RESEARCH ARTICLE OPEN ACCESS

An Efficient Resource Allocation for Heterogeneous Workloads in IaaS Clouds over AWS

[1]Shagun Sharma [2]Manish Sharma
[1]Research Scholar, [2]Professor Center for cloud infrastructure and security Suresh Gyan Vihar University, Jaipur
Corresponding Author; Shagun Sharma

ABSTRACT:
As the use of internet is increasing the corporate migrating their business from traditional computing to the cloud computing and thus no of user is increasing on cloud & load is also increasing. Thus to provide congestion free and reliable on demand service to client load balancing method is needed. Many algorithms is proposed for load balancing & auto scaling to handle the load .we can use cloud service to make load efficient model in cloud environment. This load efficient model will provide the load balancing, scaling capabilities and monitoring of solutions in the cloud environment.To achieve the above mentioned, we use public cloud services such as amazon's EC2, ELB. This research is divided into four parts such as load balancing, auto-scaling, latency based routing, resource monitoring. We will implement the individual service and test while providing load from external software tool Putty and we will produce the result for efficient load balancing.

Keyword: Load Balancing, EC2, Resource Monitoring, Load Optimization, Web Server.

Date Of Submission: 15-09-2019 Date Of Acceptance: 03-10-2019

I. INTRODUCTION

Cloud is an infrastructure or a platform which enables the computing of applications and services in reliable and elastic mode.Virtual platform means the hardware which is used to create a datacenter (the cloud) such as a server, storage, and network. Same as the software utility in the cloud is referred to as the services and applications provided to the users or clients.Cloud computing is one of the most emerging technologies which drew the attention of the entire technocrat in the field of computer science. Cloud computing is the technique which represents both cloud and the application (services). It is basically referred to as accessing computing service (resources and application) over the internet.

The cloud service provider handles the data from the remote location about that the client is unaware but an individual can access his data from anywhere simply by a system with an internet connection

Cloud computing has changed the classical computing environment in the IT industry. With cloud computing, many corporates are migrating their business from traditional computing to cloud computing in order to meet their business requirements. Cloud computing has been considered as the most revolutionary technology in the IT industry. For example, we can assume the whole internet as a single cloud in which people share space and resources from the pool of virtual space. The most important thing which is provided by cloud computing is the virtualization of resources.

NIST gives the standard definition of cloud computing as "It is the framework which enables the user or client in the computing environment to access on-demand services and a pool of resources such as servers, network, applications, etc. For example, when we save the image over the internet or send some files using the internet, we use cloud computing. Many websites are running on cloud computing because of its elasticity and auto-scaling feature. The major concern in cloud computing is the security of data as the data of the client is stored on a remote location. One can manipulate the data on the cloud server so the privacy of the data is the biggest factor in this technology. Service providers are working on this issue to resolve it.

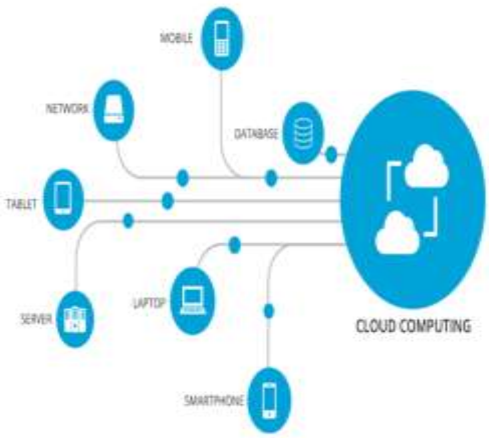


Figure 1. Example of Cloud Computing

II. REVIEW OF LITERATURE

In general, load balancing algorithms can be categorized as static and dynamic. On the basis of the current status of the node. Static load balancing technique distributes the load among the nodes on the basis of the past state while dynamic load balancing technique does not consider the past state or behavior for distributing the load, it only depends on the present state of the node. Hence it gives better results than the static one. Dynamic load balancing is of two types distributive and non-distributive. The main advantage of dynamic load balancing is, it will not stop the system if a node is down, the following criteria will be used to compare the LB algorithms.

