ISSN 22773061

Trusted Cloud Platform for Cloud Infrastructure

Punit Gupta[1], Deepika Agrawal [2]
[1]Department of Computer Science Engineering, JUIT
Himachal Pradesh, India
punitg07@gmail.com
[2]Department of Computer Science Engineering,
Government Engineering College
Sejbahar, Raipur, India
deepika721@gmail.com

ABSTRACT

Reliability and trust Models are used to enhance secure , reliable scheduling , load balancing and QoS in cloud and Distributed environment. Trust models that are being used in Distributed and Grid environment, does not qualify cloud computing environment requirements. Since the parameters that have being taken into consideration in these trust models, does not fit in the cloud Infrastructure As A Service, a suitable trust model is proposed based on the existing model that is suitable for trust value management for the cloud IaaS parameters. Based on the above achieved trust values, trust based scheduling and load balancing is done for better allocation of resources and enhancing the QOS of services been provided to the users. In this paper, an trust based cloud computing framework is proposed using trust model ,trust based scheduling and load balancing algorithms. Here we describe the design and development of trusted Cloud service model for cloud Infrastructure as a service (IaaS) known as VimCloud .

VimCloud an open source cloud computing framework that implements the tusted Cloud Service Model and trust based scheduling and load balancing algorithm . However one of the major issues in cloud IaaS is to ensure reliability and security or used data and computation. Trusted cloud service model ensures that user virual machine executes only on trusted cloud node, whose integrity and reliability is known in term of trust value . VimCloud shown practical in term of performace which is better then existing models.

Keywords

Cloud, QoS, Trust Management, Cloud IaaS, Scheduling, VMM, Distributed applications, Distributed programming.

ISSN 22773061

I. INTRODUCTION

Distributed and parallel computing has being in use since long time, but with cloud computing their benefits came into focus. Cloud computing allows user storage and computing on demand, moreover with much more reliability and computation power. With distributed environment, there comes the problem of security and reliability i.e. how to decide which datacenter is trust worthy and reliable as per the requirement and application. For example, in grid computing trust can be decided on the basis of it capability to compute multiple tasks in parallel or its failure rate.

There are many cloud IaaS frameworks that provide cloud computing services and virtualization services to the user like OpenNode [26], CloudStack [27], Eucaliptus [10], CloudSigma [28], EMOTIVE(Elastic Management of Tasks in Virtualized Environments) and Archipel. The problem with these frameworks is they do not take into consideration the properties of datacenters like fault rate and processing speed.

Different trust management models are being proposed to solve these problems, to categorize the datacenters on the basis of their trust value, being calculated on the basis of the few parameters taken into account. The task of these models is to categorize the datacenters, not only on the basis of the one parameter, but on the basis of multiple parameters collaboratively.

In general, trust management is been considered as a security issue in distributed environment which is based on key exchange. This is being considered in the fields like MANETS, VANETS and in general distributed system applications. In grid computing, where there are many datacenters and large task are to be computed at the processing level, trust has different meaning. Here it refers to the system that is most reliable based on the request, the task which is to be computed and the properties of datacenters or grids, in which the task is to be computed. The trust management models are implemented in grid computing that maps the requirement of cloud IaaS, which is explained later in this paper.

There are many models being proposed for cloud computing, especially in SaaS (software as a service) which are referred in this paper to identify the difference between the trust management models. The main difference between the cloud SaaS and IaaS is that in SaaS there are many domains and data types which has different properties. But on the other hand, in IaaS there is only one data type that is the VM (virtual machine) image. Hence we consider the processing capabilities of different VMMs (virtual machine monitors) to calculate the trust value of the datacenter or the node.

The trust value that is calculated by the trust model on the basis of different parameters, are used to prioritize the datacenters and scheduling algorithm to make use of these trust values, i.e., to schedule the request based on the trust values. Based on the trust values load balancing is done for better selection of node for better balancing of load.

