Response Time Reduction and Performance Analysis of Load Balancing Algorithms at Peak Hours in Cloud Computing

Monika Kushwaha
Pranveer Singh Institue of Technolgy
Kanpur, U.P. (208020)
Dr. A.P.J. Abdul Kalam Technical University,
Lucknow, U.P.

Saurabh Gupta
Pranveer Singh Institue of Technolgy
Kanpur, U.P. (208020)
Dr. A.P.J. Abdul Kalam Technical University,
Lucknow U.P.

ABSTRACT
We are living in Digital Age which is undoubtedly the outcome of highly developed internet and its corresponding technologies. Cloud computing has become prominent as more people, organizations, entrepreneur, are moving towards it as it facilitates them to achieve their dreams in less investment. Now people from all over the globe are demanding for various services in rapid rate which has lead to bursty workloads on data centers thereby creating peak hour optimization problem to be handled. It has also lead to problem of load balancing which should be addressed to increase the efficiency of data centers. In the present work, peak hour optimization is being aimed and algorithms are being analyzed for large scale application to find efficient algorithm for real time scenario. Further, a novel strategy is being proposed to decrease the response time and DC processing time of algorithms without increasing the overall cost.

General Terms
Cloud computing, Load balancing in peak hours

Keywords
Load Balancing, Cloud computing, Round Robin, Throttled, Equally Spread Current Execution

1. INTRODUCTION
With extensive growth and use of internet technologies, IT industry gave rise to 'Cloud Computing' which is an emerging computing paradigm and has become buzzword in academia and industry. Definition given by National Institute of Standards and Technology (NIST) of cloud computing is most widely accepted and it states, "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 [1]."

The term "cloud computing" is coined by University of Texas professor Ramnath chellapa in a talk on "new computing paradigm" in 1997. Cloud computing has evolved through various phases which include Grid Computing, Utility Computing, Application Service Provider (ASP) and Software as a Service (SaaS) [2].

Cloud computing provides real time scalable resources and services on demand on pay-per-use basis. It mainly provides three types of services: Infrastructure as a service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The services are explained in brief in Table 1 below.

Table 1. Cloud computing services summary

Category	Features	Vendors & Products
IaaS	Vendors provide virtualized hardware (i.e. servers, network resources, etc) and storage space as service.	**Amazon EC2** provides Virtual server slices known as Instances, **Amazon S3** give data storage in cloud
PaaS	Vendors provide platform on cloud to develop applications.	**Heroku** gives cloud platform for Ruby, **Google App Engine** targets python and java developers
SaaS	Vendors provide applications accessible through web.	Google Apps which provide Google calendar, Google docs and Gmail.

Cloud computing architecture consists of various components and models which uses some of the actors to provide their service. One of the common model used consist of service provider, service user and service broker. Service broker also plays an important role in improving the efficiency of data center and algorithms as it manages the traffic between incoming request and data centers available. Hence load balancing can be divided into two parts i.e. load balancing through service broker by choosing data centers efficiently and load balancing the VMs in each data centers.

Cloud computing is getting popular among business holders and organization as it helps them to start business with less initial investment required for hardware, software, etc and expand resources whenever required.

1.1 Virtualization and Load Balancing in Cloud Computing
Virtualization is foundation technology in cloud computing. Virtualization technology abstracts the physical resources in cloud computing data centers and provides virtualized resources to the customers on demand. Hypervisors like Linux Kernel-based Virtual Machine [4] and Xen [3] are used to abstract the physical server and creates multiple virtual servers known as virtual machines. Providers can customize each virtual machine according to need of customers. It helps in better use of resources and also decreases the electricity cost of data centers.

Clod computing faces many challenges among which is Load balancing. Load balancing is defined as a process of

reassigning the total load to the individual nodes such that no node is under loaded or over loaded and hence making effective resource utilization thereby minimizing the response time of the job. In real time scenario the peak hours mostly degrade the overall performance of load balancing algorithm. With increased demand of internet users the busty workload has become frequent so it is important to analyze and design algorithms to handle them as well.