- **Throughput-** Throughput is the total number of executed job in a fixed time.
- **Overhead Associated-** The amount of processing overhead to implement the LB i.e. inter-process communication.
- **Response Time –** It is defined as the total time in which the algorithm responds in the cloud network.
- **Resource Utilization –** How much amount of resource is utilized while implementing the algorithm
- **Scalability –** How efficiently it handles the load and scales the system according to need.

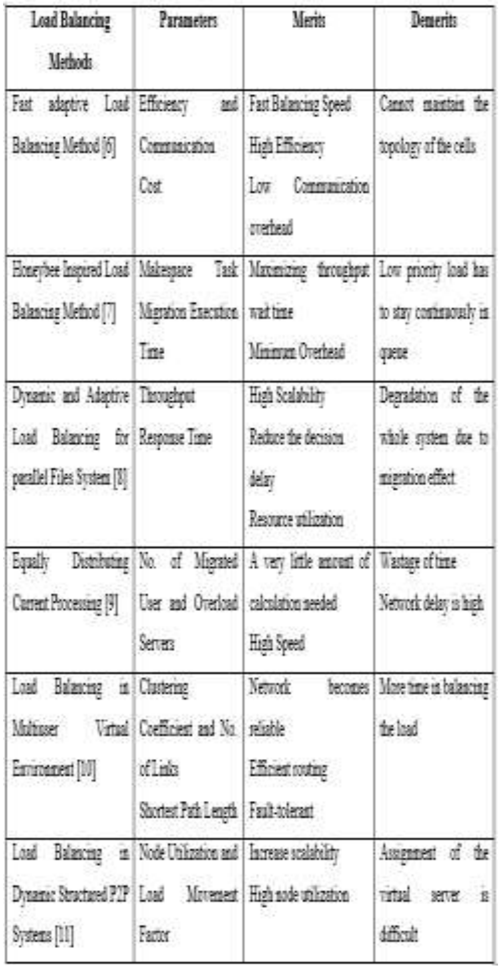
Lots of algorithms are proposed and implemented to handle the issue of load balancing in the cloud network some of the are

- **Load Balancing Method of Fast Adjustment:** This algorithm is proposedby D. Zhang et al. [6]. It is based on the binary tree which was used for dividing the big region into subdomains. This algorithm adjusts the load between the nodes from the local region to the global region. This algorithm portioned the region according to the binary tree. It must contain parent, child and leaf node. Partition is done between binary tree and cell indexes. The fast adaptive algorithm took less time with the speed of rebalancing of the load is faster. Benefits of this algorithm are low overhead, good speed of balancing with higher efficiency. The drawback of this algorithm is it doesn’t maintain the topology of nodes.
- **Honeybee Load Balancing Algorithm:** The basic idea of this algorithm is the behavioral pattern of a honey bee. Finders and reapers are two kinds of honeybee pattern are found. In the process of collecting honey, finder bees first search the honey source outside and after returning they indicate the availability of honey by doing waggle dance. Now to reap the honey from the source the reapers go outside their honeycomb and if honey is left there they indicate it by the woggle dance. Same as this M.S. George et al. [7] proposed a self-organized algorithm which was based on the decentralized honey bee theory. In this, a group made up of virtual server’s act as a honey bee. In this algorithm, each VMs maintain the priority of tasks and if an overloaded VMs want to assign priority task to under-loaded VMs, it checks which VMs has a minimum number of high priority task so that task is completed in less time. They formed a load queue sorted in ascending fashion. Information of available VMs is collected from the datacenter. High execution time, lower overhead and maximum throughput are advantages of this algorithm. The drawback of this algorithm is that a lower priority task always is in a queue if a higher priority task is available.
- **Dynamic and Adaptive Load Balancing:** This algorithm is made for transferring the files dynamically in a distributed architecture. B. Dong et al [8] presented this SALB algorithm for a large file system to handle the issue of file migration. In parallel file transfer system various issues like availability and scalability is solved by this algorithm. The central node is responsible for the decision making and if the central node is down the whole system will stop working thus reliability decreases. This issue is resolved as each virtual machine can decide to handle the load because the workload varies randomly. This algorithm addressed the issue of load balancing in the distributive file system but the whole system is degraded because of the side effect of migration.
- **Equally Distributing Current Processing:** It is a dynamic algorithm [9] which handles the load on the basis of priority and priority is decided on the basis of the size of the task. It is

a spread spectrum technique which first checks the priority of load and then distributes the load randomly on the under-loaded node.