In current scenario of cloud IaaS, we use load balancing and scheduling algorithms for better resource utilization and better QOS to the user. But there are different types of user who have fewer resources and has paid less, user with large resources and has paid more, and a free user or a public user. So taking into consideration basic load balancing algorithm where a user VM (Virtual Machine) is re-allocated based on the load on a datacenter and the cost of datacenter. But not taking into consideration the properties of the datacenter. Due to this a datacenter with High QoS (Quality of service) is been allocated to a public user and the request from the other user who has paid more will be allocated a datacenter with low QoS.

The problem of load balancing is been tried to solved using Honeybee [1], Ant Colony [19] but it has not taken into consideration the properties of a VMM in a datacenter. Scheduling problem is been tried using cost based scheduling algorithms [14] but both has not taken into consideration the properties of a VMM in a datacenter.

So we propose a trust management model to overcome this problem, by taking into consideration VMM characteristics which vary from datacenter to datacenter. Then these trust value are been used by load balancing and Scheduling algorithms proposed to improve the QOS provided to the user and better utilization of resources. .

II. RELATED WORK

A. Trust Model

Trust can be defined as an entity based on reliability and firm belief based on attribute of the entity. Trust is the firm belief in the competence of an entity to act as expected such that this firm belief is not a fixed value associated with the entity but rather it is subjected to the entity's behavior and applies only within a specific context at given time[2]. That means firm belief is dynamic value which varies timely. According to Gradison trust is the activity of collecting, codifying, analysis and evaluating evidence relating to competence, honesty, security and dependability with the purpose of making assessment and decision regarding trust relationship [3].

Trust can categorize into many type such as [3]:

a) Blind trust: This is default trust before any event on the system, and which would include agent to initiate relationship with unknown entities.

b) Conditional trust: This is a classical state of trust during the life of the agent .This condition trust is likely to evolved ,and can be subject to some sets of constraints or condition .

c) Unconditional trust: Such a trust is the probability be configured directly by an administrator, and would not be sensitive to successful/unsuccessful interaction and external recommendation of any other sources of evolution of the conditional trust.

Trust management models can be defined as the models to manage and manipulate trust value. These models are basically of two types:

a) Relative trust where trust value is calculated based on the relative reliability and dependability between two entities.

b) Direct trust that takes into consideration the reliability of each entity independently.

Many classification of trust management models are being proposed, that are behavior based, domain based, network topology based, agent based and so on [4][5][25]. In these different domains are been considered which can provide different cloud services with their specific trust value. The resources provided by same domain retain the same trust value. So the resources are being allocated to the user based on the trust value of the domain. The trust value degrades or increases depending on the activities over a file, time taken to complete a transaction, and transfer a file [6]. Table 1 shows how the trust value is stored corresponding to a domain and services.

Table 1

Domain name	Service type	Trust value/trust degree	Generation time
……	…….	…….	……

Based on reputation based trust, a model is being proposed by Faraz Azzedin for grid computing [7]. In this they have taken into consideration the past experience for calculating trust value. Past experience involve the calculation based on the last transaction and last trust value. The trust value is relatively calculated based on the reputation vector of two entities. In this two values are stored trust value and reputation. Let there be two domains D1 and D2, D1 stores the trust value about D2 based on the D1 direct relation with D2 as well as the reputation of D2. Domain experience is given by Γ(D1;D2; t), t) direct relation is given by Θ(D1;D2; t) and reputation of D1 at time t is given by Ω(D1; t). To manipulate the trust and reputation, they defined a decay function γ(t-tij) which is based on time, the last experience and reputation between two domain is stored as R(D1,D2). Trust is denoted as DTT [7]. Figure 1 explains the related calculation.

$$\Gamma(D_i,D_j,t)= \alpha \times \Theta(D_i,D_j,t)+ß\times \Omega(D_j,t)$$

$$\Theta (D_i, D_j, t) = \alpha \times DTT (D_i, D_j) +ß\times \gamma (t-t_{ij})$$

$$\Omega(Dj,t)=(\sum^{n}_{k=1}RTT(D_k,D_j)\times R(D_k,D_j)\times \gamma(t-t_{kj}))/\sum^{n}_{k=1} (D_k)$$

Another proposal given by R.Kumar [8] based on direct trust resource scheduling in grid computing. They have taken into consideration few factors such as affordability, success rate and bandwidth. On the basis of these parameters trust matrix is being created which is further used to calculate the direct trust value of the resource and schedule the resources depending on the trust value of the resource and the type of job to be executed [8]. Behavior based trust model is also one of the important model which is been given by Elvis Papalilio [9] for grid. It takes into consideration the behavior of the participant, on the basis of expected response from another participant in an interaction or transaction.Other features such as the availability, accessibility, accuracy, response time, and latency the all are taken into consideration to decide the behavior of the participant during interaction.