2. BACKGROUND AND RELATED WORK

Load balancing means removing tasks from overloaded VMs and assigning them to under loaded VMs. In this section, we will discuss most known load balancing algorithm being proposed for cloud computing environment. Shridhar G.Domanal and G.Ram Mohana Reddy [5] have proposed Modified Throttled Algorithm which is improvement over throttled algorithm and it distributes workload evenly among virtual machines. Modified throttled algorithm initially selects VM at first index and checks the state of VM. If VM is available then it is assigned with the request and VM id is returned to data center, else -1 is returned. On arrival of next request, the VM at index next to assigned VM is chosen for checking the state. Proposed algorithm is compared with existing round robin and throttled algorithm and showed considerable improvement in response time. Hamid Shoja et. al [6] have provided a comparative survey and analysis of two dynsmic load balancing algorithm i.e. round robin and throttled. Their analysis shows that overall response time and data center request servicing time remain same for both the algorithms but estimated cost of usage is different. Throttled algorithm reduces the cost over round robin hence it is efficient in terms of cost. Klaithem Al Nuaimi et. al [7] have surveyed various static and dynamic algorithm for load balancing in cloud computing (Central Load Balancing Decision Model, Index Name Server, Exponential Smooth Forecast based on Weighted Least Connection, Dual-Direction FTP, Load Balancing Min-Min,Ant Colony, Enhanced MapReduce) and told the challenges that must be taken care of to provide the most efficient load balancing algorithms. Ashwin Kumar et. al [8] have proposed a novel VM load balancing algorithm which has improved over Active VM load balancing algorithm. They have considered peak hours case and experiment showed that their proposed algorithm evenly distribute the request among all VMs unlike active VM load balancer. They have done this improvement by using a reservation table between the phase of selection and allocation of VMs.

As we know service broker policy is also part of load balancing so different service broker policies have been made. Rakesh Kumar Mishra et. al [11] proposed priority based round robin service broker algorithm, it first calculates priority of each data center according to their performance and then assign request to each data center in round robin manner. It gives better performance from some existing broker algorithm but it increases the cost.

Two other service broker algorithms proposed by researchers which we will consider in this paper for further analysis are as follows.

2.1 Service Proximity based algorithm

It is also known as closest data center algorithm. Here service broker selects the data center which is closest to the request sender's location taking into consideration transmission latency. [9][11]

2.2 Performance Optimized algorithm

It is also known as Optimize response time algorithm. Here service broker selects the data center according to best response time.[9] [11]

Three other load balancing algorithms proposed which are widely used for load balancing and are selected for further analysis in this paper are as follows.

2.3 Round Robin Algorithm (RR)

It is simplest and traditional algorithm used for load balancing. It divides the time into multiple time slices or quantum and each node is assigned to a particular time slice for execution. [6] In cloud data centers round robin works by initially selecting a VM randomly and then assigning request to VMs in circular order. Each assigned VM is moved to end of the list after allocation of request. [5] The drawback it posses is that it does not consider state of VM that whether it is heavily loaded or lightly loaded.

2.4 Throttled Algorithm

It is mainly developed for cloud scenario to load balance the VMs. Throttled load balancer maintains an index table of all VMs and their respective state (i.e. available or busy). Whenever new request arrives the table is parsed by load balancer and VM having available state is chosen and its VM id is returned to data center controller which further assigns the request to that particular VM, if suitable VM is not found then -1 is returned to data center controller. The notification of new allocation after allocating request to VM and VM de-allocation after completing of request is sent by data center controller to load balancer. [9] [10]

2.5 Equally Spread Current Execution Load Algorithm (ESCE)

This load balancing algorithm is also known as Active Monitoring Load Balancing algorithm. It works quiet similar to throttled algorithm but with change in the VM index table. In this algorithm load balancer maintains an index table of all VMs along with the number of currently allocated requests to VM. Whenever new request arrive load balancer parses the table and VM having least load is chosen and its ID is returned to data center controller which further assigns the request to that VM and notifies the load balancer of this new allocation to increase the allocation count of that VM. After request gets completed load balancer is further notified about de-allocation of VM so that it decreases the allocation count of that VM. [9] [10].

3. PROBLEM STATEMENT

In today's era internet has become daily cup of tea for people and it has lead to bursty workloads and peak hours for cloud. Most of the algorithm developed for load balancing in cloud environment does not consider the scenario of peak hour and busty workload hence their performance degrades in such conditions. Peak hours or burstiness occurs when a lot of request gets accumulates at a particular interval of time, it is mostly seen in large scale web applications, large storage systems, etc. Here peak hour optimization problem has been considered. Earlier research have been done to analyze existing algorithms like Round Robin, Throttled and Equally spread current execution algorithm, and existing service broker policy like service proximity based algorithm and performance optimized algorithm, but they are analyzed for small scale data which do not resembles to real time scenario hence the results may vary when applied in real time large scale applications, and the VM balancing algorithm are never

analyzed with combination to service broker algorithm in such a large scale data. The second problem considered here is response time and data center service processing time reduction without increasing the overall cost. To give customer satisfaction it is important for them to reduce the servicing time of request without increasing the overall cost as it will reduce the expenses. So there is need to figure out logic which could be employed to do the same.

