International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 05, May 2017
www.ijates.com
ijates
ISSN 2348 - 7550

CLOUD SERVER OPTIMIZATION WITH LOAD BALANCING MECHANISM USING DYNAMIC THRESHOLD

Baldev Singh[1], Rajiv Mahajan[2]
Research Scholar, IKG Punjab Technical University, (India)
Director, Golden College of Engg. & Technology, (India)

ABSTRACT

Cloud computing provides the way to share distributed resources and services to the cloud users on pay-as-usage basis. Multiple virtual machines are configured in cloud to respond the requests made by the users effectively and efficiently. Users accessing the services of cloud concurrently as cloud services are distributed in nature. To provide the cloud computing resources by the cloud service providers is a core and challenging issue as some of the virtual machines (VMs) may be stuck off or may not be able to respond well in time due to huge load on them. Some of the machines may be underutilized at the same time. So load balancing is utmost required in cloud computing so that the response time as well as throughput can be optimized and delay of any time can be avoided in addition to it, the cloud services can be provided as per Service Level Agreement to the cloud users. This paper focuses on the various metrics based threshold mechanism of load balancing for cloud server optimization.

Keywords: Cloud computing, Threshold, Load Balancing, VMs.

I INTRODUCTION

Technology of cloud computing provides a way of using computing and storage resources by using Internet and remote servers. Cloud computing[1,2] has also been exposed to many security threats which leads to denial of services. Load of cloud resources should be balanced in any type of threats in cloud computing. Cloud computing is defined and interpreted by various researchers. Vaquero et al [16] identified more than 20 definitions of cloud computing but the most cited definition proposed by US National Institute of Standards and Technology (NIST).

NIST defined cloud computing as: "Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources [3,16](e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." This definition reveals that Cloud provides three types of services namely, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) to the cloud community. There are mainly four types of cloud model which are Private Cloud, Public Cloud, Community Cloud and Hybrid Cloud. This classification is based on the ownership and composition of cloud.

More specifically, Cloud Computing is the combination of a technology, platform that provides hosting and storage service on the Internet. Main goal of the cloud computing is to provide scalable and inexpensive on-

demand computing infrastructures [4-6] with good quality of service levels. Many companies developing and offering cloud computing products and services but have not properly considered the implications of processing, storing and accessing data in a shared and virtualized environment. In fact, many developers of cloud-based applications struggle to include proper load balancing of cloud resources.

II CLOUD DEPLOYMENT MODELS

A cloud deployment model [7] specifies how resources within the cloud are shared. There are four primary cloud deployment models.

Private cloud: Owned by a specific entity and normally used only by that entity or one of its customers. The underlying technology may reside on-or off-site. A private cloud offers increased security at a greater cost.

Public cloud: This type pf cloud is available for use by the general public. It may be owned by a large organization or company offering cloud services. Because of its openness, the cloud may be less secure. A public cloud is usually the least expensive solution.

Community cloud: The cloud is shared by two or more organizations, typically with shared concerns.

III COMMON CLOUD OFFERINGS

The cloud model provides three types of services/ offerings. The figure 1 shows the common services [8-10] available to the cloud users:

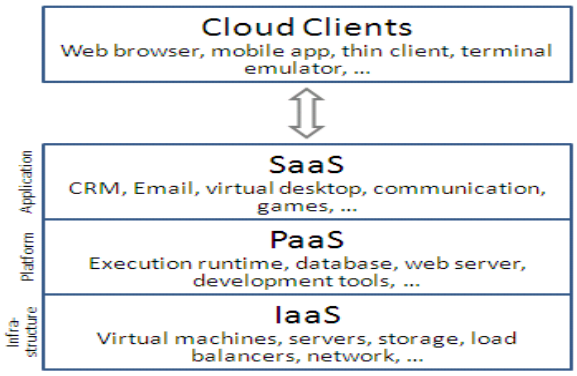

Figure 1. Cloud computing Services Model

Software as a Service (SaaS): The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email).

Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure his own applications without installing any platform or tools on their local machines. PaaS refers to providing platform layer resources, including operating system support and software development frameworks that can be used to build higher-level services.

Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.

IV ISSUES IN CLOUD COMPUTING

Various issues are there in Cloud computing related to Performance, Security, Availability, and Inability to customize. This paper focuses on the main issue related to performance aspect. Working in a Cloud environment does not do away with application performance issues [8-10]. Cloud computing resources management is itself very complex that leads to even more performance troubles than in non-Cloud environments. The ongoing monitoring process of all the applications is performed via the Cloud. Service Level Agreements are mandatory to met with otherwise negative impact leads to elimination of existence. In addition optimal performance and uptime are needed. Dynamic resources can be effectively managed using virtualization technology[11] on a Cloud computing platform. Load balancing of the entire system can be handled dynamically. Due to change in load, it may be required to remap the physical resources as well as VMS by using virtualization technology. Virtualization is core technology to implement the whole things in the cloud including load balancing[12-16] of the cloud servers. Cloud architecture contains four basic entities in general which are host, datacenters, virtual machine and application that permits to set-up a basic cloud computing environment. In cloud, a Virtual Machine (VM) is a treated as software implementation of a computing environment in which an operating system (OS) or program can be installed and run. A Virtual Machine characteristically emulates a physical computing environment. Various resources like CPU, memory, hard disk, network and other hardware resources are granted to VMs on the basis of request and are managed by a virtualization layer which translates these requests to the underlying physical hardware. By using virtualization layer numerous individual, isolated VM environments[17] can be created. System software and Application are executed on Virtual Machine on-demand. Deployments as well as development of custom application service models are made by using the VMs.

V LOAD BALANCING USING DYNAMIC THRESHOLD IN CLOUD

Load balancing is the mechanism through which the loads are to the individual nodes of the system so that the best response time and also good utilization of the resources and process can be ensured. The Load Balancing module/ systems are able to distribute the workload balancing it between multiple Cloud Servers. The main tasks in context of load balancing [18-20] are the resource provisioning or resource allocation and second one is task scheduling in distributed environment so that the cloud resources be easily available on demand.

The feasible environment or metrics are required for measuring the efficiency and effectiveness of Load Balancing algorithms/ methods. The load balancing may be centralized, distributed or hybrid depending upon the mechanism is used for the said purpose. In centralized load balancing technique [21,22] all the allocation and scheduling decision are made by a single node. The whole responsibility lying with the single node is for storing knowledge base of entire cloud network. The approach for load balancing can be static or dynamic. Although this method minimize the time required to analyze various cloud resources but it leads to a great overhead [24] on the centralized node. As failure intensity of the overloaded centralized node is high and

recovery might not be easy in case of node failure hence centralized mechanism is considered as no longer fault tolerant.

No single node is responsible for making resource provisioning [16,23,25] or task scheduling decision in case the load balancing technique is distributed. In distributed system , there is no single domain to act as monitoring the cloud network instead multiple domains monitor the network for load balancing purpose. Every node in the network is involved to create and maintain local knowledge base for efficient distribution of tasks in static environment and re-distribution in dynamic environment. The distributed load balancing system [30-32] has edge over centralized system as this system is more faults tolerant and balanced there is no single node is overloaded for load balancing. Hierarchical load balancing can be an alternative of the above approaches. It involves different levels of the cloud in load balancing decision. Hierarchical techniques mostly operate in master slave mode [24] . Tree data structure can be implemented where each and every node in the tree is balanced under the control of its parent node.

The main objectives and key features of load balancing are: (a) Even distribution of load to each resource (b) Processing time minimization for each task (c) Maximum resource utilization (d) High adaptability (e) Minimize the task migrations (f) Distributed resource discovery optimization. The load balancing management is based on the following steps:

- Obtain the load status of all servers
- Assess the status of servers
- Estimate the future load flow
- Choose receiver nodes and Migration (whenever required)

5.1 Measuring the Load Balancing in Cloud

Load on cloud server may be un-even. To manage the un-evenness of load of cloud servers refers to balancing of load so that there will not be overloading or under utilization of the cloud severs at the maximum.