B. Load Balancing Algorithm

Load Balancing is a method to distribute workload across one or more servers, network interfaces, hard drives, or other computing resources [19]. Generally datacenters have high computing hardware with higher storage capacity and high network speed but they rely on Scheduling and mainly on Load Balancing algorithm for best utilization of their resources even when there are limited datacenter resources and network resources at time of high demand.

In cloud Load balancing algorithm is used to provide the full fill the request economically by providing cheaper and reliable resource, other aim or this algorithm is to provide the feature of scalability in cloud computing i.e. adding and removing resources at any interval of time.

Goal of Load Balancing [23]

1) To improve the performance substantially.
2) To improve system stability.
3) To have scalability in system.
4) To improve the system condition under high load or request rate.

Depending on the current state of the system, load balancing algorithms can be divided into 2 categories as given in [20]:

Static: It does not depend on the current state of the system. Prior knowledge of the system is needed.

Dynamic: Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach.

Many load balancing algorithm for cloud IaaS are being proposed that are network based, request based and distributed algorithms. Some of them are discoursed here Honeybee based load balancing [1] is proposed by M.Randles to improve the load balancing by selecting the nearest node to the used which has less load on it and fulfills the request.Ant colony based algorithm proposed by Kun Li [22] and a modified ant colony based algorithm is proposed by Ratan Mishra [19]. Both the above given Ant colony based algorithm differ that the algorithm proposed by kun Li is basic ant colony algorithm but modified ant colony take into consideration some of the features of cloud IaaS and dynamic load balancing with basic ant colony algorithm to get optimized results.

C. Scheduling Algorithm

Scheduling algorithms being proposed are based on the algorithm used in operating system like round-robin, first come first serve (FCFS) and so on. These algorithms are being implemented in Eucalyptus and OpenNebula. In Eucalyptus, Greedy and Round-robin algorithms are implemented and also scheduling algorithm on power saving for datacenters [10][11]. In OpenNebula, ranking scheduling policy is being implemented and rank is based on the free CPU's [12] [13].But both the scheduling algorithms does not meet the practical scenario i.e., they do not take into consideration the cost of the datacenter, properties of the datacenters, their fault rate and VMM characteristics. As proposed by Zhiyang [14], a cost based scheduling algorithm for cloud IaaS, which takes into consideration the cost of resources of the datacenter which are cost of the CPU, memory and no. of cores. Based on these, they calculated the cost according to the user request. Cost is been calculated at different datacenters and the datacenter with optimal cost is allocated to user. There are some other algorithm which are based on history, genetic algorithm [15] [16] [17][18].

III.PROPOSED WORK

A. Trust Model

From above trust modes they only take into consideration either the request type or the request that are completed in past, to overcome that and map them to real paradigm the behavior based algorithm was proposed and rank based models were proposed. But the problem with these algorithms is that they do not take into consideration the QoS of datacenter.

The model proposed here is to increase the QoS being provided to the user. Proposed trust model take into consideration parameters of datacenter and QoS provided by datacenters which are as follows:

a) Initiation Time: How long it takes to deploy a VM.
b) Price: How much a SaaS provider has to pay per hour for using a VM from a resource provider?
c) Processing Speed: How fast the VM can process. We use Machine Instruction per Second (MIPS) of a VM as processing speed.
d) Fault Rate: It is defined as the number of faults over a period of time.
e) Bandwidth: Speed at which data transfers in and out of the VMM/datacenter.

The model proposed here is to increase the QoS being provided to the use. Trust model are used to calculate trust values for the datacenters based on the parameters as explained above, based on these parameters trust value is calculated. Then these trust values are being updated after a fixed period of time which can vary because if trust is call updated frequently then there will be no considerable changes in the trust value.

