ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 1/ January 2014

# Virtualization Technology to Allocate Data Centre Resources Dynamically Based on Application Demands in Cloud Computing

**Namita R. Jain[1] , Rakesh Rajani[2]**

[1] PG Student, Department of Computer Engineering,
Alard College of Engg. And Mgmt., Hinjewadi, Pune, Maharashtra, India
[2] Professor, Department of Computer Engineering,
Alard College of Engg. And Mgmt., Hinjewadi, Pune, Maharashtra, India

## ABSTRACT

***Cloud computing is on demand service as it offers dynamic, flexible and efficient resource allocation for reliable and guaranteed services in pay-as-you-use manner to the customers. In Cloud computing multiple cloud users can request number of cloud services simultaneously, so there must be a provision that all resources are made available to requesting user in efficient manner to satisfy their need without compromising on the performance of the resources. Cloud computing has its era and become a new age technology that has got huge importance and potentials in enterprises and markets. Clouds can make it possible to access applications and associated data from anywhere, anytime. One of the major challenges in cloud computing is related to optimizing the resources being allocated. The other challenges of resource allocation are meeting customer demands, data center management and application requirements.***

***Here the design, implementation, and evaluation of a resource management system for cloud computing services is presented. System multiplexes virtual to physical resources adaptively based on the changing demand. Skewness metric is used to combine Virtual Machines with different resource characteristics appropriately so that the capacities of servers are well utilized. This algorithm achieves both overload avoidance and green computing for systems with multi resource constraints.***

## General Terms

Cloud Computing, VMware, Load Balancing, Physical machine, Virtual machine, Data Center

## Keywords

**Resource management, Virtualization, Overload Avoidance, Green Computing, Virtual Machine Monitor, Skewness**



## 1. INTRODUCTION

The notion of Cloud computing has not only reshaped the field of distributed systems but also fundamentally changed how businesses utilize computing today. While Cloud computing provides many advanced features, it still has some shortcomings such as the relatively high operating cost for both public and private Clouds. The area of Green computing is also becoming increasingly important in a world with limited energy resources and an ever-rising demand for more computational power. Studies have found that servers in many existing data centers are often severely underutilized due to over provisioning for the peak demand. Clouds can make it possible to access applications and associated data from anywhere, anytime. But one of the major challenges in cloud computing is resource optimization. The other challenges of resource allocation are meeting customer demands, data center management, application requirements, and dynamic scalability. The application is responsible to scale up and scale down the computer nodes dynamically as per the response time of the user's queries. The scheduling delay is the key factor which leads to the need of effective and dynamic load management system. The distributed resource allocation is the most challenging problem in the resource management problem. The modern data centers, operating under the Cloud computing model are accommodating a variety of applications. These applications range from small scale up to large scale. Those that run for a few seconds (e.g. serving requests of web applications such as e-commerce and social networks portals with transient workloads) to those that run for longer periods of time (e.g. simulations or large data set processing) on shared hardware platforms. The need to manage multiple applications in a data center creates the challenge of on-demand resource provisioning and allocation in response to time-varying workloads. The data center resources are allocated to applications, based on peak load

characteristics, in order to maintain isolation and provide performance securities.

In the last few years dynamic resource allocation based on application demands in cloud computing has attracted attention of the research community. They come up with innovative ideas, new ways or techniques to face this type of challenge. Since data centers host multiple applications on a common server platform; they can dynamically reallocate resources among different applications. Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources. This mapping is largely hidden from the cloud users. Users with the Amazon EC2 service, for example, do not know where their VM instances run. It is up to the cloud provider to make sure the underlying physical machines (PMs) have sufficient resources to meet their needs. VM live migration technology makes it possible to change the mapping between VMs and PMs While applications are running. However, a policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized.