An ‘Un-evenness’ may be found while observing multiple factors represented as composite or derived parameters. It is either the values of parameters that start touching abnormal ‘lows’ or ‘highs’ at certain ‘intervals’ of data series or values of parameters start scaling higher values from the normal scale. Too much variation or too many values away from normal (balnced load) reflect the underlying erraticness of the data stream, which my come due to un-balanced load.

*Composite Metric [16,18]***:** Since cloud consists of many end points which are geographically apart but working together to provide a particular service. We need a measurement model that can work to observe these end points and reflect the load of cloud and its resources. Hence we have left only with one option is to be build a model which is composite or additive in nature. We can also think composite metric as a derived metric which consists of indicator variable(s) that get directly impacted by load of cloud servers. The metric works on the strength of predictions of associations of various manifest variables as explained below:

5.2 Assumptions

1) All the first level component variables imply knowledge of composite variables but not vice versa.
2) All effects occur through this component and are measurable.
3) Total effect can be calculated from the effects on components.

Let Cmt , β be the two variables representing composite indicator variables where β >0, there are unique indicators α, γ such that

$$Cmt = \alpha . \beta + \gamma \text{ where}$$

$0 \leq \gamma < \beta$, therefore

$Cmt = \alpha . \beta + \gamma$ represents a composite metric.

5.3 Design Principles of Composite Metric

1) The Composite metric structure must be able to incorporate following aspects for measuring the load of cloud servers:

- System Characteristics of the cloud
- Application Characteristics
- Technology Characteristics
- Network Characteristics

2) The composite metric measurements must consist of variables that have correlation and shared variance or degree to the extent that they move together

3) The composite metric manifest variables must be based on Proportional relationships with each other.

4) The composite metric must help in simplification of monitoring process

5) The composite metric must help in overcoming the "curse of dimensionality.

Formation of Composite metric: Following are the sub metrics used in formation of composite metric which is used for load balancing strategy development:

1) *KbMemUsed:* It refers to the amount of memory used in Kilbytes. It does not include the memory used by the kernel of operating system. If the memory usage approaches to 100%, it leads to slow down of system.

2) *KbMemFree:* It is an amount of free memory in KBs of a system, if the there is overload, KbMemFree value decreases. This means the value will be closed to zero. It indicates that there is no free memory for application to run on VM and system will slow down.

3) *KbBuffers:* It refers to the amount of memory used as buffers *by the kernel. The size is measured* in kilobytes. If the buffer usage approaches to 100%, it leads to slow down of system.

4) *KbCached* : It refers to the amount of memory used to cache data *by the kernel. The size is measured* in kilobytes. If the cache usage approaches to 100%, it leads to slow down of system.

5) *Memory Utilization*: To calculate the actual memory utilization, subtract kbbuffers and kbcached from kbmemused, hence we find the memory utilization in percentage. If the memory utilization approaches to 100%, it leads to slow down of system.

6) *Absolute Free Memory*: Absolute free memory is defind on the basis of the formula: KbMemUsed-KbBuffers-KbCached.

7) *Total Memory*: Total memory is sum of KbMemFree and KbMemUsed.

8) *Total Used Memory*: Total used memory in percentage of AFM/TM.

9) *Free Memory:* Free memory refers to the difference between 100 and TotalUsedMemory.

10) *LoadAvg* : This is a metric that measures the load on the VM, typically the load averages are taken at different times intervals (1 second , 5 second , 15 seconds) . In our context, we have calculated the inter-quartile of load average values . High load averages indicates performance degradation. If higher load is sustained over large number of time intervals, it is an indication of either overload or some adversity in to the play , and further calls for investigations for possible DDOS attack .

11) *%Utilization of devices(μ):* It is calculated as Blkio.ticks/ deltams*100%. Here blkio.ticks is "# of milliseconds spent doing I/Os". deltams is the time elapsed since last snapshot in ms. Device saturations will be there if the value approaches to 100%. It leads to DDOS attack alert. Higher utilization is an indicator of more load on the machine.

