The Performance Evaluation of Proactive Fault Tolerant Scheme over Cloud using CloudSim Simulator

Deepshikha Goutam
Computer Science & Engineering, CSE
GGITS, Jabalpur
goutamdeepshikha@gmail.com

Ashok Verma
Dept. of CSE
GGITS, Jabalpur
vermaashok33@gmail.com

Neha Agrawal
Dept. of CSE
GGITS, Jabalpur
neha2006srit@gmail.com

Abstract— **The main issues in a cloud based environment are security, process fail rate and performance. Fault tolerance plays a key role in ensuring high serviceability and reliability in cloud. Nowadays, demands for high fault tolerance, high serviceability and high reliability are becoming unprecedentedly strong, building a high fault tolerance, high serviceability and high reliability cloud is a critical, challenging, and urgently required task.**

A lot of research is currently underway to analyze how clouds can provide fault tolerance for an application. When the number of processes are too many and the virtual machine is overloaded then the processes are failed causing lot of rework and annoyance for the users. The major cause of the failure of the processes at the virtual machine level are overloading of virtual machines, extra resource requirements of the existing processes etc.

This paper introduces dynamic load balancing techniques for cloud environment in which RAM/Broker (resource awareness module) proactively decides whether the process can be applied on an existing virtual machine or it should be assigned to a different virtual machine created a fresh or any other existing virtual machine. So, in this way it can tackle the occurrence of the fault. This paper also proposed a mechanism which proactively decides the load on virtual machines and according to the requirement either creates a new virtual machine or uses an existing virtual machine for assigning the process. Once a process is complete, it will update the virtual machine status on the broker service so that other processes can be assigned to it.

Index Terms— Cloud, Fault Tolerance, Data Center, Data Center Broker, Virtual Machines

I. Introduction

Cloud computing means to provide the resources whether it can be storage, networks, server, application, platform everything is provided on demand. It is based on pay per use, if the customer is using the resources he or she has to pay if customer is not using the resources then there will be no charge against it. That's why we are calling it to service and there is charge of service, only if you are taking the service.

According to NIST "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."There are three Cloud Services Models and these three fundamental classifications are often referred to as "SPI model" i.e. software, platform or infrastructure as a service.

SaaS is capability in which the consumer can use the provider's applications running on the cloud. Platform as a service, the consumer can deploy, the consumer created or acquired applications created by using programming languages or tools provided by provider, on the cloud infrastructure. IaaS is a capability provided to the consumer by which, it can provision processing, storage, networks and other fundamental computing resources where the consumers can deploy and run the software (i.e. operating systems, application). There are four deployment models in cloud.

The cloud infrastructure is available to the general public is public cloud. The type of the cloud that is available solely for a single organization is called private cloud. The cloud deployment model, the infrastructure of the cloud is shared by several organizations and supports a specific community with shared concerns is called community cloud. Hybrid cloud is a cloud infrastructure that is a composition of two or more clouds i.e. private, community or public [1, 2].

Fault tolerance is a property of a system that enables it to deliver the expected service (potentially with degraded performance) in the presence of one or more faults, or component failures. Fault tolerant systems are systems which exhibit the ability to tolerate faults that may lead to system failure. This is an essential part to improve the performance of cloud platform, thus this function should be considered in clouds. When users transfer their critical systems to clouds, can the cloud serviceability achieve 100 % uptime is a question users always ask. Unfortunately, cloud serviceability and fault tolerance are still far from being perfect. Failures are normal rather than exceptional in cloud computing environments, due to large-scale time-critical data support, and because cloud platforms usually run in the form of voluntary, much cheaper, less powerful and virtual computing nodes, cloud nodes are usually connected by unpredictable communication links, thus communication failures, such as time out, will greatly influence the serviceability of clouds, and some malicious behaviors occur in clouds as user contributed nodes.

To achieve efficient fault tolerance enhancements, significant cloud serviceability improvement, and great reliability, a foolproof fault tolerance strategy on the voluntary virtual cloud infrastructure should be employed. In most current

978-1-4799-2259-14/$31.00©2014

clouds, check pointing is the process of saving system states periodically during failure-free execution, and replication the process of replicating hot data, usually to stable storage, are the two most common fault tolerance strategies. By employing the check pointing fault tolerance strategy, if a failure does occur while a system is running, the system can roll back to the latest checkpoint and restart again from this checkpoint, thereby bounding the amount of lost operations to be recomputed. By employing the replication fault tolerance strategy, replicating the popular data to multiple suitable locations, users can access the data from a nearby fault-free site. Due to high level of complexity and the voluntary virtual infrastructure nature of clouds as compared to other large-scale distributed computing environments, existing fault tolerance strategies are not sufficient to manage failures in clouds[9]. According to cloud architecture, we identify three types of failures in a cloud platform: Hardware failure, VM failure and application failure. In application failure the execution of an application will fail. In VM failure, the VM, created in physical host, fail to produce correct result. In Physical machine failure, the hardware fails to provide continuous operation [6].