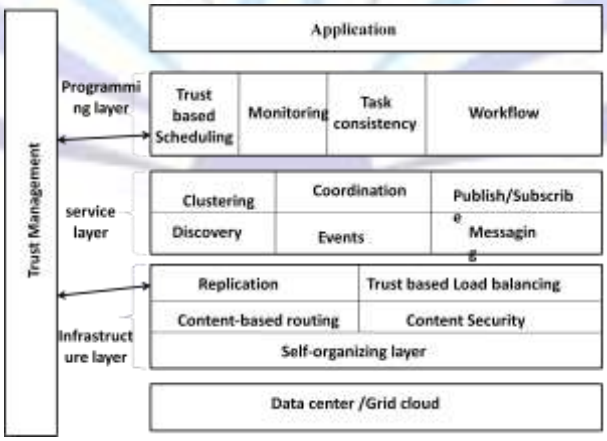

Fig 3:Figure 1 Proposed Trust Model

Whole cycle includes two steps trust Initialization and trust updation or evolution

1) Initialization.

In this the trust value of the datacenter and the client are being initialized.

1.First the datacenter trust is initialized based on the trust management model which is used to initialize the trust value on the basis of parameters explained above and updating the trust value periodically. In this if the datacenter is newly introduced, it is initialized with the default trust value. Here all the parameters are initialized i.e. the initialization time which is calculated depending on the time taken by a datacenter to initialize a default request which is generalized, But MIPS remain the same but it help to set the default trust value because if a datacenter has high MIPS then it should have high initial trust. Fault rate is initialized with default value for all datacenters.

2.From user a request along with the user-budget is been taken. Then the cost estimation for the request is done for all the corresponding datacenters. On the basis of the user-budget and the user-request cost for each datacenter classification of datacenters is done. This set of datacenters is then ordered based on their trust value i.e. their trustworthiness and reliability. Here client trust is simply the budget of the client.

Initial_Trust=Default_Value + CPU (GHz)

Cpu (GHz) = This is the cpu cycle (clock rate).

Default_trust = This is the predefined default value

2)Trust Evolution

This step includes updating the datacenter trust value periodically after a fixed interval of time. To monitor the behavior of the datacenter on the basis of their fault rate and initialization time because these parameters changes as the load on the datacenter increases or decreases [14]. And along with these parameters cost also varies [14].

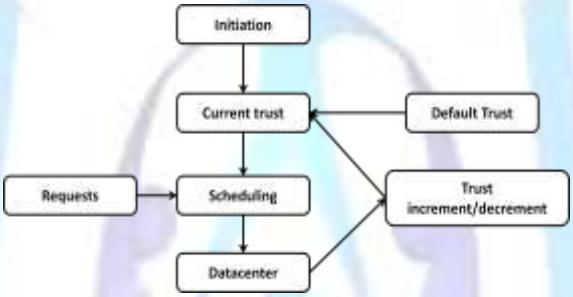

Figure 2 Trust resource scheduling

Let there be datacenters Di, Ntrust (new trust value), Ofault (old fault), Nfault (new fault), Oinitialize (old initialization time) and Ninitialize (new initialization time). and there corresponding trust value be Ti.

Ti, i=1....k,

The value to be updated in trust is calculated on the basis of following equation given by:

Ntrust = (Ofault-Nfault) + (Oinitialize-Ninitialize)

Ti = Ti + Ntrust

f new fault rate greater then old and new initiation time is greater than old. Then calculated Difference results to be positive then we increase the trust by Diff and vice versa. To calculate the cost we take into consideration follows: price of memory, RAM, CPU core, Input data transfer prize, Output data transfer prize.

Based on these trust values of datacenters two lists of trusted and un-trusted datacenters are created. Trusted list consist of datacenters with height processing power and good performance in term of fault or error

B. Scheduling Algorithm

Scheduling is done using the prioritized list of datacenters and client trust for the resources allocation. Before the scheduling were done on the non-prioritized list of datacenters only based on cost. In this first the trust of user is checked who has made the request whether trusted or not, if trusted then find resources in trusted datacenter list otherwise allocate resources in un-trusted datacenter list.

