated catalog system and to decrease the cost of deploying a service using a scheduling model. Many research works are going on to obtain the best out of the cloud. In this work, we propose a system which analyzes the cost by a scheduling model for optimizing virtual cluster placements across available cloud offers.*
Keywords: *cloud computing, federated cloud computing, intercloud, SLA, resource allocation.*

I. Introduction

Cloud computing is a novel paradigm for the provision of computing infrastructure, which aims to shift the location of the computing infrastructure to the network in order to reduce the costs of management and maintenance of hardware and software resources. Cloud computing has a service-oriented architecture in which services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), which includes equipment such as hardware, storage, servers, and networking components are made accessible over the Internet; Platform-as-a-Service (PaaS), which includes hardware and software computing plat-forms such as virtualized servers, operating systems, and the like; and Software-as-a-Service (SaaS), which includes software applications and other hosted services [1]. A cloud service differs from traditional hosting in three principal aspects. First, it is provided on demand; second, it is elastic since users can use the service have as much or as little as they want at any given time (typically by the minute or the hour); and third, the service is fully managed by the provider [2].There needs an architecture which can describe the deployment of the cloud clearly.

A facility node contains different computational resources which can compute any task sent to the cloud center. A service level agreement, SLA defines all the aspects like cloud service usage, obligations and quality of service (QOS). QOS includes, availability, through put, security, response time, blocking probability and immediate service in the system. In this paper, we model a queuing system which can evaluate the performance of the cloud centers related to the response time, blocking probability and immediate service in the system.

As the cloud computing market grows and the number of IaaS provider’s increases, the market complexity is also increasing as users have to deal with many different Virtual Machine (VM) types, pricing schemes and cloud interfaces. This trend has brought about the use of federated clouds and multi-cloud deployments. The aim of federated clouds is to integrate resources from different providers in a way that access to the resources is transparent to users while scalability and reliability are increased, and cost can be reduced [3].

So in this context, brokering mechanisms and scheduling schemes are required to collect the price data automatically. From an economic point of view, these cloud providers adopt different economic models to rent their resources. Types of economical models are static prices, which are rarely changed. Dynamic prices, which are frequently changed according to the duration used and spot prices, which are dependent on user bids. In the first stages of the cloud market only static prices were offered by providers. However, nowadays dynamic pricing schemes are being widely adopted.

In a dynamic pricing we need a cloud scheduler which is a part of cloud broker to collect updated prices periodically and apply different algorithms for scheduling each deployment over each cloud center. It also permits to optimize dynamically the total infrastructure cost by changing some VMs placement to the cheapest cloud.

II. Proposed Models

Cloud deployment model is to understand the architecture involved in deploying a cloud center. The deployment model is divided into four layers. These layers are consumer, broker, federated catalog system (internal intercloud), federated cloud system (external intercloud).Whereas to implement cost analysis, we modeled a dynamic scheduler for multicloud brokering environments.

1. Cloud deployment model

In this architecture consumer take service either from private cloud provider or public cloud provider .This model involves user, broker, federated catalog system (internal intercloud), federated cloud system (external intercloud).