## 2. RELATED WORK

Clouds can make it possible to access applications and associated data from anywhere, anytime. But one of the major challenges in cloud computing is resource optimization [4][5]. The other challenges of resource allocation are meeting customer demands, data center management, application requirements, and dynamic scalability. The application is responsible to scale up and scale down the computer nodes dynamically as per the response time of the user's queries [4]. The scheduling delay is the key factor which leads to the need of effective and dynamic load management system. The distributed resource allocation is the most challenging problem in the resource management problem. As we know, modern data centers, operating under the Cloud computing model are accommodating a variety of applications. These applications range from small scale up to large scale. Those that run for a few seconds (e.g. serving requests of web applications such as e-commerce and social networks portals with transient workloads) to those that run for longer periods of time (e.g. simulations or large data set processing) on shared hardware platforms. The need to manage multiple applications in a data center creates the challenge of on-demand resource provisioning and allocation in response to time-varying workloads. The data center resources are allocated to applications, based on peak load characteristics, in order to maintain isolation and provide performance securities [12].

In the last few years dynamic resource allocation based on application demands in cloud computing has attracted attention of the research community. They come up with innovative ideas, new ways or techniques to face this type of challenge. Since data centers host multiple applications on a common server platform; they can dynamically reallocate resources among different applications. Several allocation schemes have been proposed [13] that perform reallocation on such platforms. Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources which is proposed in [6]. Same as the VMware ESX Server use the random page sampling technique we use the same in it [9]. In [7] [8] VM live migration technology used which makes it possible to change the mapping between VMs and PMs, while applications are running. In [11] the multiplexing of VMs to PMs is managed using the Usher framework.

For data center performance some relative model has been proposed in literature [14]. A degree constraint is introduced in [15], and using this constraint a model of virtual machine allocation problem is developed. In [10] VM placement model is proposed. An efficient and economic resource allocation in high- performance computing environments is proposed in [16]. This paper, along with [5, 14] motivated us towards incorporating overload avoidance to managing load on cloud servers. The work in [5] supports green computing, overload avoidance and skewness algorithm which optimizes the number of servers in use.

## 3. EXISTING SYSTEM

A Virtual machine monitor (VMM) is a host program. It allows a single computer to support multiple and identical environments for execution. All the users see their systems as self-contained computers which is isolated from other users. Here every user is served by the same machine. Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources [6]. This mapping is actually hidden from the cloud users. For example users with the Amazon S3 or Amazon EC2 service, do not know where their VM instances run actually. It is always in the hand of cloud provider to make sure the underlying physical machines (PMs) have sufficient resources to meet their needs. As there are various VM live migration technology, it is up to them which makes it possible to change the mapping between VMs and PMs while applications are running [7][8]. The capacity of PMs can also be heterogeneous since multiple generations of hardware coexist in a data center[5].