C. Load Balancing Algorithm

Load Balancing is done using the prioritized list of datacenters and client trust. When Load balancing is started List of trusted and un-trusted datacenters/nodes is done. Trusted list consist of nodes having trust value grater then the threshold value in decreasing order i.e. the first node of list has the highest trust value. Similarly un-trusted node list consist of node with trust value less then threshold value in decreasing order.

Algorithm works as follows:

1)For un-trusted user.

When a node gets over loaded then first the VM (Virtual machine) to be migrated is selected. VM with user lowest trust value is selected i.e. user having lowest trust value and having virtual machine in that node is selected. Check whether the

selected VM is of public user or private user. If public the node to which VM is to be migrated is selected from un-trusted list. Selection is done as follows. First node of un-trusted list is checked if it is over loaded or not, if it is over loaded the move to next node in the list otherwise migrate the VM to the selected node. If no datacenter from un-trusted list could be selected then trusted node list is checked but in reverse order i.e. first the node with lowest trust value in trusted node is checked.

2)For trusted user.

When a node gets over loaded then first the VM (Virtual machine) to be migrated is selected. VM with user lowest trust value is selected i.e. user having lowest trust value and having virtual machine in that node is selected. Check whether the selected VM is of public user or private user. If private the node to which VM is to be migrated is selected from trusted list. Selection is done as follows. First node of trusted list is checked if it is over loaded or not, if it is over loaded the move to next node in the list otherwise migrate the VM to the selected node. If no datacenter from trusted list could be selected then un-trusted list is checked.

D. VimCloud Virtualization Platform

This is an open source computing platform developed to deliver infrastructure-as-an-service. As an open source virtulization management platform written in java . It is highly scalibile and good at managing high scale virtual machine clusters .it supports VMware, Qemu ,KVM ,XEN server and VitualBox. It is an web based Virtual machine manager and cloud controller. It uses trust based cloud service model , trust based scheduling and load balancing algorithm been proposed above. It hall all the basic features of Virtual machine monitor.

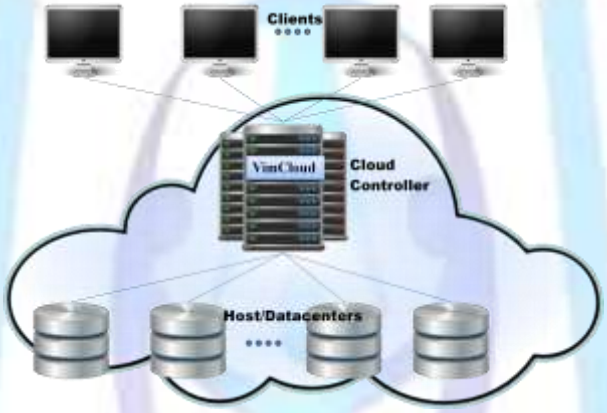

Fig 3 VimCloud Infrastructure

VimCloud in based on a modular design approach, where the role of each module is clearly defined as an chain , all the modules are connected with each other and collaboratively form an framework for cloud computing.

The main modules or the main architectural components are, Cloud controller (CC), Node controller (NC), Request controller (RC), Scheduler (SC), Load balancer (LB), Log manager (LM).

1)Cloud controller

Cloud Controller is the web interface to the cloud. The CC provides users with a set of front-end tools to request services from the cloud. CC keeps track of the cloud resources, forwards user request to the request controller. The user is only allowed to request or view its request and cannot interact with NC.

2)Node Controller

The Node Controller is used to control and manage connection with the nodes or the datacenters which provide the virtualization support. It main aim is to manage connection with all the nodes, add or delete node at any point of time. It provides the list of active nodes that are connected to the server and are ready to serve to the Scheduler and Load balancer. It also gathers information about the current state of a VM. It takes care of any fault that occurs during the creation or running state of a VM.

3)Request controller

The Request controller manages the request that are requested in the form of a queue and check whether the user can request that much resource or not, if not then reject the request otherwise forward the request to the scheduler.