12) *α :* This is basically a derived metric, that measures, how much of free memory is left when load and utilization measurement are realized. This derived metric is composed of product of ratios of load to free memory. The derivation of these formulae is multiplicative as they impact each other adversely rather than just additively.

13) *Frequency of Block*–Read: The number of disk read commands completed on each VM on the Host , the block size may vary the overall rate .

14) *Block–Write* : The number of disk write commands completed on each VM on the Host , the block size may vary the overall rate.

15) *β:* This metric measures how much does new jobs need to wait when the process of read/write blocks is occurring.. Being a ratio, it truly measures with respect to the technology used in disk. The (await) metric reflects how much of the disk operations are impacting tardiness time for the jobs in VM in queue. If the basic-r/w operations are intensive and proportionate to the waiting time, would measures more the r/w operations and more is the waiting time.

16) *γ:* This metric basically reflects the network characteristic of the VM. It covers three types of packet(s). It measures the average of those packets volumes (no. of packets received to the number of packets sent) and added with the traffic related to http.

Following are some commands and related statistics [26-29] examples that are related to the composite metric and are helpful to measure the load of servers. It further helps in load balancing strategy. The following screenshot displays the Global Average Activities by All CPUs. [26-29]

```
tecmint@tecmint ~ $ mpstat
Linux 3.11.0-23-generic (tecmint.com)   Thursday 04 September 2014      _i686_  (2 CPU)
12:23:57  IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice
12:23:57  IST  all   37.35    0.01    4.72    2.96    0.00    0.07    0.00    0.00    0.00
```

The following screenshot displays statistics about all CPUs one by one starting from 0. 0 will the first one.

```
tecmint@tecmint ~ $ mpstat -P ALL
Linux 3.11.0-23-generic (tecmint.com)   Thursday 04 September 2014      _i686_  (2 CPU)
12:29:26  IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice
12:29:26  IST  all   37.33    0.01    4.57    2.58    0.00    0.07    0.00    0.00    0.00
12:29:26  IST    0   37.90    0.01    4.96    2.62    0.00    0.03    0.00    0.00    0.00
12:29:26  IST    1   36.75    0.01    4.19    2.54    0.00    0.11    0.00    0.00    0.00
```

The following screenshot displays the statistics [26-29] for N number of iterations after n seconds interval with average of each CPU. The following values are depicted:

- **CPU:** processor number. The keyword *all* indicates that statistics are calculated as averages among all processors.
- **%usr:** show the percentage of CPU utilisation that occurred while executing at the user level (application).
- **%nice:** show the percentage of CPU utilisation that occurred while executing at the user level with nice priority.
- **%sys:** show the percentage of CPU utilization that occurred while executing at the system level (kernel). Note that this does not include time spent servicing hardware and software interrupts.
- **%iowait:** show the percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request.
- **%irq:** show the percentage of time spent by the CPU or CPUs to service hardware interrupts.
- **%soft:** show the percentage of time spent by the CPU or CPUs to service software interrupts.
- **%steal:** show the percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor.
- **%guest:** show the percentage of time spent by the CPU or CPUs to run a virtual processor.
- **%idle:** show the percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.

```
tecmint@tecmint ~ $ mpstat -P ALL 2 5
Linux 3.11.0-23-generic (tecmint.com)   Thursday 04 September 2014      _i686_  (2 CPU)
12:36:21  IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice
12:36:23  IST  all   53.38    0.00    2.26    0.00    0.00    0.00    0.00    0.00    0.00
12:36:23  IST    0   46.23    0.00    1.51    0.00    0.00    0.00    0.00    0.00    0.00
12:36:23  IST    1   60.80    0.00    3.02    0.00    0.00    0.00    0.00    0.00    0.00
12:36:23  IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice
12:36:25  IST  all   34.18    0.00    2.30    0.00    0.00    0.00    0.00    0.00    0.00
12:36:25  IST    0   31.63    0.00    1.53    0.00    0.00    0.00    0.00    0.00    0.00
12:36:25  IST    1   36.73    0.00    2.55    0.00    0.00    0.00    0.00    0.00    0.00
12:36:25  IST  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice
12:36:27  IST  all   33.42    0.00    5.06    0.25    0.00    0.25    0.00    0.00    0.00
12:36:27  IST    0   34.34    0.00    4.04    0.00    0.00    0.00    0.00    0.00    0.00
12:36:27  IST    1   32.82    0.00    6.15    0.51    0.00    0.00    0.00    0.00    0.00
```