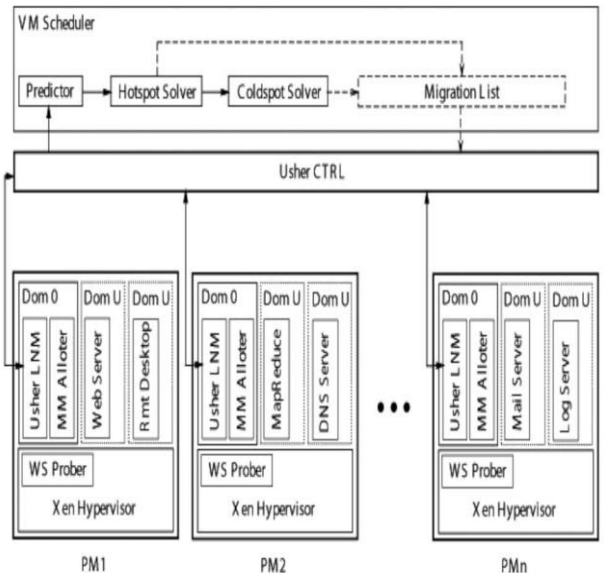


Fig. 1: Dynamic Resource Allocation Using Virtual Machine

Figure 1 shows the architecture of the existing system. Each

Physical Machine (PM) runs the virtual machine monitor (VMM) i.e. Xen hypervisor which supports a privileged domain 0 and one or more domain U [6]. Each VM in domain U incorporates one or more applications for example: Web server, remote desktop, DNS, Mail, Map/Reduce, etc. It is assumed that all PM's share backend storage. The multiplexing of VMs to PMs is managed using the Usher framework [11]. The main logic of this system is implemented as a set of plug-ins to Usher. Each node runs an Usher Local Node Manager (LNM) on domain 0. The use of the node is to collect the usage statistics of resources for each VM on it. By monitoring the scheduling events in Xen we can calculate the CPU and network usage [6]. Basically the memory usage within a VM is not visible to the hypervisor. It's possible by observing its swap activities. But, the guest OS is required to install a separate swap partition and it may be too late to adjust the memory allocation. That's why we implemented a working set prober (WS Prober) on each hypervisor to estimate the working set sizes of VMs running on it. As the VMware ESX Server use the random page sampling technique we use the same in it [9]. The statistics related information collected at each PM is forwarded to the Usher central controller (Usher CTRL) where VM scheduler runs. The VM Scheduler is invoked periodically as per the requirement. The VM scheduler receives the resource demand history of VMs, as well as the capacity and the load history of PMs, and the current layout of VMs on PMs from the LNM. The scheduler has several components such as- The predictor, which predicts the future resource demands of VMs and the future load of PMs based on past statistics related information. We can compute the load of a PM by accumulating the resource usage of its VMs. The LNM at each node tries to satisfy the new demands locally by adjusting the resource allocation of VMs sharing the same VMM. But, Xen can change the CPU allocation among the VMs by adjusting their weights in its CPU scheduler[5]. The MM Allotter on domain 0 of each node, as shown in figure is responsible for adjusting the local memory allocation. The hot spot solver in VM Scheduler notices if the resource consumption of any PM is above the hot verge (i.e., a hot spot) or not. If yes then some VMs running on them will be transferred away to reduce their load. The cold spot solver checks that the average utilization of actively used PMs (APMs) is below the green computing threshold or not. If so, some of those PMs could possibly be turned off to save energy. It recognizes the set of PMs whose consumption is below the cold threshold (i.e., cold spots) and then tries to transfer (migrate) away all their VMs. It then creates a migration list of VMs and compiles it and passes it to the Usher CTRL for execution[5].

## 4. ISSUES & CHALLENGES IN EXISTING SYSTEM

1. Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources.
2. This mapping is largely hidden from the cloud users. Users with the Amazon EC2 service, for example, do not know where their VM instances run.
3. It is up to the cloud provider to make sure the underlying physical machines (PMs) have sufficient resources to meet their needs.
4. VM live migration technology makes it possible to change the mapping between VMs and PMs While applications are running.
5. The capacity of PMs can also be heterogeneous because multiple generations of hardware coexist in a data center.
6. A policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized.
7. This is challenging when the resource needs of VMs are heterogeneous due to the diverse set of applications they run and vary with time as the workloads grow and shrink. The two main disadvantages are overload avoidance and green computing.

## 5. PROPOSED METHODOLOGY

In this paper, we present the design and implementation of an automatic, computerized resource management system. We tries to achieve the two main goals that are overload avoidance and green computing, illustrated as follows-

### 5.1 Skewness Algorithm

A normal distribution is a bell-shaped distribution of data where the mean, median and mode all coincide. A frequency curve showing a normal distribution would look like this:

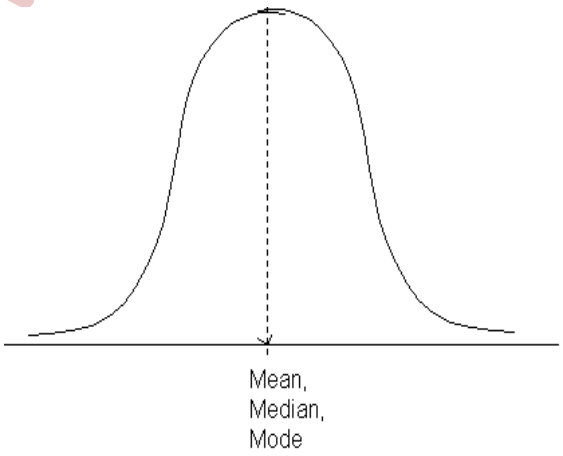


Fig. 2:- Frequency curve with normal distribution

In a normal distribution, approximately 68% of the values lie within one standard deviation of the mean and approximately 95% of the data lies within two standard deviations of the mean. If there are extreme values towards the positive end of a distribution, the distribution is said to be positively skewed. In a positively skewed distribution, the mean is greater than the mode. A negatively skewed distribution, on the other hand, has a mean which is less than the mode because of the presence of extreme values at the negative end of the distribution.

