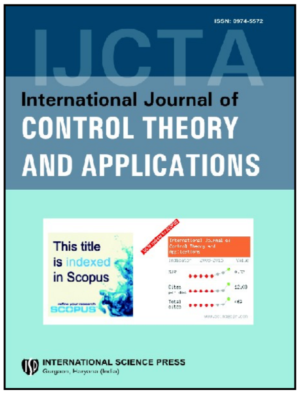

International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 10 • 2017

Proactive Scheduling in Cloud Computing

Ripandeep Kaur[1] and Gurjotkaur[2]

[1] *Department of CSE Chandigarh University, Gharuan, Punjab, India, E-mail: ripansandhu180@gmail.com*
[2] *Assistant Professor Department of CSE Chandigarh University, Gharuan, Punjab, India, E-mail: randhawa789 @gmail.com*

Abstract: Autonomous defect conscious arrange is a quality entirely relevant for cloud computing and it is similar to acceptance of load deviation. In this context, this paper recommend an defect attentive instruction paired autonomous appoint for cloudcomputing based on autonomous computing conception In procedure to approve the recommended explanation, we execute two analysis one with conventionalaccess and alternative with arrangementconcession failing attentive accession. The results displaythe efficiency of the scheme.

Keywords: Fault tolerance, scheduling, performance metrics, cloud computing, QoS.

1. INTRODUCTION

Cloud computing is a recent advancement wherein IT infrastructure and applications are provided as 'services' to end-users under a usage-based payment model. It can leverage virtualized services even on the fly based on requirements (workload patterns and QoS) varying with time[1] According to NIST definition: "Cloud computing (CC) 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 (SP) interaction[2]Cloud service users demand for their services end-to-end QoS assurance, high levels of service reliability, and continued availability to their SPs. Now a days, IT enterprise is adopting cloud computing in order to reduce the total cost involved and also improve the QoS delivered to the customers.

There are no standard metrics or a standard way to ensure QoS to the customers. There are several models or algorithms that are proposed to ensure QoS to the users and proper management of workloads to provide QoS and performance. So in CC, there are various important research issues which need to be focused for its efficient performance is fault tolerance and scheduling[3]

There has been various types of scheduling algorithm exist in cloud computing system. Most of them can be applied in the cloud environment with suitable verifications. The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. During scheduling, existing algorithms are not fully capable to evaluate the fault and take decisions accordingly. Multiple reasons exist for low performance in scheduling algorithms. Majority of literature focussed on the work to decrease response time in order to provide scheduling in the cloud environments with Quality of Service (QoS). Therefore performance is definitely one of the major in using existing scheduling algorithm, but improving performance with enhancing the fault tolerance of the cloud system is one of the major research area which is not been explored very well[4], [5]. To provide guaranteed Quality of Service (QoS) to users, it is necessary that jobs should be efficiently mapped to given resources.

Service Level Agreement (SLA) is the major parameter i.e. considered for assuring QoS and it is responsibility of SPs whether at infrastructure, platform or software level- provide quality guarantees usually in terms of availability and performance to their customers in the form of SLAs. So it should be fault-tolerant, and recovery time should be minimal to avoid SLA violation. The replica should be maintained near the customer's location to reduce the recovery time after any failure or disaster. So SLA should include the availability, response time, and degree of support[6]

This research paper proposes a service ranking algorithm in a CC on the basis of detailed performance monitoring and historical analysis and based on their contribution, a weight age is assign to all service quality factors or performance metrics and as a final point aggregated to compute ranking score (R) of a service by developed formula. This new model is used for VM allocation, re-allocation and placement with consideration of best/high ranked virtual machine/datacenter available. Workload requested by the users under pre-analyzing the job requests and the resource status of the data center considering various parameters like Reliability, Reputation, Network Latency, Processing time, Availability etc.

We further summarize our objectives as under:

- To develop a system that pre assume the time consumption of workload to identify the high ranked VMs/DCs.
- To establish the effectiveness of the system in correspondence to Service Level Agreements (SLAs) Violations. We will be evaluating the impact of faults on scheduling and improving scheduling by minimization of SLA violations.
- To develop a QoS system for users in terms of response time i.e. time taken from the Cloud to respond user's request.