4. PROPOSED WORK

To resolve the above stated problem simulation of the large scale application like Facebook, twitter, etc are done creating a hypothetical application with peak hours and analysis of three widely used algorithms i.e. Round Robin, Throttled and Equally spread current execution algorithm is done to check their performance when such a huge data is inserted and peak hours are included. Combined performance of algorithms with two service broker policies that are closest data center policy and optimal response time policy are also being analyzed. After getting the most efficient pair of service broker policy and VM balancing algorithm we will it is further tried to reduce the response time and data center request processing time without increasing the cost by changing the VMs ratio in each data center on the basis of peak hour load timing and data center processing time.

5. EXPERIMENTAL SETUP AND SIMULATION

Experiment and analysis has been done through simulator known as CloudAnalyst. CloudAnalyst [9] is a simulation toolkit and is built on top of CloudSim toolkit using Java, Java Swing and SimJava. It provides a GUI interface to simulate the parameters as required. Below figure 1 shows the architecture of cloud analyst.

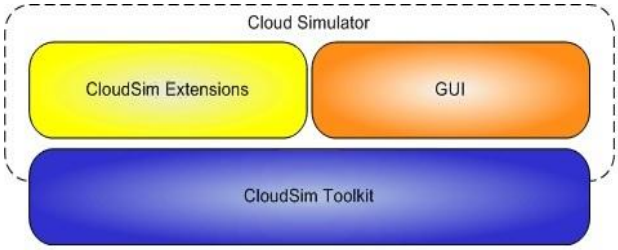

Fig 1: CloudAnalyst architecture.

5.1 Simulation

A social networking site like Facebook, twitter, etc also uses cloud computing and it has a lot of variation in user request and peak hours hence making it suitable for our simulation setup. Our most popular social networking site is Facebook. On 31/12/2012 approximate distribution of the Facebook user base across the globe was the [12] following:

Table 2. facebook users in each continent

Continents	Users in millions
North America	182
South America	142
Europe	250
Asia	254
Africa	51
Oceania	14

This data shows that cloud computing needs to handle a vast no. of users. For even one hour if half of the users got online at a same time at stated uploading things than it creates vast workload on application which we called bursty workload and time is called peak hours.

For evaluating the algorithms and broker policies and creating scenario similar to real time environment a hypothetical application has been simulated which is at $1/20^{th}$ of the scale of Facebook.

Users are considered at six different continents, hence six user base are made. A single time zone has been considered for each user base and it is assumed that people all over the globe uses the application in the evening after returning from their work for two hours. It is also assumed that 5% of total registered user of our hypothetical application remains active in the peak hour time and one tenth of the peak hour user remains online in off-peak hours.

Table 3. User base configuration

User base	Region	Simultaneous Online Users During Peak Hrs	Simultaneous Online Users During Off-peak Hrs
UB1	0-North America	455000	45500
UB2	1-South America	355000	35500
UB3	2-Europe	625000	62500
UB4	3-Asia	635000	63500
UB5	4-Africa	127500	12750
UB6	5-Ocenia	35000	3500

Other parameters being configured are that here four data center (DC) are used DC1- region 0, DC2- region 1, DC3- region 2 and DC4- region 3. Each data center has initially given 60 VMs. Each virtual machine size is 100MB, with RAM memory of 1 GB and 10 MB of available bandwidth. Data Center architecture used is x86, OS – Linux, VMM – Xen with twenty physical hardware units in each DC. User grouping factor in user bases used is 1000, request grouping factor in data centers used is 100 and executable instruction length per request used is 250 bytes. Load balancing algorithms being used for analysis are – Round Robin, Throttled algorithm and Equally spread current execution load algorithm. Service broker algorithms being used are Service proximity based algorithm and performance optimized algorithm.

In experiment first of all two cases are considered which are as follows.

CASE 1- Comparison of three load balancing algorithms using service proximity based service broker algorithm.