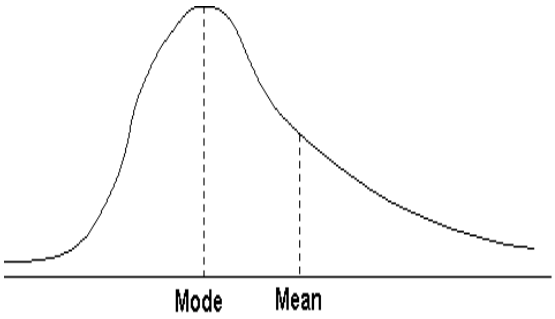


Fig.3:- Frequency curve with negatively skewed distribution

Skewness is a measure of the asymmetry or unevenness of the probability distribution. A distribution may either be positively or negatively skewed. The concept of skewness is introduced to compute the unevenness in the utilization of multiple resources on a server.

There are a number of ways of measuring skewness:

In favor of performance and stability, a pragmatic algorithm is designed i.e. skewness into the aforementioned Cloud system. It is inspired by the fact that if a PM runs too many memory-intensive VMs with light CPU load, much CPU resources will be wasted because it does not have enough memory for an extra VM. The concept of skewness is used to qualify the unevenness in the utilization of multiple resources on a server. Let n be the number of resources and ri be the utilization of the i-th resource[5]. The resource skewness of a server p is defined as follows

$$skewness(p) = \sqrt{\sum_{i=1}^{n}\left(\frac{r_i}{\overline{r}} - 1\right)^2},$$

where $\overline{r}$ is the average utilization of all resources for server p. By minimizing the skewness, the different types of workloads can be combined nicely and improve the overall utilization of server resources.

- Pearson's coefficient of skewness = mean-mode/Standard deviation
- Pearson's coefficient of skewness = 3(mean-median)/Standard deviation

where,

mean: arithmetic average

standard deviation: square root of the variance

variance: average of squared deviations from the mean

median: value that separates larger half of the numbers from the smaller half

mode: most frequent value found in the set

- Skewness is a statistic that is used to measure the symmetry of the distribution for a set of data. The skewness of an analysis domain is calculated as follows:
- Skewness = K3 / ESD 3

where:

K3 = [n · Si (Ei - En)3] / [(n-1)·(n-2)] if n ≥3;

K3 = 0 if n < 3

Summation is over all samples i in the region

n is the number of samples included in the summation.

Ei is the linear Sv of sample i (m2/m3) - set to 0 for any sample where Ei < mSv or Ei > MSv,

En is the observed Mean Energy of the region in linear units (m2/m3) = 10Sv/10

mSv is the minimum integration threshold (dB re 1 m-1) at time of processing.

MSv is the maximum integration threshold (dB re 1 m-1) at time of processing.

ESD Standard_deviation (Standard Deviation of the Energy)

$$E_{SD} = \sqrt{\sum_{i} \frac{(E_i - E_n)^2}{n-1}}$$

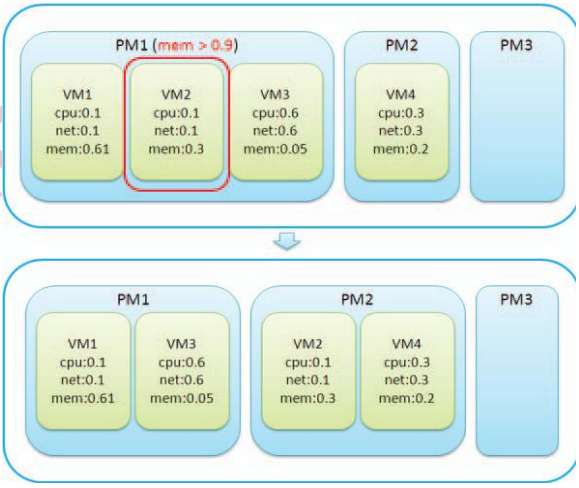


Fig. 4 System Architecture for Skewness Algorithm

## 5.2 Load Prediction Algorithm

It can capture the future resource usages of applications accurately without looking inside the VMs. The algorithm can capture the rising trend of resource usage patterns and help reduce the placement churn significantly. It is done by observing past logs generated and forecasting the future load.