- **Load Balancing Algorithm for Multiple User Virtual Environments:** This proposed architecture [10] is a hyper verse system which is responsible for hosting of the virtual world. In which load balancing algorithm was self-organized because of it. The whole network is divided into smaller cells. This network is handled by the public server. The basic idea behind this algorithm is to create a smaller hotspot to calculate the exact load of the object. This algorithm creates lots of overhead is higher initially thus required a large amount of time.

Table 1: Comparison of Different Load Balancing Techniques



Load Balancing Methods	Parameters	Merits	Demerits
Fast adaptive Load Balancing Method [6]	Efficiency and Communication Cost	Fast Balancing Speed High Efficiency Low Communication overhead	Cannot maintain the topology of the cells
Honeybee Inspired Load Balancing Method [7]	Makespace Task Migration Execution Time	Maximizing throughput wait time Minimum Overhead	Low priority load has to stay continuously in queue
Dynamic and Adaptive Load Balancing for parallel Files System [8]	Throughput Response Time	High Scalability Reduce the decision delay Resource utilization	Degradation of the whole system due to migration effect
Equally Distributing Current Processing [9]	No. of Migrated User and Overload Servers	A very little amount of calculation needed High Speed	Wastage of time Network delay is high
Load Balancing in Multiuser Virtual Environment [10]	Clustering Coefficient and No. of Links Shortest Path Length	Network becomes reliable Efficient routing Fault-tolerant	More time in balancing the load
Load Balancing in Dynamic Structured P2P Systems [11]	Node Utilization and Load Movement Factor	Increase scalability High node utilization	Assignment of the virtual server is difficult

III. PROPOSED METHOD

3.1 Cloud Division and Creating Instances-

The first thing to implement load balancing we must ensure availability. This can be done by dividing cloud in the different service region. We will use two regions in this system 1) US region 2) Asia region. In this service regions under the different availability zone, we will create the General purpose small instances using the Elastic cloud computing services. An Apache webserver running a web application will be deployed on the instances for analyzing the performances.

3.2 Load Balancing

Load balancing will be handled in four-parts
(i) latency based load balancing
(ii) local regional load balancing
(iii) auto-scaling to handle the excess load
(iv) Resource monitoring

- **Load Generation** – Virtual load will be generated through putty, the terminal emulation software on the different instances from the different PC in the lab.
- **Main Load Balancer (Latency based Load Balancing)** – The main load balancer which is a software load balancer based on the latency in which load is distributed among the different service region based upon the location of request, this is done by DNS resolver and create hosted zone. Latency based routing choose latent region instead of choosing any random service region to forward load for the process. This main load balancer will forward the traffic to the regional load balancer. In this proposed work we have created one hosted zone **"**www.cloudefy.in**"** and alias as us.cloudefy.in and asia.cloudefy.in and is created as a name and CNAME for each service region. The request coming for "www.cloudefy.in" will be first resolved by DNS resolver then will be sent to the other. We can also implement the weighted rule in latency based routing to improve load handling.
- **Local Load Balancing or Regional Load Balancing** – A number of secondary load balancer can be created. This load balancer balanced the load between the instances in a different availability zone. These load balancers can also balance the load in the same availability in the same service region. It can't balance the load between multiple service regions. Load balancing is done according to the Round Robin algorithm. A load balancer balances the load only for the protocol for which it is configured. The instances health check is performed on the HTTP protocol. In the proposed work we are balancing the load on HTTP /TCP protocol.
- **Load Handling** – To handle the excess load, the system must have scaling properties. In this proposed work, we are providing the auto-scaling feature by using the Auto-scaling group of EC2. In this, we have defined some scaling

policy to scale up and scale down the system. Conditions for auto-scaling can be defined by using different parameter available according to the application and incoming load pattern such as disk read and write up, CPU utilization, network utilization, we will use the maximum CPU utilization because of our service is web service and we are generating virtual load. Whenever the threshold is breached, the auto-scaling performs an action to accommodate the change in the system. User can also fix the size of the group to secure from application intensive attack.

