Australian Journal of Basic and Applied Sciences, 9(27) August 2015, Pages: 340-347

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Multi Agent Based Dynamic Resource Allocation In Cloud Environment For Improving Quality of Service

[1]M. Kandan and [2]R.Manimegalai

[1]*Research Scholar, Department of Information and Communication Engineering, Anna University, Chennai, Tamilnadu, India.*
[2]*Professor, Department of Computer Science and Engineering, Park College of Engineering and Technology, Coimbatore, Tamilnadu, India.*

ARTICLE INFO

Article history:
Received 23 June 2015
Accepted 25 August 2015
Available online 2 September 2015

Keywords:
Cloud Computing, Resource Allocation, Quality of Service (QoS), Service Oriented Architecture (SOA).

ABSTRACT

Cloud computing is an evolving paradigm in the computing world and is expected to make a heavy impact on human lives in future. Cloud computing involves management of various computing resources such as servers, networks and storage. As very large numbers of heterogeneous resources are used by thousands of clients in real time, it is essential to allocate resources dynamically. With diverse applications of Cloud, allocation of various resources such as memory, speed and bandwidth to jobs is an extremely complex task. In this paper, a Multi-Agent based Dynamic Resource Allocation (MADRA) strategy is proposed to improve the Quality of Service (QoS) in terms of time and energy efficiency. The simulation result shows the performance evaluation of the proposed MADRA approach than the existing approach Service Level Agreement (SLA) in various factors like throughput, delay, request completion time and energy consumption. The efficiency of MADRA approach in terms of resource allocation, time taken for request completion and remaining energy are obtained as 99.75%, 92% and 96.34% respectively compared to the existing approach.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: M.Kandan and R.Manimegalai., Multi Agent Based Dynamic Resource Allocation In Cloud Environment For Improving Quality of Service. ***Aust. J. Basic & Appl. Sci.,*** *9(27): 340-347, 2015*

INTRODUCTION

Cloud computing utilize the functionalities of Internet and acts as an ecosystem to connect the data centers and the servers, which are distributed across the globe. Cloud computing offers IT services to all the users worldwide. It hosts important applications that spread across various scientific and business domains. Resource allocation is one of the major and challenging issues in Cloud computing. Resource allocation mainly focuses on improving the performance of the Cloud with less energy consumption. Fuzzy and pattern recognition algorithms have been used to identify nodes with different capacities of storage and energy level (Yegnanarayana, B., 1994).

The traditional resource allocation algorithms, such as improved clonal selection algorithm (Wanneng Shu, Wei Wang and Yunji Wang, 2014) are not scalable for large Clouds and leads to performance degradation. Therefore to overcome this disadvantage, dynamic hierarchical approaches (Sabarish, B.A., S. Shanmugapriya, 2012) are adopted to reduce the communication traffic. Cloud data centers results into high operational cost and power consumption. Therefore, green Cloud technologies are evolving to reduce the operational cost and power consumption.

Corresponding Author: M.Kandan, Research Scholar, Department of Information and Communication Engineering, Anna University, Chennai, Tamilnadu, India.
Email: kandan.mit@gmail.com

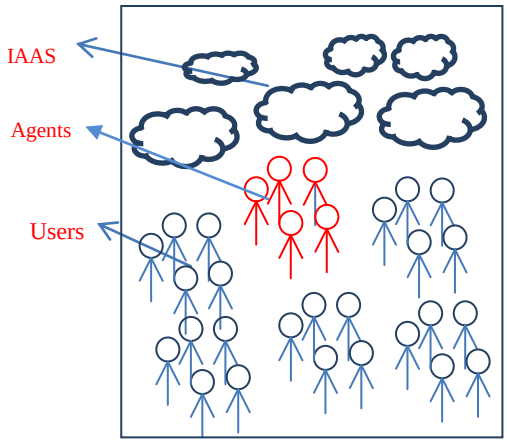

Fig. 1: Cloud Users vs. Cloud Resources

Resource Allocation and Scheduling (RAS) focuses on providing multiple services and space for the tasks submitted by users. There are two categories of allocation techniques, namely static and dynamic resource allocation. Meeting the user demands and requirements of applications is an extremely challenging issue in Cloud. Large number of Cloud users is waiting to access the Cloud resources which are shown in Figure-1. It describes the architecture of the Cloud environment and the resources in the Cloud. There are infrastructures, agents for connecting Cloud resources like infrastructure and users in the internet Cloud. Various kinds of resources can be connected by the appropriate agents in the Cloud.