4)Scheduler

Here the main scheduling takes places that is been discoursed above. Scheduler pick a request from request queue and is been scheduled according to the algorithm.

5)Load balancer

The Load balancer is used to keep a check over the load on each node, whether all nodes are equally loaded or not and manage the load using above discoursed algorithm.

6)Log manager

Log manager keep a Log of each and every activity of all the modules in form of log files, even the errors that occur and the request failures. The proposed trust model is tested and verified on this cloud platform and compared with previously proposed algorithms along with scheduling and load balancing algorithms.

IV. EXPERIMENTAL RESULTS

In this for testing the performance of the proposed trust model, it was implemented in a real cloud IaaS environment using 'Qemu' [10] as the VMM (Virtual machine monitor), libvirt to instruct with VMM. VimCloud is tested with 4 datacenters, 500 and 400 user requests. With three trusted and three un-trusted datacenters. In Figure 4 inside refers to total count of trusted user requests allocated to trusted datacenters and total count of un-trusted user requests allocated to trusted datacenters.

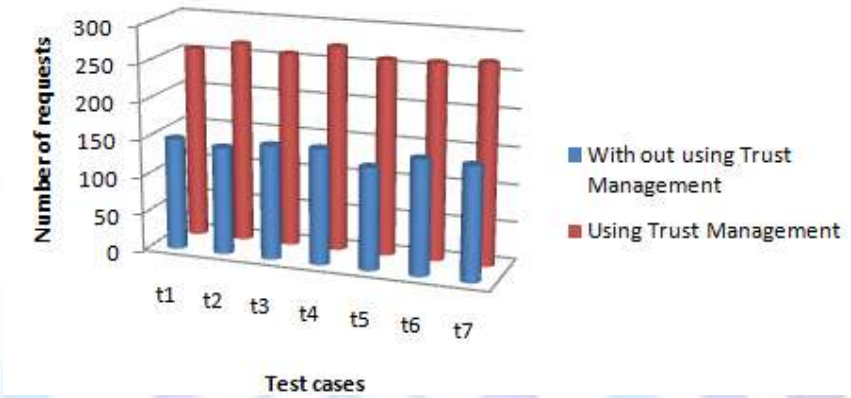

Fig4 Comparison of Total count of un-trusted user requests allocated to trusted datacenters with and without using Trust Management.

In this paper we have presented a design and development of a trust based cloud service model known as VimCloud. It guarantees to the trusted user will get high QoS by running their VM on trusted datacenters with high reliability. For future work this trust model may be compared with other models and see the improvement in the QoS.

V. CONCLUSION

In this paper different type of attacks, IDS models have been discussed with their characteristics and drawbacks. To overcome the drawbacks, a Behavior based IDS is proposed which performs better than other previously proposed models in terms of security provided to the user from different attacks. For future work proposed IDS may be merged with other models and observations be made regarding further improvement in the security.

REFERENCES

[1] Randles, M.; Lamb, D.; Taleb-Bendiab, A," A Comparative Study into Distributed Load BalancingAlgorithms for Cloud Computing",In Advanced Information Networking and Applications Workshops (WAINA) 2010 ,pp: 551 – 556.

[2] F.Azzedin,M.Maheswaran ," Toward trust-aware resource management in grid computing systems ",In cluste computing and the grid ,2002 ,pp. 452,may 2002.

[3] T.Grandison ,M.Sloman "Trust management formal techniques and system",In proceeding of second IFIP conference ,2002.

[4] W.Li ,X.Wang,Fu,"Study on several trust models in grid environment ",journel of Fuzhou university natural science edition , pp. 189-193,2006.

[5] Wenjuan Li ,Lingdi Ping," Trust Model to Enhance Security and Interoperability of Cloud Environment" ,In Lecture Notes in Computer Science, 2009, Vol. 5931,pp.

[6] Wenjuan Li, Lingdi Ping, Xuezeng Pan," Use trust management module to achieve effectivesecurity mechanisms in cloud environment ",In Electronics and Information Engineering (ICEIE), 2010,vol. 1,pp. V1-14-V1-19.