2. LITERATURE REVIEW

Stelios Sidiroglu[7] an ASSURE, a new self-healing software (s/w) based approach that presents rescue points (RP) for detecting/tolerating/recovering from s/w failures in server apps while preserving system availability and integrity. Using fuzzing, they identify rescue point and implemented by checkpoint/restart technique. When fault detects initially, it uses an application replica to find out what RPs can be employed for recover execution of future programs. This approach implemented on various applications of server like proxy servers, domain name, database and web. The main goal of this approach is to healing s/w services automatically from s/w failures that were previously unidentified or not known.

Hai Jin[8] introduce a Help, a novel self-healing s/w based approach which is considering extension of original approach ASSURE that applies error virtualization and weighted RP (WRP) methods which helps server applications to avoid faulty path. It can survive s/w failures and to make sure high availability of service in CC environment. Help presents two approaches. First, WRPs for recover from faults that are complicated to handle for ASSURE. Second, to adopt two-level RP database which helps to share information related to faults with applications that are helpful for further faults recovery.

Sheheryar Malik[9]an AFTRC (Adaptive FT in Real time CC) model. By computing reliability(R) of every VM, a system tolerates faults. After every cycle, reliability of every VM is changed because of its adaptive behavior. A main goal of this model is to assign R weights to every VM and removing/adding a VM, if it is not performing efficient in real time environment. AFTRC also provides backward/forward recovery in case if any VM doesn't achieve minimum reliability level and it also uses replication technique to achieve FT.

Dilbag Singh[10]a smart failover approach for offering high availability to the cloud's customers by using new algorithm namely; integrated checkpointing with load balancing (ICWLB) and to reduce overheads of checkpointing by using multilevel checkpoint. A proposed strategies used two different algorithms namely; global and local checkpointing algorithms. This approach has been made performance comparison of various metrics like Maximum/Minimum Execution time (Max/Min ET), Maximum/Minimum Waiting time (Max/Min WT) with some existing methods and also shows a proposed strategy gives better results than other strategies.

Pranesh Das[11] a smart failover approach namely; Virtualization FT (VFT) to attain the FT by using redundancy or replication technique. They presented a virtualization technique where the Load balancer (LB) distributing loads to those nodes whose related computing nodes have excellent performance history which further measure by using Success rate of those computing nodes. This model helps to decrease timing of services and to improve the availability by decision maker and cloud manager modules.

Deepak Poola[12]a scheduling algorithm to schedule workflow tasks or jobs on CC resources with the help of spot instances (SI) and on-demand instances (ODI) pricing models and also to reduce execution cost in case of tasks deadline. A proposed algorithm is used bidding method to decrease cost and bids according to the requirement of workflow. They also tolerate faults against early extinction of SI and robust against CC instances variations in performance. This work saves cost upto 14% by using checkpointing technique.

Mohammed Amoon[13] proposed an economy based FT framework to maintain monetary profit by providing dynamic number of replicas and to tolerate faults for avoiding failures. A main work presented by two algorithms namely VMC (VM Classification) and FTSS (FT Strategy Selection). VMC classifies cloud VMs by using available information of usage service time and probability of failures VMs and to select most valuable VM that are profitable for cloud. FTSS is basically used for selecting suitable FT approach for selected virtual machine that depends on requirements of customers like time deadline and cost of cloud applications. This framework used various FT approaches like Proactive and Reactive FT and provide hybrid FT. In Reactive, it uses various strategies like checkpointing, replication and further used parallel and multiversion mechanisms of replication strategy.

Anju Bala[14]an Autonomic FT (AFT) scheduling approach to assist the execution of parallel tasks in cloud computing applications like scientific workflows (SW). Cloud Service providers involve well-organized scheduling fault tolerant (FT) and Hybrid heuristics (HH) techniques. HH merges the various features of FCFS, Min-Min and Max-Child heuristic. In FT technique, due to over-consumption of resources if task failure happens then VM migration (VMM) automatically migrates the VM. AFT approach significantly reducing make-span, standard deviation and total mean execution time and improve performance of SW.

Punit Gupta[15]a FLHB Scheduling algorithm for cloud IaaS. It provides higher quality of services to the customer with least cost and also considers various datacenters quality of service parameters like System load (MIPS), Network load, initialization time and Fault Rate for improving the performance and quality of services to the customer in cloud infrastructure environment.