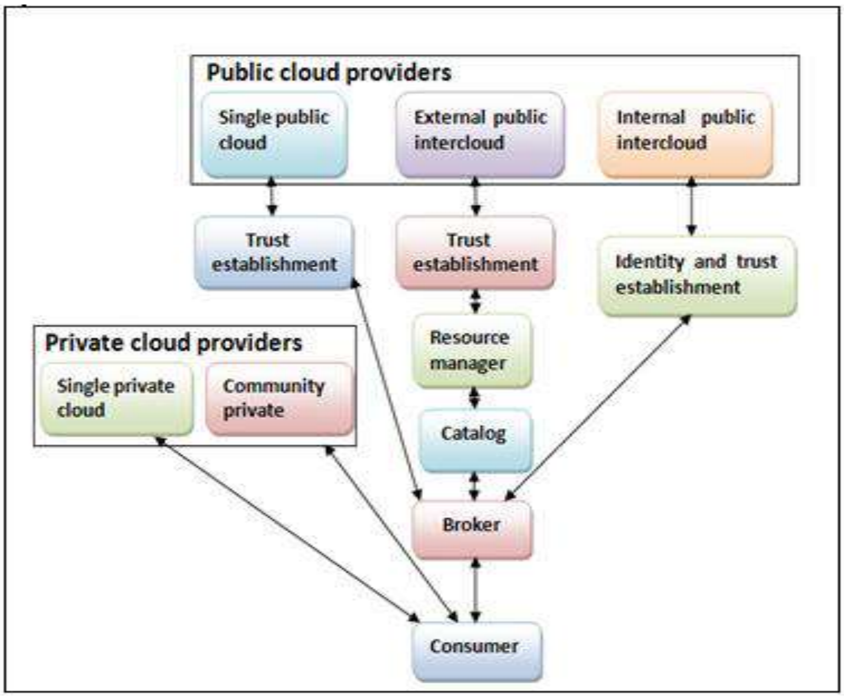

Figure 1: Cloud service architecture

1.1 User

If a user will use private cloud resources then he should use only that cloud provider services but if that user use public cloud resources then that user have to pay money to use public cloud resources. In a public cloud, there is no need of brokering as the user can have an agreement for the quality of service (QoS).

1.2 broker

Cloud computing is based on pay as use model. Cloud computing is an internet based application where resources are in distributed manner. So the consumer needs dynamic update of the resources available and also make negotiation for those resources dynamically on the basis of demand and payment.

So in this dynamic environment service level agreement (SLA) is finding a good approach for this negotiation and agreement between consumer and service provider.

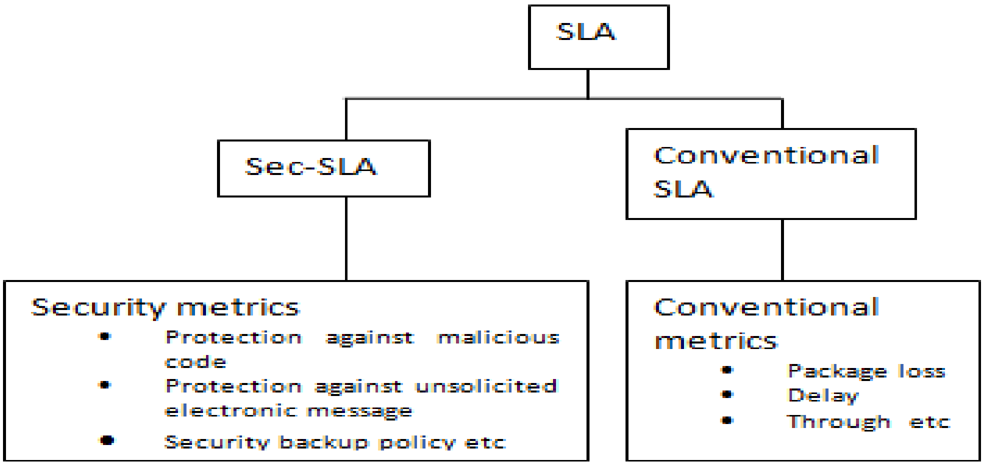

Figure 2: SLA classification SLA has three phases: Creation –operation- removal.

Cloud computing is internet based, dynamically scalable and virtualized. So because of all these features and to manage the cloud resources dynamically and on demand basis the SLA need some set of rules.

These rules will not bind to consumer to take his decision on the basis of these rules but these rules will only suggest to the customer which service will be better for him.

1.3 federated cloud system (internal intercloud):

This approach gives the most efficient, scalable and optimal provisioning of resources. To make this federated catalog we need a systematic process. This process is divided in four phases:

a) First phase: In this phase federated system need to verify the identity of cloud providers for authentication.
b) Second phase**:** Discovering different services provide by public cloud provider and updating catalog time dynamically.
c) Third phase**:** Maintain different public cloud resources information in catalog and schedule them on the basis of different parameter. These parameters called templates for different-different consumers and these templates are highly confidential and secure.
d) Fourth phase**:** It is broker which work on the basis of SLA.