3.3. Amazon Web Services Environment

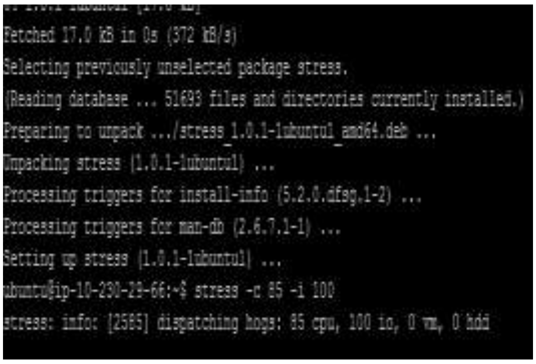
Cloud computing has a great impact in market and most exciting service provider is Amazon web services. In this research we used amazon's services to develop a load efficient model. AWS is one of the top Infrastructure as service cloud service provider or public cloud service provider, according to information available on internet AWS is a group of remote computing facilities form a cloud platform which is accessible over the internet. Most known service of AWS is EC2 & S3.

Amazon web service divided in service regions and located in 10 different location in the world these regions are US West (northern California), East Asia(Tokyo),US East (N.Verginia),US West (Oregon), China (Beijing), Brazil (Sau Paulo), Europe (Ireland), Australia (Sydney). Service region means the it contain whole country and the cloud services and data will be handled & stored in same region. Multiple availability zone is created in the each service region to avoided outages between the users.

Amazon web services officially started giving service as public cloud in 2006, user can access the client side application over HTTP and SOAP protocols. AWS provides the storage, computer power and other flexible service on the basis of pay as u go without any upfront cost.

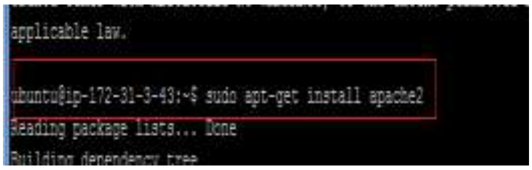
Load generation

Virtual load will be generation though putties the terminal emulation software on the different instances from the different PC in the lab. To generate the load in putty we must install stress component in apache server. This can be done by following command
"**Sudo apt-get install stress"** and the required load is generated by the "stress **–c 80 –m 50 -1= 1000**". As shoen in fig.



```
Fetched 17.0 kB in 0s (372 kB/s)
Selecting previously unselected package stress.
(Reading database ... 51693 files and directories currently installed.)
Preparing to unpack .../stress_1.0.1-1ubuntu1_amd64.deb ...
Unpacking stress (1.0.1-1ubuntu1) ...
Processing triggers for install-info (5.2.0.dfsg.1-2) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up stress (1.0.1-1ubuntu1) ...
ubuntu@ip-10-230-29-66:~$ stress -c 85 -i 100
stress: info: [2565] dispatching hogs: 85 cpu, 100 io, 0 vm, 0 hdd
```

Testing: - for testing we have deployed simple web application on each instance differentiating the different availability zone . An apche server is installed on these instances by the command "**<u>SUDO APT-GET APACHE2"</u>** this will install the apache web server on instances. Following images will show the creation of instances. We can connect to our instances by using either DNS address or IP address.