I. System MODEL

In this section our proposed system model which explains Fault Aware Scheduling Technique (FAST), as shown in figure 1 where workload generator is responsible for creating workloads. It is similar to the users who are requesting for VMs. These users defined a set of quality parameters which needs to be met by any system. Therefore a model is required which involved the following steps:

1. A monitoring application collects the following values and after retrieving monitored values fuzzy prediction process is initiated which sets the min and max performance of VM components, i.e: for

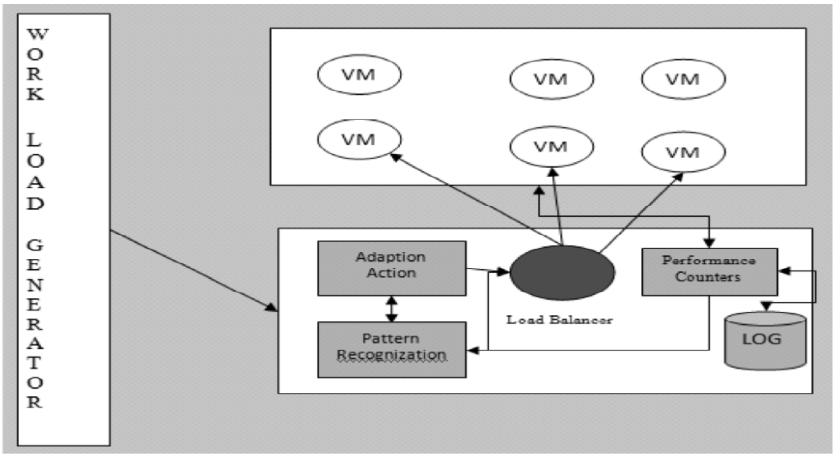

Figure 1: Proposed System Model

each request to process there is requirement for CPU which ranges [CPU_{min} ~ CPU_{max}], similarly for Memory [MEM_{min} ~ MEM_{max}] and for Bandwidth [BW_{min} ~ BW_{max}], after obtaining the degree of truthness for each component of VM (fuzzification) these values get put into LOG.

2. Clustering of VMs: Each VM, with a common set of configuration is put into a common cluster.
3. A commonly used VM allocation policy (Round Robin algorithm) is used to allocate the incoming request to theseclusters.
4. A constant tracking of SLA violations is done and in the event of any positive sign, a pattern of VM working is obtained by comparing the current value with the LOG.
5. The pattern algorithm which is based on density-based spatial clustering [10] which identifies the distribution of data in the current cluster and generates a trigger in the case of any change required. Hence the first SLA violation is acting as the threshold value and is represented by *å*.
6. After getting the first SLA, the process of inputting is started and obtained results arerefuzzified. After this the current performance is logged in. Now this new cluster is used for scheduling new job requests, which is done by identified faulty VMs who are not meeting the requirements of users.
7. After identifying these which helps load balancer to take decision by redirecting the incoming request to the VM who are working up to their capacity and very minimal requests is inflow towards faulty VMs.

Algorithm 1: Clustering

```
For every $VM_i$
$Res_i$= Get_monitored_result(VM);//
                              [$CPU_{min}$ ~ $CPU_{max}$],
                              [$MEM_{min}$ ~ /
                              //$MEM_{max}$],
                              [$BW_{min}$ ~ $BW_{max}$],
LOG($Res_i$);
Set(min,max)$_I$ // Fuzzification
Get_result= Match_Cluster_to_$VM_i$ (Cluster_Name, $VM_i$)
          If (Get_result=TRUE)
                    Set_ $VM_i$(Cluster_Name)
endIf
```

Algorithm 2: Tracking Faults

```
Set_threshold = first_SLA_violation (VMi)
          If (SLA_violation = = TRUE)
          Set(min,max) // ReFuzzification
Get_result= Match_Cluster_to_VMi
(Cluster_Name, VMi)
          If (Get_result=TRUE)
                    Set_ VMi (Cluster_Name)
endIf
```

II. Experimental Set up and Results

Following are the simulation parameters:

Number of Datacenters: 4

Number of Host/DC: 1

Number of VM/Host: 4

Experiment No. 1: In this experiment, a fault is created by lowering the CPU capacity, which directly lowered the CPU consumption. In Figure 2 we can observe the CPU consumption where VM 1 is offering lowered capacity.