Load prediction has significant impacts on resource allocation. With an over-estimated load, a scheduler may allocate more resources than necessary. Therefore some of the resources are wasted. On the contrary, with an under-estimated load, the resource allocation may be insufficient. Consequently, VOD user may complain the video is not fluent and online game players may get angry because they cannot control an avatar.

The two categories of load prediction algorithm are widely

adopted. One category composed of variations of the Exponentially Weighted Moving Average (EWMA) algorithm. It is designed based on the assumption that the future value of a random variable has strong relation to its recent history. It has been used in TCP for Round Trip Time (RTT) estimation for decades. Algorithms of the other category adopt the auto-regressive (AR) model. It requires more computation than EMWA based algorithms. But it can incorporate periodicity, which is hard to be utilized in EWMA alternatives, for better precision.

### *5.2.1 EWMA Variations*

With the original EWMA, load at time t is calculated by

$$E(t) = \alpha * O(t) + (1 - \alpha) * E(t-1),$$
$$0 \leq \alpha \leq 1,$$

where E(t) and O(t)are the estimated and the observed load at time t, respectively. The parameter alpha reflects a tradeoff between stability and responsiveness[5]. The larger the alpha is, the more agile the estimated load will be (low gain). On the contrary, the smaller the alpha is, the more stable the estimated load will be (high gain). The load prediction algorithm is a variation of EWMA. It uses a high gain EWMA and a low gain EWMA. If the latest observed load does not deviate much from recent observations, the low gain EWMA is used. Otherwise the high gain EWMA is used. This eliminates occasionally noisy observations. The output is further processed by a hysteresis filter for stabilization. The working set size estimator in ESX server also incorporates a similar technique. It uses three EWMAs with high, medium and low gain. The highest EWMA is selected as output to avoid under estimation as much as possible. A "Fast Up and Slow Down"(FUSD) predicting algorithm is designed for the load predictor in the VM Scheduler of the PKU Cloud. It is worth noticing that EWMA does not capture the rising trends of resource usage. For example, when a sequence of O(t) = 10; 20; 30; and 40 is seen, it is reasonable to predict the next value to be 50. Unfortunately, when alpha is between 0 and 1, the predicted value is always between the historical value and the observed one. This phenomenon easily cause under provisioning when load is rising. To reflect the "acceleration", an innovative approach is considered by setting alpha to a negative value. On the other hand, when the observed resource usage is going down, it should be conservative in reducing the estimation by using a normal alpha. That's why it is called "Fast Up and Slow Down". It dramatically reduces the number of hot spots and live migration for Skewness and bin-packing VM schedulers.

### *5.2.2 The AR Model*

In some works, future load is modeled as a linear function of several other factors such as the load history, time, or resource allocation. The parameters can be calculated by training with data in the past. Then the model can predict the future load. This methodology is called Auto-Regression (AR), represented as AR(p), where p is the number of factors considered in this model. AR model works well for periodical load. The Sandpiper VM scheduler adopts AR. It models the load at time t as a linear function of the average of n latest observations. It cannot utilize periodicity because it is unaware if the application is periodical. In the research on provisioning servers for connection intensive services, AR(n) is used to predict the number and login rate of MSN clients. The load is modeled as a linear function of six independent variables, two of the most recent observations and four of the observations at the same time in last four weeks. The results shows perfect fit between the predicted and the observed load. This is because the load of MSN clients presents perfect periodicity in its weekly pattern. The most popular Internet applications present such characteristics.

## 5.3 Green Computing

Green computing is defined as the study of planning, designing , manufacturing, consuming, and disposing of computers, servers, and its associated subsystems such as monitors, printers, storage devices, and networking and communications systems efficiently, effectively and successfully with minimal or no impact on the environment. The goals of green computing are to reduce the use of hazardous and harmful resources as well as maximizing the energy efficiency of the resources during the product's lifetime[12]. It also helps to promote the recyclability or biodegradability of invalid products and factory waste or wastage materials. Research related to this is continues into key areas such as making the use of computers as energy-efficient as possible, and designing algorithms and systems for efficiency-related computer skills. The number of PMs required in the system should be minimized as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off temporarily to save the energy[12].