There are various techniques through which Fault Tolerance can be handled. The reactive FT policy is to allow the occurrence of fault, error and failure and then tackle the fault and provide service continuously. It reduces the effect of failure on application execution when the failure effectively occurs. Check pointing/Restart is a reactive fault tolerance in which, when a task fails, it is allowed to be restarted from the recently checked pointed state rather than from the beginning. It is an efficient task level fault tolerance technique for long running applications. Another reactive FT technique is data replication in which various task replicas are run on different resources, for the execution to succeed until the entire replicated task is crashed[10,12].

The proactive fault tolerance policy is to avoid the occurrence of fault, error and failure by predicting them and proactively replace the suspended component means detect the problem before it actually come. Software Rejuvenation is a proactive FT technique that designs the system for periodic reboots. It restarts the system with clean state. Another proactive FT technique is Self Healing in which When multiple instances of an application are running on multiple virtual machines, it automatically handles failure of application instances [10, 12].

II. Related Work

Author [3] has proposed the model which incorporates that system tolerates the faults and makes decision on the basis of reliability of the processing nodes i.e. VMs. According to his model, VM reliability is adaptive which changes after every computing cycle. If a VM produce a correct result within the time limit, it's reliability increases and if it fails to produce a result within time or correct result, it's reliability decreases. If the node continues to fail, it is removed and a new node is added.

In cloud environment, author [4] has found that there are many obstacles for adoption and growth of cloud computing. To tackle these obstacles author has proposed various solution.

In this work author has presented only theoretical or conceptual way to tackle these problem and fault tolerance is a factor which affects the performance predictability factor of this given table in cloud computing. Author [5] has design the framework for creating and managing FT that shades the implementation details of the reliability techniques from the users by means of dedicated service layer. This scheme allows the users to specify and apply the desired level of FT without requiring any knowledge about its implementation. Author [6] has analyzed the concept of automatic repair in the implementation of FT in cloud computing environment. In automatic repair, he used collaborative approach to solve FT in cloud computing. Collaborative approach constitutes an interesting tradeoff between exclusive management by the provider and exclusive management by the customer. In cloud computing platform author [7] has focused in another research area that what will be happened, when we run high performance computing application over the cloud and FT is one of the problem which impact the performance of cloud infrastructure to run HPC application. He also focused on the various challenges for HPC application in the cloud. Author[8] has proposed the scheme in which execution of HPC application is performed on virtual cluster and on these virtual cluster VMs are placed and these clusters are interconnected with infiniband network. To understand infiniband network is complex, not easy as Ethernet network, that's why it is not popular as Ethernet and live migration of virtual machine in an infiniband based configuration is still an unsolved problem. High fault tolerance and high serviceability issues in cloud are addressed by author [9]. They analyzed, modeled and evaluate the performance of dynamic fault tolerance policy, with the use of cloudsim simulator. They analyzed the mathematical relationship between different failure rates and different fault tolerance policies like checkpointing and data replication. They modeled DAFT (dynamic adaptive fault tolerance) which is a combination of dynamic adaptive checkpointing fault tolerance and dynamic adaptive data replication fault tolerance policy and evaluate these DAFT under various conditions in large scale cloud data center. Various fault tolerance techniques and research challenges in fault tolerance techniques, tools to implement these fault tolerance techniques are covered by author [10]. They proposed cloud virtualized system architecture based on HAproxy, through which fault tolerance will be automatically dealt. For this automatic dealing form fault, data replication mechanism is used. The live migration in VMs brings several benefits such as fault tolerance, improved reliability, improved throughput in cloud computing. Author [11] evaluates the effects of live migration of virtual machines on the performance of an application running inside VMs. They found that migration is acceptable and workable but some where it downs the service level agreement and if the migration is performed so some where it affects the cost. Authors [12] discussed about the fault taxonomy and need of fault tolerance and compare the various fault tolerance strategies.