[7] Azzedin.F,Maheswaran,M,"Towards Trust-Aware Resource Management in Grid Computing Systems", In Cluster Computing and the Grid, 2002.pp. 452,may 2002.

[8] Selvi, S. Thamarai ,Balakrishnan, P,Kumar, R.,Rajendar, K,"Trust Based Grid Scheduling Algorithm for Commercial Grids",In Conference on Computational Intelligence and Multimedia Applications, 2007,vol 1.pp. 545-551,dec 2007.

[9] Papalilo, E.Freisleben, B.," Managing Behaviour Trust in Grids Using Statistical Methods of Quality Assurance",In Information Assurance and Security, 2007.pp 319-324,Sept 2007.

[10] Eucalyptus Public Cloud. http://open.eucalyptus.com.

[11] Daniel Nurmi, Rich Wolski, ChrisGrzegorczyk, Graziano Obertelli,Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov," The Eucalyptus Open-source Cloud-computing System",In Proceedings of Cloud Computing and Its Applications,chicago,Illinois.

[12] libvirt virtualization API. http://www.libvirt.org/.

[13] OpenNebula Open Source Toolkit for Cloud Computing . http://opennebula.org/.

[14] Yang, Zhi; Yin, Changqin; Liu, Yan,"A Cost-Based Resource Scheduling Paradigm in CloudComputing",In PDCAT, 2011,pp. 417-422,Oct 2011.

[15] Selvarani, S., Sadhasivam, G.S.,"Improved cost-based algorithm for task scheduling incloud computing",In Computational Intelligence and Computing Research (ICCIC), 2010,pp.1-5 .Dec 2010.

[16] Murata, Y.,Egawa, R.; Higashida, M.,Kobayashi, H.," History-Based Job Scheduling Mechanism for theVector Computing Cloud",In Applications and the Internet (SAINT), 2010,pp. 125-128 ,July 2010.

[17] Huang Qi-yi, Huang Ting-lei," An optimistic job scheduling strategy based on QoSfor Cloud Computing",In Intelligent Computing and Integrated Systems (ICISS), 2010,pp 673-675 ,Oct 2010.

[18] Chenhong Zhao,Shanshan Zhang, Qingfeng Liu,Jian Xie, Jicheng Hu,"Independent Tasks Scheduling Based on GeneticAlgorithm in Cloud Computing",In Wireless Communications, Networking and Mobile Computing, 2009.pp. 1-4 .Sept 2009.

[19] Ratan. M ,A Jaiswal," Ant colony Optimization: A Solution of Load balancing in Cloud", International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012.

[20] Ali M. Alakeel,"A Guide to Dynamic Load Balancing in Distributed Computer Systems", IJCSNS International Journal of Computer Science and Network Security,VOL.10 No.6, June 2010.

[21] Kai Zhu; Huaguang Song; Lijing Liu; Jinzhu Gao; Guojian Cheng,"Hybrid Genetic Algorithm for Cloud Computing Applications"In Services Computing Conference (APSCC), 2011,pp: 182 – 187.

[22] Kun Li; Gaochao Xu; Guangyu Zhao; Yushuang Dong; Wang, D," Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization",In Chinagrid Conference (ChinaGrid), 2011,pp:3-9.

[23] David Escalante and Andrew J. Korty," Cloud Services: Policy and Assessment", EDUCAUSE Review, vol. 46, no. 4 ,2011.

[24] Zing Yang,Changqin Yin,Lui Yang, "A cost-based resource scheduling paradigm in cloud computing"in PDCAT ,2011-12 ,pp. 417-422, january 2011.

[25] C.Castelfranchi,"Trust mediation in knowledge management and sharing ",In procedding of second international conference on trust management 2004 ,pp. 304-318 ,2004

[26] OpenNode .www.opennodecloud.com

[27] CloudStack. www.cloudstack.org

[28] CloudSigma. www.cloudsigma.com

[29] M. K. Goyal, A. Aggarwal, P. Gupta, and P. Kumar, "QoS based trust management model for Cloud IaaS," 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, pp. 843–847, Dec. 2012.