The following screenshot displays network statistics using '-n DEV'.

```
tecmint@tecmint ~ $ sar -n DEV 1 3 | egrep -v lo
Linux 3.11.0-23-generic (tecmint.com)   Thursday 04 September 2014      _i686_  (2 CPU)
02:11:59  IST     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s   rxcmp/s   txcmp/s  rxmcst
02:12:00  IST     wlan0      8.00     10.00      1.23      0.92      0.00      0.00      0.
02:12:00  IST    vmnet8      0.00      0.00      0.00      0.00      0.00      0.00      0.
02:12:00  IST      eth0      0.00      0.00      0.00      0.00      0.00      0.00      0.
02:12:00  IST    vmnet1      0.00      0.00      0.00      0.00      0.00      0.00      0.
```

The following screenshot displays block device statistics like iostat using '-d'.

```
tecmint@tecmint ~ $ sar -d 1 3
Linux 3.11.0-23-generic (tecmint.com)   Thursday 04 September 2014      _i686_  (2 CPU)
02:13:17  IST       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svc
02:13:18  IST    dev8-0      0.00      0.00      0.00      0.00      0.00      0.00      0.
02:13:18  IST       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svc
02:13:19  IST    dev8-0      0.00      0.00      0.00      0.00      0.00      0.00      0.
02:13:19  IST       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svc
02:13:20  IST    dev8-0      7.00     32.00     80.00     16.00      0.11     15.43     15.
```

The following screenshot shows the output of the **top** program [26-29] that provides a dynamic real-time view of a running system. It's very useful for determining processes which use the most CPU (and not just that) at the time of monitoring. In shcreenshot below, the Line number 3, marked in blue, shows CPU state percentages based on the interval since the last refresh. Values shown are as follows :

- **us:** time running un-niced user processes.
- **sy:** time running kernel processes.
- **ni:** time running niced user processes.
- **id:** time spent idle.
- **wa**: time waiting for I/O completion.
- **hi:** time spent servicing hardware interrupts.
- **si:** time spent servicing software interrupts.
- **st:** time stolen from this vm by the hypervisor.

```
$ top
top - 20:53:16 up 3 days, 21:08,  2 users,  load average: 0.86, 0.66, 0.32
Tasks: 127 total,   2 running, 124 sleeping,   0 stopped,   1 zombie
%Cpu(s):  7.0 us, 24.7 sy, 0.0 ni, 48.9 id, 17.7 wa, 0.0 hi, 1.6 si, 0.0 st
KiB Mem:   1022744 total,  1008568 used,    14176 free,      584 buffers
KiB Swap:   991228 total,   105604 used,   885624 free,   662508 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
29390 sandy     20   0  3536  560  472 R  66.0  0.1   0:10.54 dd
   23 root      20   0     0    0    0 S   5.3  0.0   1:22.64 kswapd0
 2860 mysql     20   0  321m 164m 2992 S   0.7 16.5  68:55.36 mysqld
29370 sandy     20   0  4512 1348  952 R   0.7  0.1   0:02.09 top
 1923 root      20   0  279m  96m  984 S   0.3  9.7  53:45.61 nessusd
 1945 root      20   0  2052  424  352 S   0.3  0.0   1:19.69 vnstatd
 2642 zabbix    20   0  3036  572  508 S   0.3  0.1   7:45.23 zabbix_agentd
 3103 zabbix    20   0 59872 9512 9216 S   0.3  0.9   4:07.73 zabbix_server
 3147 www-data  20   0 24132 1080  896 S   0.3  0.1   6:15.05 zmdc.pl
29346 sandy     20   0  9452 1492  816 S   0.3  0.1   0:00.35 sshd
```