III. Proposed Jobs and Its Algorithm

A number of fault tolerant frameworks, strategies and policies that allow the applications to run against failure have been proposed, but there is lot of research room is available to cope up with the fault, for an application running on cloud infrastructure.

When numbers of processes are too many and any virtual machine (VM) is overloaded then the processes are failed cause lot of rework and annoyance for the users. The major cause of failure of the process at the VM is the overloading of VMs.

An approach is proposed, named as dynamic proactive fault tolerant technique over cloud platform. This Proposed approach is implemented with the use of cloudsim simulator.

In this approach fault tolerance is a proactive mean simply avoid the occurrence of fault when an application or process run or submit inside the VM. It proactively decides that on which VM, which applications are to be run, means the process can be applied on an existing VM or it should be assigned to a fresh newly created VM. These VM will be placed inside a host and host will be placed inside a data center. On which host, which VM will be placed, totally depends on the free resources of host. If the host has much amount of free resources, the VM will be placed on that host as compare to the other host, and after creating host in datacenter and VM in host and after submitting the applications on VMs and after the completion of applications, VMs will be automatically destroyed.

To implement this technique cloudsim simulator is used. It provides certain actors which perform the action according to their function.

Actors which perform actions:

1. Cloudlet Generator- Cloudlets are the processes or applications that run on cloudsim simulator. Cloudlet generator will generate the number of cloudlets and submit all the cloudlet list to the broker.
2. VM Generator- It will generates the number of VMs in cloudsim simulator and submit all the VMs to the broker.
3. Datacenter (DC) - It is a resource provisionars, which contains one or more than one host.
4. Host- Physical machine on which logical machines will be placed.
5. Virtual Machine(VM)- Logical machine on which application will be run.
6. Broker- Broker is a agent, who will submit the VMs in specific host and then bind cloudlets, applications, processes on specific VM and after completion of all the cloudlets, free VMs will be automatically destroyed.

To implement this work two algorithms are proposed.

ALGORITHM 1

1. Use of cloudsim simulator
2. We have default host an default virtual machine on it
3. Create a datacenter Datacenter datacenter0=create Datacenter("DATACENTER_ID")
4. Create the number of host on datacenter, on which the virtual machines will be allocated and on that VMs the cloudlets (Process) will be executed. These hosts have fixed resources like fixed RAM, Fixed MIPS etc.
5. Creation of cloudlets
   Cloudlet is created with the use of cloudlet generator defined in cloudsim. In our work, we have created 100 cloudlets, each has different size. The size of cloudlet will be varied with the use of random generator created by us.
   ```
   Int  getRandom(Max)
   {Random random= new Random ();
   int rnd = 0;
   while(rnd<150) {
   rnd = random.nextInt(val);      /* return uniformly distributed int value between 0 and Specified value.*/
   }
   Return rnd }
   ```
6. After the creation of cloudlets ,all are submitted to the cloudlist data structure.
   CloudletList.add(cloudlets)
7. Creation of VMs from VM generator defined in cloudsim simulator.In our work, we have created 10 instances of VMs, each has fixed mips, fixed RAM, fiexd bandwindth etc.
8. After the creation of VMs, all are submitted to the vmlist data structure.
   vmlist.add(VMs)
9. Creation of broker in cloudsim environment.
   DatacenterBroker broker=createBroker()
10. Cloudlet list will be submitted to the broker.
    broker.submitCloudletList(cloudletList)
11. VM list will be submitted to the broker.
    Broker.submitVmList(vmlist)
12. In next step, broker has to allocate the VM in a host such that host can accommodate and execute the cloudlets. On which host, VM is allocated, is described by MyVmAllocationPolicy() function, presented in Algorithm 2.
13. After allocating VMs in specific host, broker will bind the cloudlets into VMs.
    bindCloudletToVm(cloudlet,VM_i).
    To bind cloudlets into VMs in following way
    Cloudlets0 will be executed by the VM0, cloudlet1 will be executed by VM1......Cloudlet9 will be executed by VM9, again Cloudlet10 will be executed in VM0 etc.

After the successful execution of cloudlets, output is displayed and no cloudlet will go in trap. The cloudlets are submitted to that host, which have sufficient resources (freepes, freeRam) to execute the cloudlet. With the use of this approach, we can avoid the occurrence of fault in the execution of processes, called proactive fault tolerance.