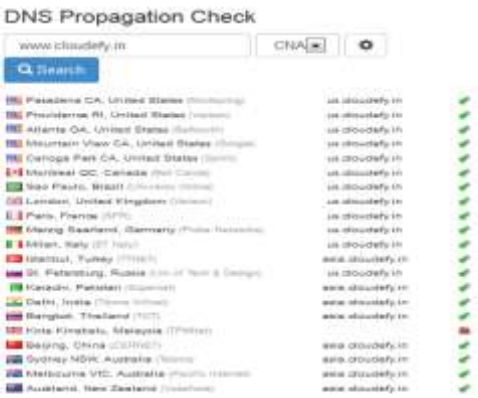
```
applicable law.

ubuntu@ip-172-31-3-43:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
```

The main feature of EC2 is its security mechanism which enables user with the inbound and outbound access control. While configuring the security groups and ACLs. While considering the economical point, EC2 provides instance on a very low rate. It has 3 types of instances -

- On-demand instances: On-demand instances are charged on the basis of per hour means this frees the user from the planning complexity.
- Reserved instances – These instances are reserved for frequent use. It needs the only on-time payment for the period of time one can share the reserved instances before the term expires.
- Spot instances: It is based on a bid system for unused instances and uses the instances for the time spot. Price does not exceed their bid system.

IV. RESULTS

With the proxy server and the main load balancer, we search www.cloudefy.in from different geographical regions and move to our nearest hosting service as shown in the image, this request leads to our first latency-based routing based in latency to the regional load balancer. 2) Represent us us.clodefy.in and asia.cloudefy.in requested Byalansarara loading process.



DNS Propagation Check

www.cloudefy.in CNA

Search

Pasadena CA, United States	us.cloudefy.in	✓
Providence RI, United States	us.cloudefy.in	✓
Atlanta GA, United States	us.cloudefy.in	✓
Mountain View CA, United States	us.cloudefy.in	✓
Canoga Park CA, United States	us.cloudefy.in	✓
Montreal QC, Canada	us.cloudefy.in	✓
Sao Paulo, Brazil	us.cloudefy.in	✓
London, United Kingdom	us.cloudefy.in	✓
Paris, France	us.cloudefy.in	✓
Merzig Saarland, Germany	us.cloudefy.in	✓
Milan, Italy	us.cloudefy.in	✓
Istanbul, Turkey	www.cloudefy.in	✓
St. Petersburg, Russia	us.cloudefy.in	✓
Karachi, Pakistan	www.cloudefy.in	✓
Delhi, India	www.cloudefy.in	✓
Bangkok, Thailand	www.cloudefy.in	✓
Kota Kinabalu, Malaysia		✗
Beijing, China	www.cloudefy.in	✓
Sydney NSW, Australia	www.cloudefy.in	✓
Melbourne VIC, Australia	www.cloudefy.in	✓
Auckland, New Zealand	www.cloudefy.in	✓

Fig 2: Server Based on region

Load balancer in regional load balancer distributes load in round robbery fashion. As shown in fig. The requested image is randomly generated and thrown on the load balancer. For example some virtual loads have been implemented as we can see that the load of the image is distributed in the examples so it is the load graph between the second suggested issues of balancing our EC2 as shown in the following images.