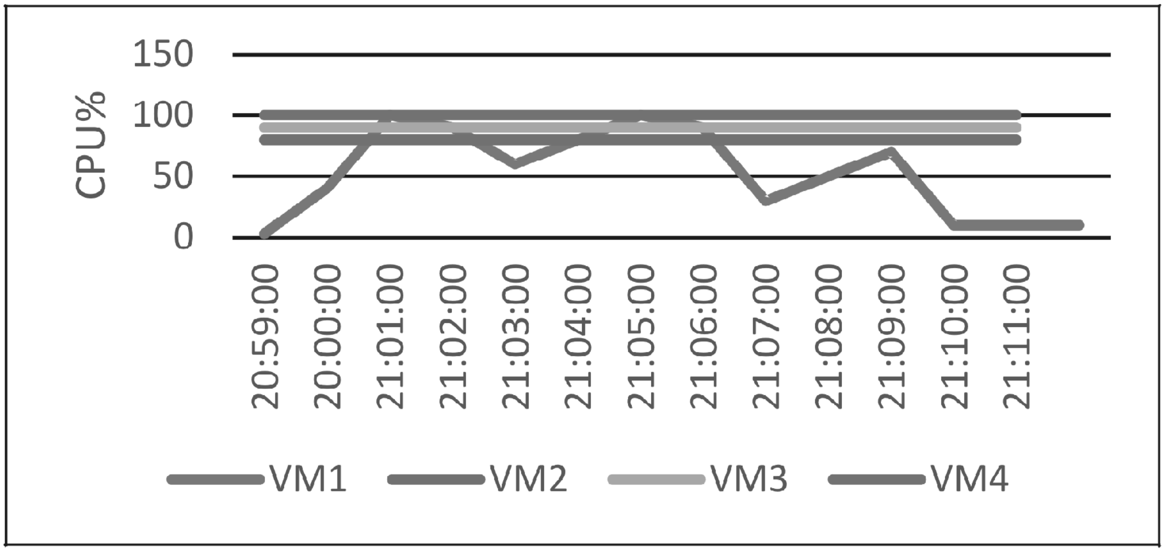

Figure 2: Average CPU consumption

Experiment No. 2: In this experiment we analyze the response time of commonly used round robin algorithm. We can observe that with the introduction of faults the response time is increased drastically.Therefore the average response time i.e. 7.9 ms for the all the requests to process shown in Figure 3 which is very high.

Experiment No. 3: In Figure 4, SLA violation is observed for the round robin algorithm which is very high for VM1.