Various command [26-29] like sar utilities, vmstat, iostat etc., one can find the load of cloud servers on the basis of various hardware resources. Using the related statistics of various metric, a composite metric values is to be computed dynamically, on the basis of which load statistics is calculated and examined. This composite metric value is then compared with the threshold value which is calculated dynamically for different cloud severs and then optimal load balancing technique is suggested for the load balancing purpose.

VI CONCLUSION AND FUTURE WORK

Load balancing is one of the major altercation in cloud computing. It is required to allocate the workload evenly among all the cloud servers so that high resource utilization ratio and user satisfaction can be ensured. Various load balancing techniques/algorithms are proposed by researchers. To identify the best strategy of load balancing in cloud environment, an dynamic threshold based load balancing mechanism is proposed in which composite value is computed and compared with the dynamic threshold of load of cloud severs and then appropriate load balancing is suggested to implement. The appropriate strategy of load balancing for improving the system and network performance, scalability and optimal resource utilization is the concern for future enhancement of the proposed work.

REFERENCES

[1] Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao and Li Ruan, "A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers," Journal of Parallel Distrib. Comput. 72, pp. 1254–1268, 2012.

[2] Ioannis Konstantinou, Dimitrios Tsoumakos and Nectarios Koziris, "Fast and Cost-Effective Online Load-Balancing in Distributed Range-Queriable Systems," IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 8, pp. 1350-1364, 2011.

[3] Sheng Di , Cho-Li Wang, "Decentralized proactive resource allocation for maximizing throughput of P2P Grid," In the Journal Proceedings of Parallel Distrib. Comput. 72, pp. 308–321, 2012.

[4] Emilio Ancillotti, Raffaele Bruno, Marco Conti, Antonio Pinizzotto, "Load-aware routing in mesh networks: Models, algorithms and experimentation," In the Journal Proceedings of Computer Communications 34, pp. 948–961, 2011.

[5] P. Munoz, R. Barco and I. de la Bandera, "Optimization of load balancing using fuzzy Q-Learning for next generation wireless networks," In the Journal Proceedings of Expert Systems with Applications, Vol. 8, Issue. 11, pp. 1469 – 1479, 2012.

[6] A.Saffar, R. Hooshmand, A. Khodabakhshian, "A new fuzzy optimal reconfiguration of distribution systems for loss reduction and load balancing using ant colony search-based algorithm," Journal of Applied Soft Computing, Vol. 11, Issue. 5, pp. 4021–4028, 2011.

[7] Che-Lung, Hung, Hsiao-hsi Wang and Yu-Chen Hu, "Efficient Load Balancing Algorithm for Cloud Computing Network," 2010

[8] Zhenyu, Li, Gaogang Xie, Kai Hwang and Zhongcheng Li, "Churn-Resilient Protocol for Massive Data Dissemination in P2P Networks," IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 8, pp. 1342- 1349, 2011.

[9] Daniel Warenke and Odej Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud," IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, pp. 985-997, 2011.

[10] David Carrera, Malgorzata Steinder, Ian Whalley, Jordi Torres and Eduard Ayugade, "Autonomic Placement of Mixed Batch and Transactional Workloads," IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 2, pp. 219- 231,2012.

[11] Hung-Chang Hsiao, Hao Liao, Ssu-Ta Chen and Kuo-Chan Huang, "Load Balance with Imperfect Information in Structured Peer-toPeer Systems," IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 4, 2pp. 634- 649, 2011.

[12] Jasma Balasangameshwara and NedunchezhianRaju, "A hybrid policy for fault tolerant load balancing in grid computing environments," Journal of Network and Computer Applications, Vol. 35, pp. 412–422, 2012.