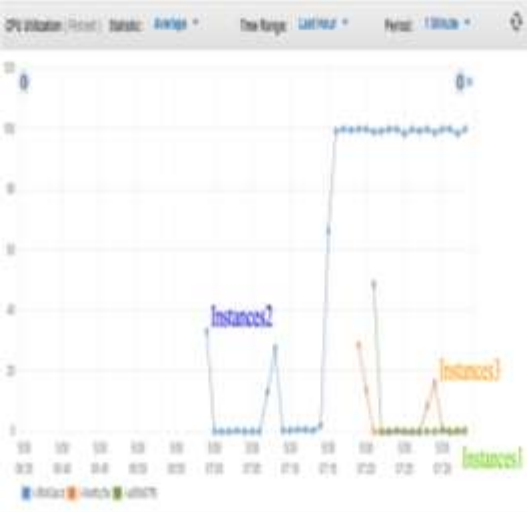


Fig: 03 load performance of EC2 Instance used

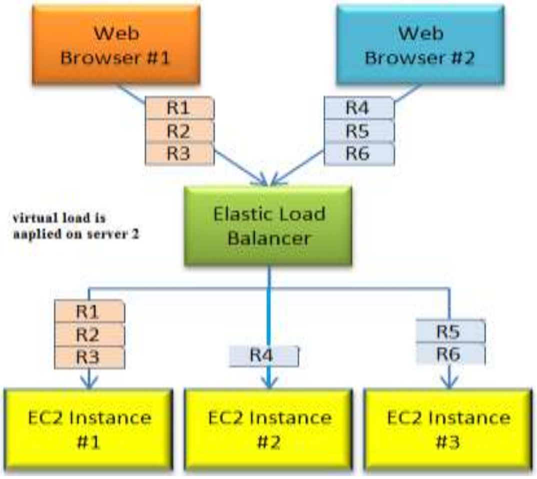
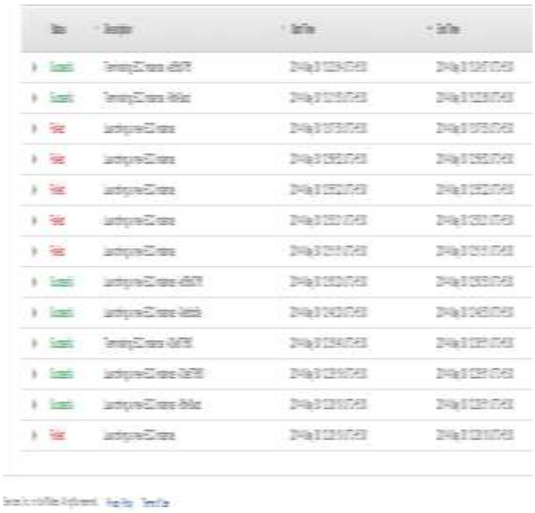


Fig: 04 Proposed Architecture of Auto Load balancing

Auto Scaling Results are collected from the Overview of Auto Scaling Group activity in Cloud Clock. Load pattern and polis are used, this result is in accordance with the load pattern sown in step 5. As we defined at the beginning the size of the primary group was 2 instances. However, we further define that when maximum CPU usage is low, 30% closes an instance that will appear in the fourth line from the bottom of the image. The auto scaling system automatically launches new instances (5th and 6th lines) when we apply the load. The auto scaling system closes the newly created instances after removing the litter. (First and second lines) Cloud network system error failed.



V. CONCLUSION

This work is focus on the load balancing technique and over the cloud computing and technique used for load balancing. There are lots of technique used like auto scaling, job based synchronization, and ant colony technique are used.

But when it comes to cloud the performance of load balancing technique is not up to mark. So improve the performance of cloud computing using load balancing we proposed an algorithm. In proposed algorithm we hosted the web server over the cloud and request is generated from the different client to the server. Cloud web server capture the request and analysis the load on server based on job nature it transfer the request of client to any particular server. We use amazon cloud environment for hosted the cloud servers. Result show that load balancing technique improve the job execution rate by 20% and when it compare with the exiting load balancing technique then its performance ration is better than ant colony technique and SJF and FCFS job shedding process .

REFERENCES

[1]. Eddy Caron, Luis Rodero-Merino, Frédéric Desprez, Adrian Muresan, “Auto-Scaling, Load Balancing and Monitoring in Commercial and Open-Source Clouds. [Research Report] RR-7857, pp.27.hal-00668713, INRIA. 2012.

[2]. Miss.Rudra Koteswaramma, “Client-Side Load Balancing and Resource Monitoring in Cloud”, International Journal of Engineering Research and ApplicationsISSN: 2248 9622, Vol. 2, Issue 6, pp.167-171, November-December 2012

[3]. N. Ajith Singh, M. Hemalatha, “An approach on semi distributed load balancing algorithmforcloud computing systems” International Journal of Computer Applications Vol-56 No.12 2012.

[4]. Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanazadeh, and Christopher, IPCSIT Vol-14, IACSIT Press Singapore 2011

[5]. Amazon web services cloud watch Web Site,November 2013. (https://aws.amazon.com/about-aws/whats-new/2013/)

[6]. Dongliang Zhang, Changjun Jiang,Shu Li, “A fast adaptive load balancing method for parallel particle-based simulations”, Simulation Modelling Practice and Theory 17 (2009) 1032–1042.

[7]. Dhinesh Babu L.D, P. VenkataKrishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments”, Applied Soft Computing, pp. 2292–2303, 2013.