CASE 2- Comparison of three load balancing algorithms using performance optimized service broker algorithm.

Executing of above cases will give best pair of VM load balancing algorithm and service broker algorithm/policy. Further Performance Optimized service broker algorithm is taken as service broker policy as it has come out to be best in result and make further cases by changing the ratio of number of VMs used in each DC without increasing the total No. of VM used in whole application.

CASE 3- Changing the VM ratio in each data center as follows: DC1- 85, DC2- 30, DC3- 65, and DC4- 60 and comparing three algorithms under Performance Optimized service broker algorithm

After execution of case 3 the best VM balancing algorithm is found which comes out to be throttled algorithm hence further throttled algorithm and Performance Optimized service broker algorithm are used and VM ratios are changed to reduce their response time as follows.

CASE 4- Changing the VM ratio in each data center as follows: DC1- 85, DC2- 35, DC3- 70, and DC4- 50 using throttled algorithm and Performance Optimized service broker algorithm.

CASE 5- Changing the VM ratio in each data center as follows: DC1- 80 DC2- 40, DC3- 70, and DC4- 50 throttled algorithm and Performance Optimized service broker algorithm.

6. RESULT AND ANALYSIS

The experiment is carried out using above configuration for each cases and following result is obtained.

6.1 CASE 1 and CASE 2

6.1.1 Result

After executing CASE 1 and CASE 2 following results are obtained and result values are organized as shown below.

Table 4. Cumulative result of CASE1 and CASE2

Load Balancing Algorithm	Parameters	Service broker policy	
		Service proximity based algorithm	Performance optimized algorithm
Round Robin	Overall response time (ms)	832.02	672.29
	Data Center processing time (ms)	752.12	562.19
	Overall Cost ($)	5904.16	5904.16
Equally Spread Current Execution Load (ESCE)	Overall response time (ms)	832.22	668.62
	Data Center processing time	752.33	558.05
	Overall Cost ($)	5904.16	5904.16
Throttled algorithm	Overall response time (ms)	458.88	394.71
	Data Center processing time	382.41	288.36
	Overall Cost ($)	5904.16	5904.16

6.1.2 Performance Analysis

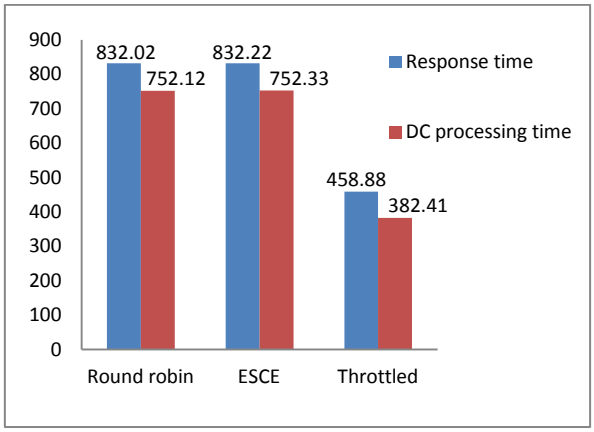

Fig 2: comparison of three algorithms in Service Proximity Based service broker policy (CASE 1)

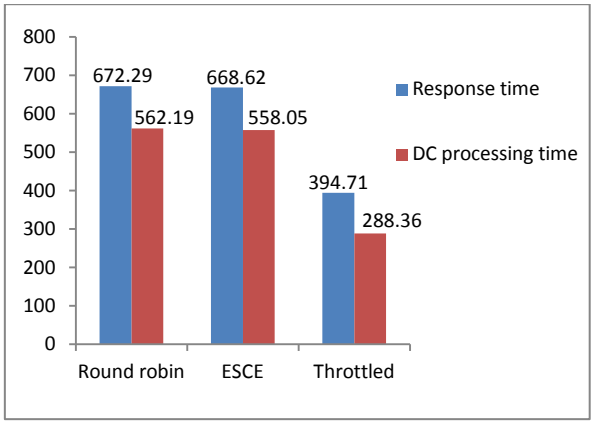

Fig 3: comparison of three algorithms in Performance Optimized service broker policy (CASE 2)

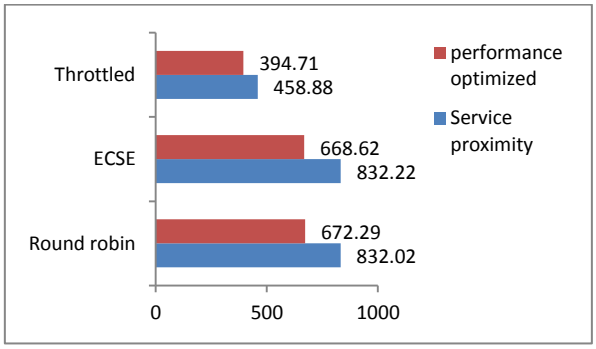

Fig 4: comparison of Response time of two service broker algorithms for all three VM balancing algo.