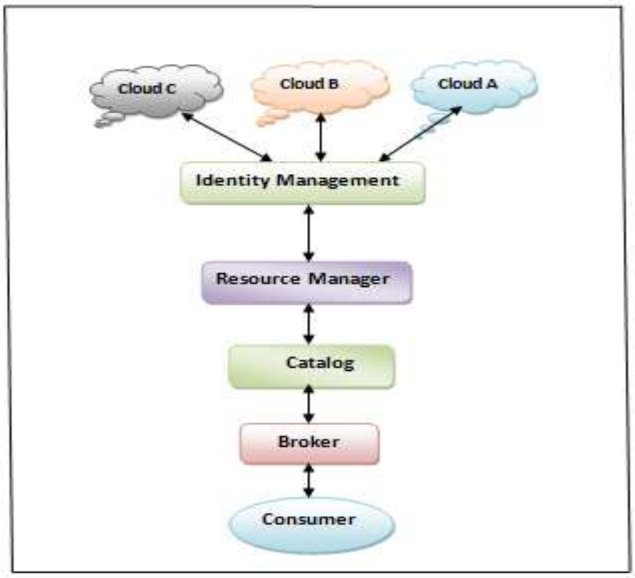

Figure 3: Federated catalog system architecture

1.4 federated cloud system (internal intercloud):

As in external intercloud we need to maintain an external catalog system. In internal intercloud we need not maintain any resource catalog system. In internal intercloud if a public cloud resource provider (here we called this cloud as local cloud) unable to fulfill the requirement of its customer then that public cloud provider ask for needed resources from another public cloud provider (here we called this type cloud as global cloud) and provide these resources to its customer continuously without any interruption and without knowing by the customer about actual service provider or global cloud. In internal intercloud the local cloud make the virtual connection with global cloud and in this virtual connection the local cloud seems the global cloud resources as its own resources. In this internal federated cloud system local cloud provider will automatically ask for services from global cloud provider where customer will get uninterrupted service without knowing background process. it work on Single Sign On(SSO) approach. In this approach the cloud which is directly attached by customer is called local cloud and other cloud which gives services to Local Cloud (LC) is called Global Cloud (GC).Internal intercloud has 4 phases

a) First phase: Establishing trust between different cloud.
b) Second phase: Discovery of desired resources.
c) Third phase: Migration of resources between clouds
d) Fourth phase: Security of transmission data

III. System Architecture

In this work a dynamic scheduler for multi-cloud brokering environments is implemented, and focused on the architecture shown in Figure 5.

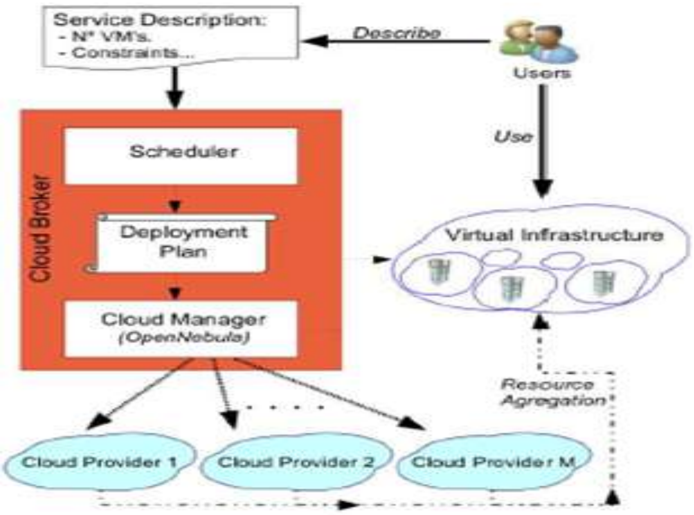

Figure 4: cloud brokering architecture overview

In this architecture, users can request a given number of virtual machines to deploy among available clouds and specify some restrictions. At the same time, cloud providers can offer different kind of resources associated to particular pricing schemes. Finally, the cloud broker is an intermediary between users and providers. Cloud broker is used to virtual resources of a virtual infrastructure in several cloud providers and to manage and monitor the system according to user criteria. To obtain this, we use two different components.

- Scheduler: This component is responsible for taking the placement decision, which is addressed using a dynamic approach. The dynamic approach is suitable for variable conditions. We have implemented this cloud broker's scheduling module for managing dynamic pricing situations. In order to achieve an optimal deployment of virtual resources according to user criteria, the scheduler can use several algorithms which are described in the next section. These algorithms should run periodically to adapt the resource deployment to the variable cloud pricing conditions. The scheduler output is a deployment plan, which is composed by a list of VM templates. Each template includes the target cloud provider to deploy the VM in, and some attributes for the selected provider. Also, this deployment plan is used as an input for the second component.
- Cloud manager: This component addresses the management and monitoring actions using the *Open Nebula* virtual infrastructure manager [5]. Open Nebula provides an uniform and generic user interface to deploy, monitor and control VMs in a pool of distributed physical resources. The use of specialized adapters is a well-known technique to achieve a multi-provider interoperation. In general, Open Nebula is able to interoperate with some cloud providers, like Amazon Elastic Computing Cloud (EC2) [5], Elastic Hosts (EH) or other clouds compatible with Delta cloud-based API. These adapters convert the general requests made by the virtualization components of Open Nebula to manage VMs through the respective APIs.