14. After running the cloudlets, VMs will be de allocated by broker with the use of MyNewVmAllocation policy () function.

ALGORITHM 2
MyNewVmAllocation Policy

This policy shows that on which host, vm will be allocated and after allocation ,cloudlets can run on it.
VmAllocation Policy Simple() is allocation policy provided by cloudsim simulator. And from this policy host is found which has less Pes in use.In our work, we modified it with salt of RAM.

1. Find out the host which is underutilized.
2. If (underutilizedHost)
   then VmAllocation
   else
   wait to find underutilized host
3. Underutilized host is found in two way
   A) In the context of free pes.
   B) In the context of free RAM.
      Let the host H1 which has less pes in use. So in this case H1 is underutilized and host H2 has RAM is in use. S, in this context H2 is underutilized. so we wil find the difference of mips of the host and difference of RAM of the host. It may be possible that one common host has less pes in use and less RAM in use. Since the pes are in MIPS and RAM are in KB,MB. So, use this formula to find out the value for the comparisons.

Diffmips=(freemips(H1)/(freemips(H1)+freemips(H2)))*.5

DiffRAM=(freeRAM(H2)/(freeRAM(H2)+freeRAM(H1)))*.5
   If(H1==H2 || Diffmips > DiffRAM)
   Then
   VMAllocation on host H1
   Else
   VMAllocation on host H2
4. How to find free pes and free RAM.
   getfreePes() and getfreeRAM() function defined in cloudsim simulator.
5. After the allocation and execution of cloudlets VM will be destroyed.
   host.vmDestroy(vm);

These algorithms will help to avoid the occurrence of fault in cloud based environment.

IV. Experimental Set up and Results

We present tests and evaluation that we undertake in cloudsim simulator. Cloudsim is a new generalized and extensible simulation framework that enables seamless modelling, simulation, and experimentation of emerging Cloud computing infrastructures and management services. The simulation framework has the following novel features:
support for modelling and instantiation of large scale Cloud computing infrastructure, including data centers on a single physical computing node and java virtual machine, a self-contained platform for modelling data centers, service brokers, scheduling, and allocations policies, availability of virtualization engine, which aids in creation and management of multiple, independent, and co-hosted virtualized services on a data center node, flexibility to switch between space-shared and time-shared allocation of processing cores to virtualized services.
These compelling features of CloudSim would speed up the development of new algorithms, methods, and protocols in Cloud computing, hence contributing towards quicker evolution of the paradigm.
The tests were conducted on a i3 machine having configuration: 2.20GHz with 1MB of cache and 2 GB of RAM running a standard windows 7, JDK 1.7 and netbeans IDE 7.0.1.
To evaluate the overhead in building a simulated Cloud computing environment that consists of a single data center, a broker, three hosts and a user, we perform series of experiments on this environment and find out the result.

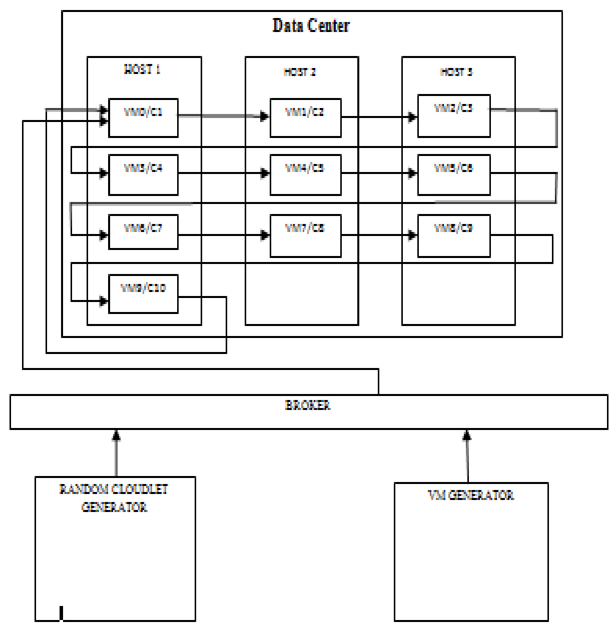

Figure 1 Diagrams of Proposed Algorithms

In this approach 100 cloudlets are created and they are submitted on the VMs on the basis of the proposed algorithms. The result is shown in the form of graph and these graphs are made on basis of result generated by cloudsim simulator.

RESULT 1- this graph shows the relationship between Cloudlets and completion time. It shows that all cloudlets will be executed no one is left for execution.