On analyzing the above graphs it is found that Throttled algorithm is more efficient in terms of response time and DC processing time in both the cases. Fig 4 shows that Performance Optimized service broker algorithm gives better result than service proximity based algorithm.

CASE 3

After observing above result it is found that Performance Optimized service broker algorithm gives better result so CASE 3 was formed using it only to focus only on further response time reduction. Here VM ratio is changed in each DC as follows: DC1- 85, DC2- 30, DC3- 65, and DC4- 60. This change in ratio is being done after observing the DC

processing time of each DC in CASE 2 as shown in table 5 below.

Table 5: DC processing time for throttled algorithm in CASE 2

Data Center	Avg (ms)
DC1	395.81
DC2	178.81
DC3	475.15
DC4	45.51

As it is seen in the above table 5 that DC2 and DC4 are having very less value as compared to other two which shows that these two DCs are having more VMs than required so we changed the ratio of VMs accordingly to see the change in response time. It is also taken care that total no. of VMs remain same because we don't want to increase the cost.

6.1.3 Result

After executing CASE 3 following result is obtained as shown in table 7 below.

Table 6. Result of CASE 3

Load Balancing Algorithm	Overall response time (ms)	Data Center processing time (ms)	Overall Cost ($)
Round Robin	629.60	518.47	5892.15
ESCE	629.98	518.33	5892.15
Throttled	372.35	263.15	5892.15

6.1.4 Performance Analysis

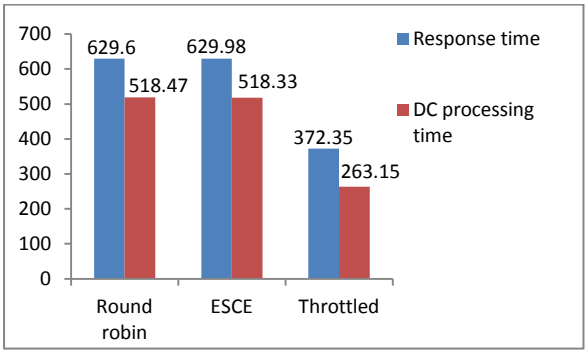

Fig 5: comparison of three algorithms in Performance Optimized service broker policy with new (CASE 3)

Here it is seen that by changing the number of VMs in each DC without increasing the total no. of VMs has reduced the response time by approx 33 ms as well as DC processing time 32 ms of all three algorithms as compared to CASE 2 and CASE 1. It also shows that throttled algorithm gives best result in all cases.

6.2 CASE 4 and CASE 5

After the above observations Throttled algorithm and Performance Optimized service broker algorithm is chosen for further analysis to reduce the response time and other parameters by observing DC processing time and changing the ratios further. DC processing time of throttled algorithm in CASE 3 is observed to set new VM ratios.

Table 7: DC processing time for throttled algorithm in CASE 3

Data Center	Avg (ms)
DC1	273.10
DC2	368.81
DC3	437.99
DC4	45.01

As observed from above table that DC4 is further being decreased as compare to CASE 2 so it shows we can remove further VMs from dc4 and give them to other DCs. So new VM ratio formed for CASE 4 - DC1- 85, DC2- 35, DC3- 70, and DC4- 50 and again analyzing like above ratio for CASE 5- DC1- 80 DC2- 40, DC3- 70, and DC4- 50 is made.