2.1 Problem formulation

In this section we propose some scheduling algorithms for optimal deployments in a dynamic cloud pricing scenario. In this scenario, the goal is to deploy a clustered service composed of a set of components which are executed as virtual machines in different cloud providers.

In this work, we deploy a fixed number (n) of VMs, $v1 \ldots vn$ across m available clouds $c1 \ldots cm$ to optimize user criteria such as cost or load balance.

The scheduling period is treated as next one hour period to schedule as we worked in periods of one hour. Hence, the scheduler will be executed before the beginning of each scheduling period. This action can prompt a total or partial reconfiguration of the virtual infrastructure. Each cloud provider can offer a given number of hardware configurations, called instance types.

In this approach we have some considerations.

- We define *t* as any one-hour period.
- We consider a 0-1 integer programming formulation where $X_{i,k}(t) = 1$ if virtual machine vi is placed at cloud ck during period *t*, and 0 otherwise.
- Using *t*, we define $P_k(t)$, $1 \leq k \leq m$ as the real price paid for a virtual machine deployed in cloud *k* during period *t*.

We want to minimize is the Total Infrastructure Cost (TIC(t)) which is defined as the sum of the cost of each virtual machine for a given time period. But user does not know prices for next hour because dynamic demand makes prices change. In fact, prices of cloud resources in period *t* are unknown until period *t* finishes.

In this work we call *oracle* prices to the best possible prices in each scheduled period. If we could know these prices, we would act as an oracle for taking decisions and we would get the optimal ones. Oracle prices are unknown until the scheduling period ends, and we use them to check the quality of the proposed model by comparing our results with these prices. In order to achieve these prices using the scheduler, we define an *Oracle* function shown in (1).

$$TICOracle(t) = \sum_{i}^{n} \sum_{k}^{m} Xi,k(t) * Pk(t) \quad (1)$$

However, as the goal is to find an optimal resource deployment for next period *t* before knowing the prices of each cloud provider, one valid solution would be trying to predict next hour prices using estimations. For that purpose, we define: Ek(t), $1 \le k \le m$ as the estimated price of a virtual machine deployed in cloud *k* during period *t*. We need some parameters to base our decisions for predicting these Ek(t) prices. Thus, we have defined two additional parameters:
Average price of a cloud provider: Avk.

$$Avk(t) = [\sum Pk(x)] / [(t-1) - t0], \ 1 \le k \le m \quad (2)$$

In (2) we calculate the sum of last observed prices of a virtual machine in a particular cloud provider. And then, this sum is divided by the period of time used for this calculation, showing an average value.
If we use Avk parameter over a time interval to adjust our function to minimize, we would obtain a logical prediction about where to deploy our virtual resources. However, the scheduler would not know in which part of the time interval did the cheapest prices appear, if near or far from the hour to Predict. So, recent price trend become interesting for making predictions. For that purpose, we have defined the trend parameter as follow:

- Trend of a cloud provider: Tnm.

$$Tnm(t) = 1,05 \text{ if } Pm(t-1) > Pm(t-2) \ge Pm(t-3)$$
$$Tnm(t) = 0,95 \text{ if } Pm(t-1) < Pm(t-2) \le Pm(t-3)$$
$$Tnm(t) = 1 \text{ otherwise} \quad (3)$$