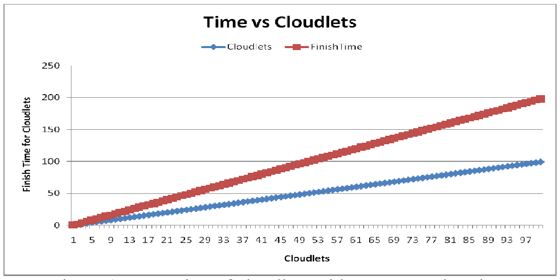

Figure 2. Execution of cloudlets with respect to the Time

RESULT 2- This graph shows the relationship between Cloudlets and VMs. It shows that all cloudlets submitted to the VMs in the order of proposed algorithm and all Cloudlets are executed in the VMs, no cloudlet will remain to submit and completion.

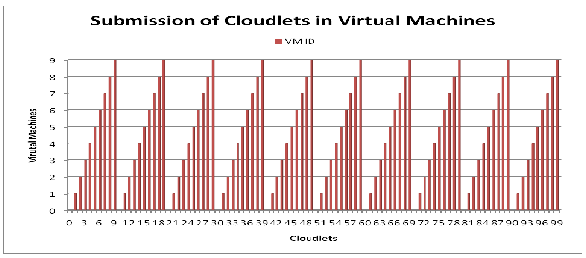

Figure 3. Execution of cloudlets in VMs

RESULT 3- This figure shows the relationship between VMs and time require executing the applications. It totally depends on the capacity of VMs and resource requirement of applications.

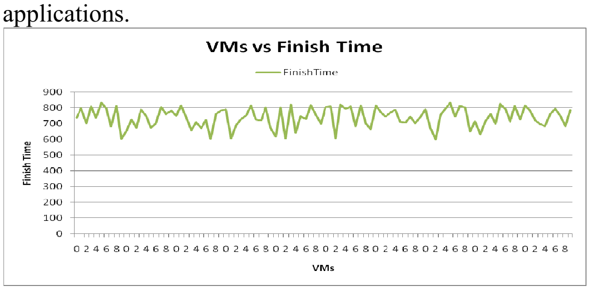

Figure 4. Time to execute the cloudlets by VMs

RESULT 4- Figure3 shows the relationship between the Cloudlets and their size. And figure 4 shows the relationship between Cloudlets and the MIPS they require.

From this we analyze that each cloudlets has different file size and different MIPS so it requires different time for completion but no one will be stuck to execute.

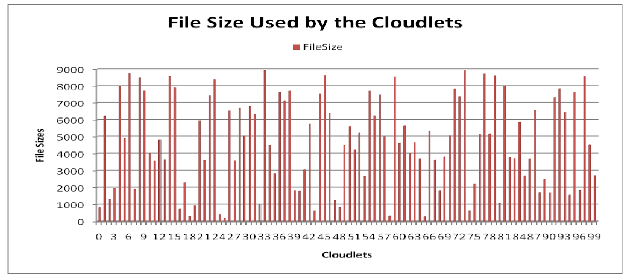

Figure 5. The different size of the cloudlets

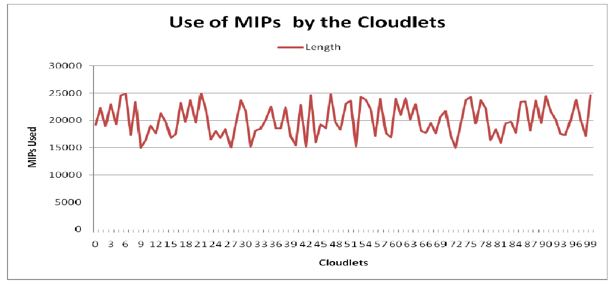

Figure 6. The total length of cloudlets

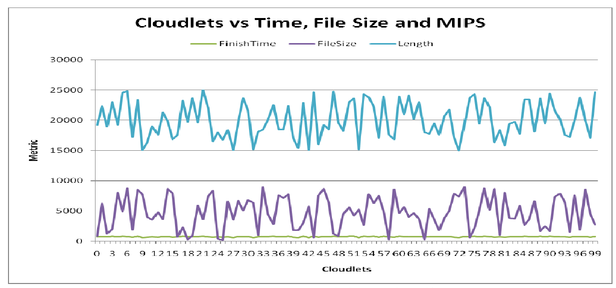

Figure 7. Metric Model vs Cloudlets

It is shown from the figures that all the cloudlets, applications; processes are completing their execution without any fault. So the throughput of the system will be automatically improved.