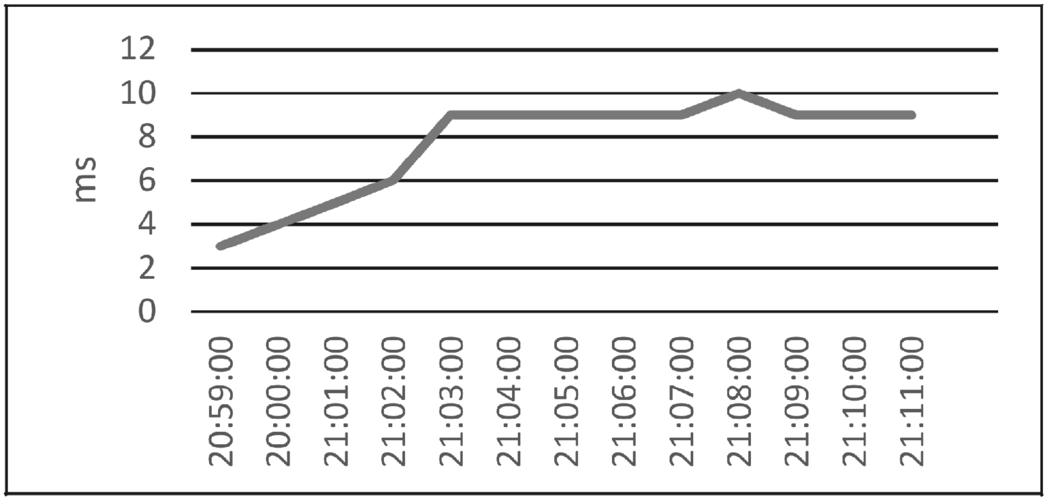

Figure 3: Response time

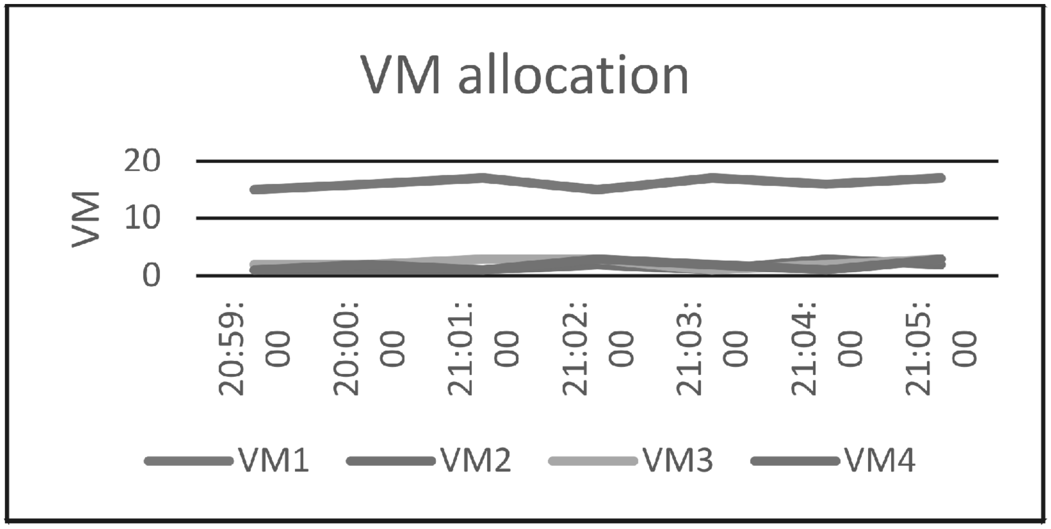

Figure 4: SLA violation for VM allocation policy

Experiment No. 4: In this experiment, again a fault is created to observe the behavior of proposed algorithm (FAST). An average CPU usage for 2 VMs is constant but for third VM it is fluctuating, hence underperforming CPU observed. Hence a optimal requests are being allotted, because the workload was better distributed among the better performing virtual machines and also the response time of requests initially had the lowest values, and most requests were allocated to the VM 2,3 and 4. The increase in workload leads to more allocations to only these VMs. Also the observed Response time in this case is 4.46 ms.

This directly corresponds to the performance of SLA violations as we can observe in Figure 5 all the VMs are performing equally. This is done by moving under performing VM to its right cluster.

Both experiments used the same workload and resource allocation strategy. However, the thresholds were different because of different SLA violations.