Trend is defined as the relationship between three last observed prices. The last known price can be higher, lower or the same as the previous one. We define three types of trends: increasing, when next price is supposed to get higher, decreasing, when next price is supposed to get lower, and constant trend, when next price is supposed to maintain the previous prices trend. An increasing trend in a cloud provider means that this provider is holding a high load level in this interval. Similarly, a decreasing trend in a cloud provider means that it is not overloaded and its physical resources are up to receive more clients.
The possible values for Tnm(t) are the penalty that Ek(t) will suffer depending on the trend. With these two new parameters, Avm(t) and Tnm(t), we can define the estimated cost variable
as:

$$Ek(t) = Avk(t) * Tnk(t), \ 1 \le k \le m \quad (4)$$

Looking back at the definition of TIC, now the estimated total infrastructure cost is defined as the sum of virtual machines in each cloud multiplied by the estimated cost per hour. The equation to minimize is TIC (estimated):

$$TICest(t) = \sum_{i}^{n} \sum_{k}^{m} Xi,k(t) * Ek(t) \quad (5)$$

From (4) and (5) we obtain the following expression:

$$TICest(t) = \sum_{K}^{m} Xi,k(t) * Avk(t) * Tnk(t) \quad (6)$$

By minimizing equation (6) we address the challenge of deploying virtual resources in the cheapest clouds.

IV. Experimental Results

In order to evaluate the scheduling model, we followed these steps: first we collected historical data from a real cloud provider; then we observed them and made some experiments to extract an optimal deployment for each hour of the observed period. After these steps, we compared obtained results with the cost of maintaining the whole cluster in a fixed placement and also with oracle prices. For these experiments we have simulated a virtual infrastructure composed by 10 VMs during a 24 hours period. These VMs use an Amazon small instance type and they are initially placed at EU region because of its cheapest price.

1. Data collection

In this work, we obtained the experimental data by consulting Amazon EC2 historical spot prices. We use spot prices variations to simulate a realistic dynamic pricing cloud environment. We observed Amazon EC2 historical spot prices of a small instance in three different regions: United States West (USW), Asia (AS) and Europe (EU). Generally, each resource type in any placement has a different start price and then it fluctuates based on its particular demand. So, we extract a hourly list of prices of each region until one day (24 hours) was completed. The trend of these prices can be observed in Fig.6.

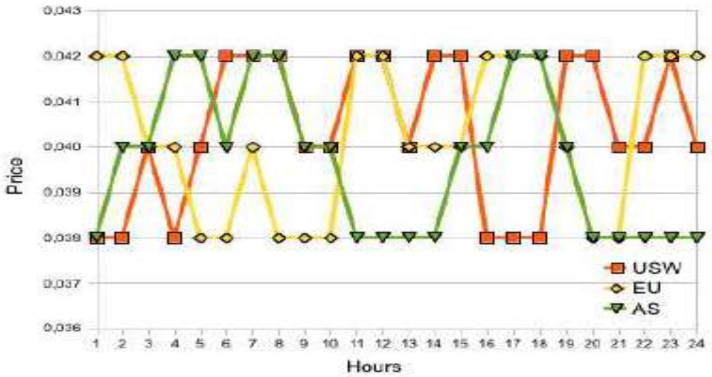

Figure 5: hourly prices

Then we have compared the static and oracle deployments. The static deployment consists on maintaining the whole set of virtual machines in the same placement during the 24 hour period. Hence, we consider two different cases:

- Single cloud deployment: the entire cluster is deployed in one cloud region. As we work with three regions, we measured this cost in each one.
- Balanced deployment: VMs are equitably distributed among different cloud providers.

The oracle deployment consist of moving the whole cluster to the cheapest cloud region in each period *t* assuming that we know a priori the real prices of each time period. Fig.3 shows the difference between both types of deployment in a 24 hours experiment. Comparing the static deployments, the balanced one is cheaper than two of the single ones, so if we could not make any placement change and we don't know future prices, this deployment ensure a good choice. Moreover, we were interested in knowing the improvement potential of the dynamic scheduling by comparing the oracle deployment with the static ones.

2. Price estimation

The following step to perform once we observed the potential of dynamic scheduling was to check the quality of the price estimation model used by the scheduler module. For making good-enough predictions, estimated prices are expected to approximate to the real ones.