V. Conclusion

In this work, proactive fault tolerance is achieved, so that application failure will be tackled in cloud based environment. When the applications are submitted they will be completed by the VMs in cloud data center and creation of VMs in the host is dynamic that is based on the free source of host if the host has free resources then create the VMs in that host so that underutilized host will be utilized and over utilized host will continue its work.

This proactive fault tolerance is achieved by focusing on proactive load balancing over the cloud for managing the processes and virtual machines so that, the waste of resources can be avoided and users will not be finding delays due to their process failures over the cloud computing environment.

Proactive management of the processes over cloud will increase the performance of the cloud and will avoid unwanted delays caused by the processes running over the system.

VI. Future Work

The proposed work has been implemented on simulation environment using standard machines, in future the same can be deployed over the real cloud environment and test it for its accuracy and performance.

A further improvement in the proposed algorithm may be required for real high performance computing environments over the cloud in which the computation intensive applications or any scientific application will be executed.

At IaaS and PaaS layers changes may be helpful in increasing the performance of the proposed system.

In this approach, the arrangement of VMs and the submission of processes in VMs are according to proposed algorithms so that no VMs will be overloaded and execution of the processes will be successfully completed. But what will be happened when the VM is running the applications and suddenly it get crashed because of unwanted interrupt or it may be happened that, that host will be crashed on which the VMs are placed.

The aforementioned issues can be implemented in future as well.

VII. References

[1] International Journal of Electronics and Computer Science Engineering (IJECSE) ISSN 2277-1956 Volume1,Number 3 "Current Trends in Cloud Computing : A Survey of Cloud Computing Systems.

[2] Cloud computing: state-of-the-art and research challenges, J Q. Zhang · L. Cheng · R. Boutaba, JInternet Serv Appl (2010) 1: 7 18 DOI 10.1007/s13174-010-0007-6

[3] Sheheryar Malik and Fabrice Huet "Adaptive Fault Tolerance in Real Time Cloud Computing" 2011 IEEE World Congress on Services

[4] Above the Clouds: A Berkeley View of Cloud Computing, UC Berkeley Reliable Adaptive Distributed Systems Laboratory _ http://radlab.cs.berkeley.edu/ 2009

[5] Ravi Jhawar_, Vincenzo Piuri_, Marco Santambrogio "A Comprehensive Conceptual System-Level Approach to Fault Tolerance in Cloud Computing" 978-1-4673-07 5/12, 2012 IEEE

[6] Alain Tchana, Laurent Broto and Daniel Hagimon "Approaches to Cloud Computing Fault Tolerance", 2012 IEEE.

[7] Sanjay P. Ahuja, Sindhu Mani Journal of Emerging Trends in Computing and Information Sciences (CIS) ISSN 2079-8407 vol.3, No 2. Feb2012 "The State of High Performance Computing in the Cloud.

[8] Viktor Mauch, Marcel Kunze, Marius Hillenbrand " High Performance Cloud Computing" 2012 Elsevier.

[9] Dawei Sun.Guiran Chang. Changsheng Miao. Xingwei Wang Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environment Springer 2013.

[10]Anuj bala, Inderveer Chana" Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing" International Journal of Computer Science Issues(IJCSI), Vol. 9, Issue 1, No 1, Jan 2012

[11]William Voorsluys, James Broberg, Srikumar Venugopal, Rajkumar Buyya " Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation".

[12] Prasenjit Kumar Patra, Harshpreet Singh, Gurpreet Singh " Fault Tolerance Techniques and Comparative implementation in Cloud Computing", International Journal of Computer Applications2013

[13] Shuai Zhang, Shufen Zhang, Xuebin Chen, Xiuzhen Huo "Cloud Computing Research and Development Trend" 978-0-7695-3940-9/10, 2010 IEEE DOI 10.1109/ICFN.2010.58

[14] Dawei Suna, Guiran Changb, Lina Suna, Xingwei Wanga"Surveying and Analyzing Security, Privacy and Trust Issues in Cloud Computing Environments." Elsavier volume15, 2011.

[15] Cloud Security Alliance, 2009, p14, www.cloudsecurityalliance.org

[16] Stanoevska-Slabeva and Wozniak "Cloud Computings effect on Enterprises." 2011.

[17] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros"Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities" IEEE 2009.