Green Cloud computing is intended to accomplish the efficient processing as well as utilization of computing infrastructure. It is also used to minimize the energy consumption. In this way we can ensure that the future growth of Cloud computing is sustainable. Otherwise, Cloud computing with increasingly universal front-end client devices which are continuously interacting with back-end data centers will cause a huge growth of energy usage[3]. The following approaches are used for green computing, that are-

1) Product durability

2) Algorithmic efficiency & productivity

3) Resource allocation

4) Virtualization

5) Power management etc.

## 6. CONCLUSION

Dynamic resource allocation is growing need of cloud providers for more number of users and with the less response time. Cloud Computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the internet. Recent computers are sufficiently powerful to use virtualization to present the deception of many smaller VMs, each running a separate OS instance. We present a system that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. We introduce the concept of "skewness" to measure the

unevenness in the multidimensional resource utilization of a server.

## 7. ACKNOWLEDGMENTS

I wish to thank all the people who gave me an unending support right from stage the idea was conceived. I would like to thank Mr. Rakesh Rajani, my project guide for their helpful comments and suggestions. I express my sincere and profound thanks to Ms. Vani Hiremani & Ms. Kalpana Saharan, who always stood as the helping and guiding support for me.

## 8. REFERENCES

[1] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," technical report, Univ. of California, Berkeley, Feb. 2009

[2] Andrew J. Younge, Gregorvon Laszewski, Lizhe Wang "Efficient Resource Management for Cloud Computing Environments", Sonia Lopez-Alarcon, Warren Carithers

[3] Liang-Teh Lee, Kang-Yuan Liu, Hui-Yang Huang and Chia- Ying Tseng, "A Dynamic Resource Management with Energy Saving Mechanism for Supporting Cloud Computing," in International Journal of Grid and Distributed Computing Vol. 6, No.1, Feb, 2013.

[4] Chandrashekhar S. Pawar, R.B.Wagh, "A review of resource allocation policies in cloud computing", World Journal of Science and Technology 2012, 2(3):165-167 ISSN: 2231 – 2587

[5] Zhen Xiao, Senior Member, IEEE, Weijia Song, and Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 6, June 2013

[6] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," Proc. ACM Symp. Operating Systems Principles (SOSP '03), Oct. 2003.

[7] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," Proc. Symp. Networked Systems Design and Implementation (NSDI '05), May 2005.

[8] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast Transparent Migration for Virtual Machines," Proc. USENIX Ann. Technical Conf., 2005

[9] C.A. Waldspurger, "Memory Resource Management in VMware ESX Server," Proc. Symp. OS Design and Implementation (OSDI '02), Aug. 2002.

[10] Narander Kumar, Shalini Agarwal, Vipin Saxena, " Overload Avoidance Model using Optimal Placement of Virtual Machines in Cloud Data Cetres", International Journal of Computer Applications (0975 – 8887) Volume 73– No.11, July 2013

[11] M. McNett, D. Gupta, A. Vahdat, and G.M. Voelker, "Usher: An Extensible Framework for Managing Clusters of Virtual Machines," Proc. Large Installation System Administration Conf. (LISA '07), Nov. 2007.

[12] Rajkumar Buyya, Anton Beloglazov, and Jemal Abawajy, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges".

[13] A. Chandra, W. Gong, and P. Shenoy., " Dynamic Resource Allocation for Shared Data Centers Using Online Measurements".In Proceedings of Eleventh International Workshop on Quality of Service (IWQoS 2003), June 2003.

[14] Anton Beloglazov, Rajkumar Buyya "Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers Under Quality of Service Constraints", , IEEE Transactions on Parallel and Distributed Systems, vol. 24 no. 7, pp. 1366-1379, 2013

[15] Olivier Beaumont, Lionel Eyraud-Dubois, Christopher Thraves Caro, and Hejer Rejeb, "Heterogeneous Resource Allocation under Degree Constraints" , IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 5, 2013 doi :10.1109/tpds.2012.175.

[16] Kyle Chard, Kris Bubendorfer, " High Performance Resource Allocation Strategies for Computational Economies", IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, 2013, doi: 10.1109/TPDS.2012.102.