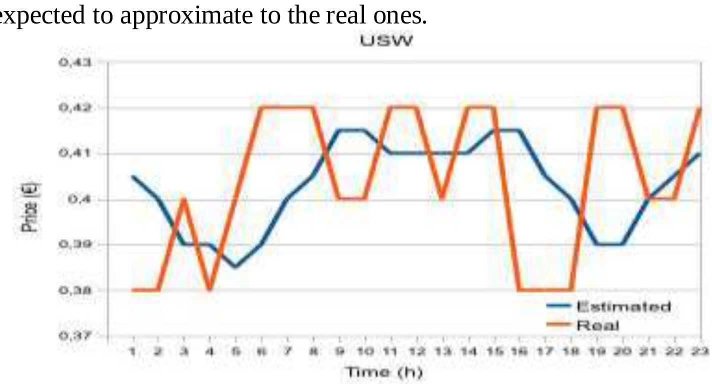

Figure 6: Price estimation applied to USW data

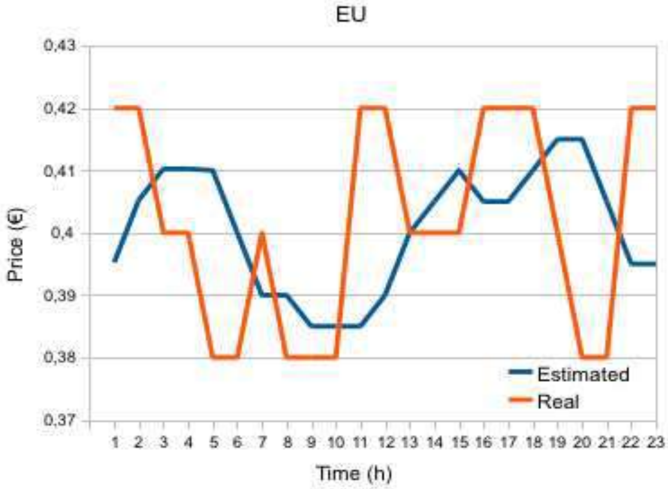

Figure 7: Price estimation applied to EU data

Estimated prices are calculated with equation (4). In Fig.6 and Fig.7 are depicted the estimated prices and trends for USW and EU regions respectively, compared to the real ones. As can be observed, the scheduler predicts prices and trends in a satisfactory way.

V. Conclusion

This paper tells about different ways of providing resources to consumer from cloud. Firstly paper distinguishes between private cloud and public cloud. The cloud deployment model was explained where a brokering is required only in private cloud.SLA is useful for a user. And also a novel cloud broker approach for virtual infrastructures deployments in dynamic pricing multicloud environments is introduced. This broker is composed by a cloud scheduler who addresses the dynamic deployment challenge. The scheduler is based on a prediction model which takes into account the historical prices of available cloud providers. Using these prices, it calculates their average and trend and predicts the best next hour deployment. The aim of this scheduler is to decrease user's investment while maintaining a particular user-managed virtual cluster performance. To achieve these goal variable prices offered by cloud providers by moving part of the cluster to the cheapest placement is used.

References

[1] B. Furht, Cloud Computing Fundamentals, Handbook of Cloud Computing, pp. 3-19, Springer, 2010.

[2] H. Khazaei, J. Mi~si~c, and V.B. Mi~si~c, Performance Analysis of Cloud Centers, Proc. Seventh Int'l ICST Conf. Nov. 2010.

[3] M. Mihailescu and Y. M. Teo, "Dynamic resource pricing on federated clouds" 10th IEEE/ACM international conference on cluster ,cloud and grid computing.

[4] Grimmett and D. Stirzaker, Probability and Random Processes, third ed. Oxford Univ. Press, July 2010.

[5] *Sotomayor.R.monotero,I.LIorento.and I.Foster* "Virtual infrastructure management in private and hybrid clouds," *Internet Computing IEEE*, vol. 13, no. 5, pp. 14 –22, sept.-oct. 2009.

[6] Bernstein, D.; Vij, D.; "Intercloud Security Considerations," Cloud Computing Technology and science , IEEE Second International Conference on , vol., no., pp.537-544, Nov. 30 2010-Dec. 3 2010.

[7] Celesti, A.; Tusa, F.; Villari, M.; Puliafito, A.; , "Three-Phase Cross-Cloud Federation Model: The Cloud SSO Authentication," Advances in Future Internet (AFIN), 2010 Second International Conference on , vol., no., pp.94-101, 18-25 July 2010.

[8] Celesti, A.; Tusa, F.; Villari, M.; Puliafito, A.; , "Security and Cloud Computing: InterCloud Identity Management Infrastructure," Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on , vol., no., pp.263-265, 28-30 June 2010.

[9] Ashley Chonka,YangXiang n, WanleiZhou, AlessioBonti(2011), "Cloud security defense to protect cloud computing against HTTP-DoS and XML-DoS attacks" Network and Computer Applications 34 (2011) 1097–1107.