6.2.1 Result

After executing CASE 4 and CASE 5 following result are obtained as shown in table 6 below. In table below we have shown that how three ratios used have changed the result, here **DC-VM Ratio 1: CASE 3 VM ratio, DC-VM Ratio 2: CASE 4 VM ratio and DC-VM Ratio 3: CASE 5 VM ratio.**

Table 8. Result of CASE 4 and CASE 5

Load Balancing Algorithm	Overall response time (ms)	Data Center processing time (ms)	Overall Cost ($)
DC-VM Ratio 1	372.35	263.15	5892.15
DC-VM Ratio 2	370.82	261.94	5892.15
DC-VM Ratio 13	366.34	257.56	5892.15

6.2.2 Performance Analysis

Graph has been plotted for the above tabular data showing three VM ratio being used and their respective result

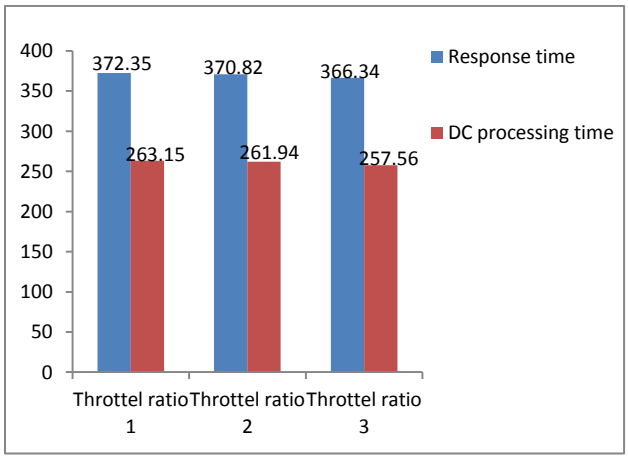

Fig 6: comparison of three ratios with Throttled algorithm and Performance Optimized service broker policy (CASE 4 and CASE 5)

Here it is observed that by changing the ratio of VM in each DC has reduced the response time and DC processing time without increase in cost hence the proposed work is successful in reducing the overhead and increasing the efficiency of DCs.

7. CONCLUSION AND FUTURE SCOPE

In this paper, analysis of the performance of three VM load balancing algorithm along with two service broker policy has been done for large scale application considering peak hours and bursty workloads. Results of the simulation showed that Throttled algorithm is efficient among all three algorithms and Performance Optimized service broker policy is best among the two policies. Further this research work also proposed a strategy to reduce the response time by changing the VM ratio in DCs. Simulation above showed that proposed strategy works and response time along with DC processing time got reduced considerably. Future work can be done in developing algorithms which can analyze the DC processing time of each DC and whenever their value falls below a defined threshold then VMs get migrated to other DC having higher value.

8. REFERENCES

[1] Lee Badger ,Tim Grance, Robert Patt-Corner and Jeff Voas, "Cloud Computing Synopsis and Recommendations", National Institute of Standards and Technology, Information Technology Laboratory, NIST Special Publication 800-146.

[2] A history of cloud computing. http://www.computerweekly.com/feature/A-history-of-cloud-computing.

[3] The Xen Project. http://xen.org/.

[4] KVM (Kernel-based Virtual Machine). http://www.linux-kvm.org/.

[5] S. G. Domanal, G. R. Mohana Reddy, "Load Balancing in cloud computing Using Modified Throttled Algorithm," IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp.1-7, October 2013.

[6] Hamid Shoja, Hossein Nahid and Reza Azizi, "A Comparative Survey On Load Balancing Algorithms In Cloud Computing", in proc. 5th IEEE International Conference on Computing,Communication and Networking Technologies (ICCCNT),pp.1-5, July 2014.jjvv

[7] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing:Challenges and Algorithms", in proc. Second Symposium on Network Cloud Computing and Applications (NCCA), IEEE, pp. 137-142, December 2012.

[8] Ashwin Kumar Kulkarn and Annappa .B," Load Balancing Strategy for Optimal Peak Hour Performance in Cloud Datacenters", IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES),pp.1-5, February 2015.

[9] B. Wickremasinghe, R.N.Calheiros, R.Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing", in proc. of the 24th International Conference on Advanced Information Networking and Applications (AINA 2010), Perth, Australia, 2010.

[10] Sandeep patel, Ritesh Pael and Hetal Patel, "CloudAnalyst : A Survey of Load Balancing Policies", International Journal of Computer Applications (0975 – 8887) Volume 117 – No. 21, May 2015.

[11] Rakesh Kumar Mishra, Sandeep Kumar and Sreenu Naik B, "Priority Based Round-Robin Service Broker Algorithm For Cloud-Analyst" , in proc. International Advance Computing Conference (IACC), IEEE, pp.878-881, February 2014.

[12] Internet World Stats. www.internetworldstats