[13] Yuehua Wang, ZhongZhou, LingLiu and WeiWu, "Replica-aided load balancing in overlay networks," Journal of Network and Computer Applications, Vol. 36, Issue. 1, pp. 388–401, 2013.

[14] Yunhua Deng and Rynson W.H. Lau, "On Delay Adjustment for Dynamic Load Balancing in Distributed Virtual Environments," IEEE Transactions on Visualization and Computer Graphics, Vol. 18, No. 4, pp. 529- 537, 2012.

[15] Jinhua Hu, Jianhua Gu, Guofei Sun and Tianhai Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment," 3rd International Symposium on Parallel Architectures, Algorithms and Programming, pp. 89-96, 2010.

[16] Baldev Singh, S.N. Panda, Samra G.S., Mahajan Rajiv, "Detecting DDOS Attacks in Cloud- A Novel Approach" in the International Journal of Computer Science and Information Security, Vol. 15, No. 5, 2016

[17] 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, pp. 153-160, 2010.

[18] Baldev Singh, S.N. Panda, "An Adaptive Approach to Mitigate DDOS Attacks in Cloud" In the *International Journal of Advanced Computer Science and Applications, Vol. 6, No. 10, 2015.*

[19] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao and Shun-Sheng Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network," 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Vol. 1, pp. 108-113, 2010.

[20] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong and Dan Wang, "Cloud Task scheduling based on Load Balancing Ant Colony Optimization," In the Proceedings of 6th Annual Chinagrid Conference (ChinaGrid), pp 3-4, 2011.

[21] Rui Wang, Wei Le and Xuejie Zhang, "Design and Implementation of an Efficient Load-Balancing Method for Virtual Machine Cluster Based on Cloud Service," 4th IET International Conference on Wireless, Mobile & Multimedia Networks (ICWMMN 2011), pp. 321-324, 2011.

[22] Lorpunmanee, S., Sap, M.N, Abdul Hanan Abdullah, A.H., "An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment" in Proceedings of World Academy of Science, English and Technology Volume 23 august 2007, ISSN 1307-6884, 2007.

[23] Buyya, R., Ranjan, R., Calheiros, R.N., "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities" in Proceedings of the 7th High Performance Computing and Simulation (HPCS 2009) Conference, Leipzig, Germany, 2009 .

[24] Jing Yao and Ju-hou He, "Load Balancing Strategy of Cloud Computing based on Artificial Bee Algorithm," 8th International Conference on Computing Technology and Information Management (ICCM), Vol. 1, 2012, pp. 185-189.

[25] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid, "Availability and Load Balancing in Cloud Computing," International Conference on Computer and Software Modeling, Vol. 14, 2011, pp. 134-140.

[26] Accessed from "http://www.thegeekstuff.com/2011/03/sar-examples/?utm_source=feedburner " Accessed on 10-04-2017.

[27] Accessed from 'https://www.lisenet.com/2014/measure-and-troubleshoot-linux-cpu-resource-usage/ " Accessed on 12-04-2017.

[28] Accessed from "https://www.lisenet.com/2014/measure-and-troubleshoot-linux-disk-io-resource-usage/: Accessed on 20-04-2017.

[29] Accessed from http://linoxide.com/linux-command/linux-iostat-command/ Accessed on 20-04-2017.

[30] Ayman G. Fayoumi, "Performance Evaluation Of A Cloud Based Load Balancer Severing Pareto Traffic," Journal of Theoretical and Applied Information Technology, Vol. 32, No.1, 2011.

[31] P. Mohamed shameem and R.S. Saji, "A Methodological Survey On Load Balancing Techniques In Cloud Computing" In The International Journal Of Engineering And Technology. Vol. 5, No.5. 2013.

[32] Pragati Priyadarshinee and Pragya Jain,"Load Balancing and Parallelism in Cloud Computing," International Journal of Engineering and Advanced Technology, Vol. 1, Issue. 5, 2012, pp. 486-489.