[8]. Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao, Li Ruan, “A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers”, J. Parallel Distribution Computing. 72 (2012) 1254–1268.

[9]. Yunhua Deng, Rynson W.H. Lau, “Heat diffusion based dynamic load balancing for distributed virtual environments”, in: Proceedings of the17th ACM Symposium on Virtual Reality Software and Technology, ACM, 2010, pp. 203–210.

[10]. Markus Esch, Eric Tobias, “Decentralized scale-free network construction and load balancing in Massive Multiuser Virtual Environments”,in:Collaborative Computing: Networking, Applications and Worksharing, Collaborate Com, 2010, 6th International Conference on, IEEE, 2010, pp. 1–10.

[11]. B.Godfrey, K. Lakshminarayanan, S. Surana, R. Karp, I. Stoica, “Load balancing in dynamic structured P2P systems”, in: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 4, IEEE, 2004, pp. 2253–2262.

[12]. Ashalatha R and J. Agarkhed, “Evaluation of Auto Scaling and Load Balancing Featuresin Cloud”, International Journal of Computer Applications, Volume 117 – No. 6, pp. 30-33, May 2015.

[13]. T. Lorido-Botran · J. Miguel-Alonso and J. A. Lozano, “A Review of Auto-scaling Techniques for ElasticApplications in Cloud Environments”, Spinger Grid Computing,October 2014.

[14]. S. Taherizadeh and V. Stankovski, “Dynamic Multi-level Auto-scalingRules for Containerized Applications”, Computer and Communications Networks and Systems, The computer journal,The British Computer Society, May 2018.

[15]. W. Dawoud, I. Takouna, and C. Meinel. 2012. Elastic Virtual Machine for Fine-Grained Cloud Resource Provisioning.Communications in Computer and Information Science, Springer, Vol. 269. 11–25, 2012

[16]. Roy H. Campbell, Charles A. Kamhouaand Kevin A. Kwiat, “Assured Cloud Computing”, ISBN:9781119428633, IEEE Computer Society, 2018

[17]. C. JIANG,J. WAN,Xianghua XU,Yunfa LI,X. YOU and D. YU, “Dynamic Voltage/Frequency Scaling for Power Reduction in Data Centers: Enough or Not?”, IEEE ISECS International Colloquium on Computing, Communication, Control, and Management, pp. 428-431, 2009.

[18]. Dr.Vijay Kumar Tiwari and Shikha, “Security and Privacy of Identity Information in CloudComputing”, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), Volume 5, Issue 2, PP 7-16, 2018.

[19]. N Raveendran, “Impact of Cloud Computing on Data Mining System”, International Journal of Advanced Research in Computer Science, Volume 3, No. 6, Nov. (Special Issue), 2012

[20]. Joseph F. Ruscio, Michael A. Heffner and Srinidhi Varadarajan, “DejaVu: Transparent User-Level Checkpointing, Migration, and Recovery forDistributed Systems”, 1-4244-0910-1/07 IEEE, 2007

[21]. Qunying Sun and Zhiyuan Hu, “Security for Networks Virtual Access of Cloud Computing”, IEEE Computer Society Fourth International Conference on Multimedia Information Networking and Security, pp. 749-752, 2012

[22]. R. Ranjan, A. Harwood, and R. Buyya, “Peer-to-Peer Based Resource Discovery in Global Grids: A Tutorial”. IEEE Communications Surveys and Tutorials, Volume 10, Issue 2, Pages 6-33, IEEE Communication Society, 2008.

[23]. “Microsoft Azure.” [Online]. Available at http://azure.microsoft.com/en-us/

[24]. Yi Wei, K. Sukumar, C. Vecchiola, D. Karunamoorthy and R. Buyya, “Aneka Cloud Application and Its Integration with Windows Azure”, Chapter 27, Department of Innovation, Australia, March 2011.

Shagun Sharma" An Efficient Resource Allocation for Heterogeneous Workloads in IaaS Clouds over AWS" International Journal of Engineering Research and Applications (IJERA), vol. 9, no. 9, 2019, pp69-75