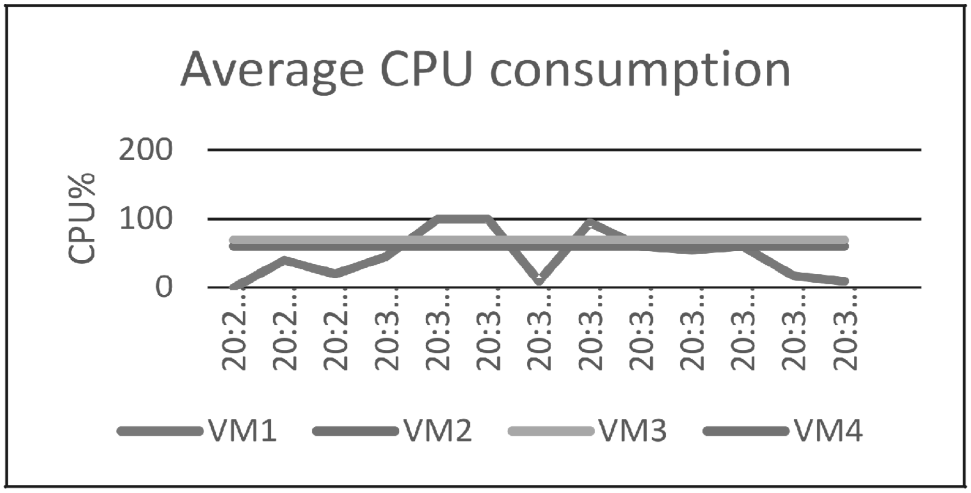

Figure 5: VM Performance Indicator

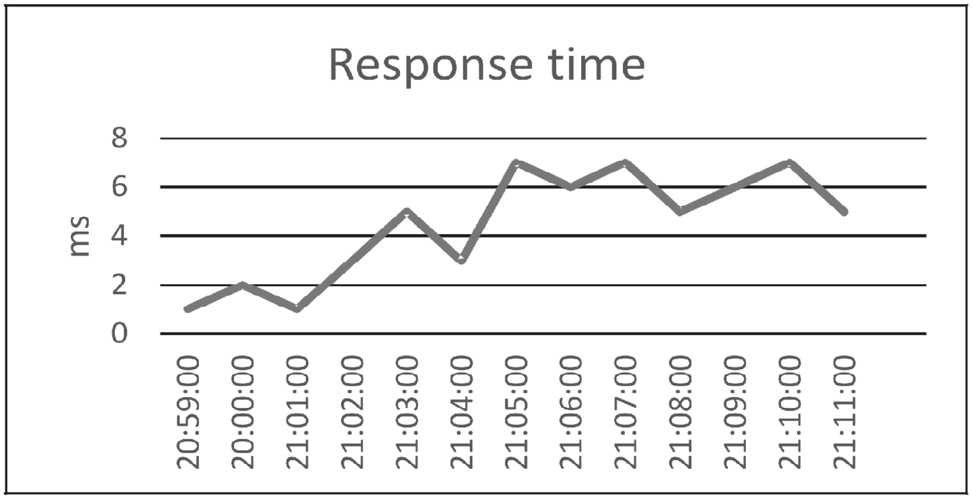

Figure 6: Response time for FAST

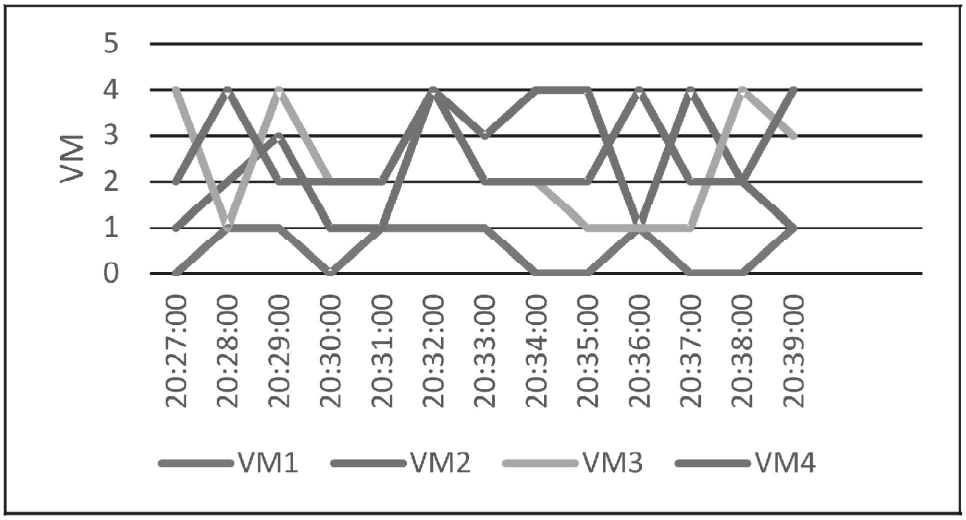

Figure 7: SLA Violations

3. CONCLUSION & FUTURE WORK

Fault aware cloud computing environments to support the elastic provisioning has proved to be very beneficial. Experiments conducted for validating the architecture clearly depict that autonomic computing and cloud computing can be used together with various technologies and different providers. The future work involves different criteria that should be used for rules design (e.g., average response time of requests or latency). Furthermore, the use of other levels of control loops may improve the architecture's effectiveness, focusing on better performance.

REFERENCES

[1] Indukuri R. Krishnam Raju, Penmasta Suresh Varma, M. V. Rama Sundari, G. Jose Moses, "Deadline Aware Two Stage Scheduling Algorithm in Cloud Computing", Indian Journal of Science and Technology, 2016 Jan, 9(4).

[2] S. Rajarajeswari, K. Somasundaram, "Data Confidentiality and Privacy in Cloud Computing", Indian Journal of Science and Technology, 2016 Jan, 9(4).

[3] M. Reji, P. C. Kishore Raja, Christeena Joseph, Radhika Baskar, "Performance Metrics of Wormhole Detection using Path Tracing Algorithm", Indian Journal of Science and Technology, 2015 Aug, 8(17), Doino:10.

[4] Q. Zhang Q, L. Cheng L, and R. Boutaba, R . "Cloud computing: state-of-the-art and research challenges" In Cloud Computing, 2010 The Brazilian Computer Society Conference on, pp. 7-18, Springer , 2010. 2010, pp. 7-18.

[5] G. Rohini, A. Srinivasan, "Multi Server Based Cloud-Assisted Real-time Transrating for Http Live Streaming", Indian Journal of Science and Technology, 2016 Jan, 9.