Several combinations of computing nodes are used in Cloud to cater to the needs of jobs in execution. Nimbus, Cumulus is Cloud services that are used to obtain the configuration of sever. Table1 shows a typical set of Cloud computing client nodes along with their resources.

Table 1: Sample Computing Cores

ID	Processor	Memory (MB)	Hard Disk Capacity (MB)
1	1	256	10
2	1	512	15
3	1	1024	20
4	2	2048	20
5	2	1024	20
6	4	2048	40
7	4	2048	80

Table-1, explains the CPU utilization in terms of temporary memory (RAM) and persistent memory for Cloud environment. It says that the Cloud computing requires more space in temporary memory and less space in persistent memory. Also the memory utilization depends on the number of processor to be involved in the Cloud process.

Background Study:

Parallel processing and distributed computing are merged together and are named as Cloud computing. It is well known that the number of users and number of resources available in parallel, as well as in distributed are merely huge in numbers and it is not easy to fulfill the user request. User request for various resources and allocating the resources is a main problem to be solved nowadays. In this paper it is aimed to provide a better solution for Cloud in terms of resource allocation. Also each stage of the Cloud process is handled by agents like request handling, response providing, processing the request, searching the resource and matching the accurate resource for the request etc. There are various existing resources provides various kinds of solution for resource allocation. They are given here one by one. The work optimizes the number of servers in use by using virtualization technique to allocate resource dynamically as per user demands. This work also supports green computing. Here skewness is used to measure the unevenness in resource utilization by a server. When skewness is minimized then different types of workloads can be combined and this improves the overall utilization of server resources. The work also proposes a group of heuristics which prevents overload in the system while saving energy (Zhen Xiao, WeijiaSong and Qi Chen, 2013). In paper (Aftab Ahmed Chandio et al., 2014), resource allocation and energy efficient techniques have been used to deliver the QoS in large scale environment. This will greatly reduce operational cost. In paper (Aftab Ahmed Chandio et al., 2014), detailed study has been done on job scheduling in large scale system. The overall system

performance is improved while system utilization is increased in (Aftab Ahmed Chandio et al., 2014).

The problem in allocation of computer and network resources in a large Infrastructure-as-a-Service (IaaS) Cloud is considered in (Wuhib, F., et al., 2015). The problem of optimally allocating resources is formulated. Evaluation is done through simulation for a large Cloud to place Virtual Data Centers (VDCs) dynamically even under changing load conditions (Wuhib, F., et al., 2015). In paper (Abdul Hameed, et al., 2014), the author has tried to find a solution for improving the energy efficiency to allocate cod resources. The main aim of the paper is to find out the new open challenges in allocating energy efficient resource. Paper (Priyanka Mod, Professor Mayank Bhatt, 2014) proposes a method of Ant Colony Optimization Algorithm. This method uses genetic operations in enhancing the movement of ant towards a solution state. It provides an overall description of the techniques in Cloud. Many challenges were invented to promote the Cloud computing development in (Priyanka Mod, Professor Mayank Bhatt, 2014). In paper (DeepaVerma, R.K., Somani, 2014), discusses distinctive features of Cloud computing such as grid computing, utility computing etc., and their challenges. This paper proposes a tuning mechanism to provide efficient provision to guarantee quality of service in required level. It also includes considering the load balance for virtual migration.

A QoS constrained resource allocation technique is proposed in (GuiyiWeiAthanasios V. et al., 2010) which is suitable for sophisticated parallel computing problems. In (GuiyiWeiAthanasios V. et al., 2010), a solution based on Game theory is used, which involves a two-step procedure, to solve the problem of resource allocation. In the first step, a Binary Integer Programming method is used to solve the optimal problem independently, without considering resource multiplexing. Then, an evolutionary mechanism is used for allocating resources to different users optimally. The proposed resource allocation strategies in ((GuiyiWeiAthanasios V. et al., 2010)) consider both optimization and fairness into account and are scalable to a large proportion of Cloud-based computing services. In order to avoid lengthy response time a method is approached in Cloud computing for dynamic resource management. The tasks are performed by autonomous node agents in a data center. They carry out these events using the PROMETHEE method. This is a promising method to develop scalability and flexibility (YagızOnatYazır, et al., 2010). In the recent trend green Cloud computing approach is used to improve the computer resources utilization. It reduces energy consumption and thereby improving power efficiency. Various technologies were used such as virtualization and migration of virtual machines (Pragya, ManjeetGupta, 2015). The definition of "Cloud" is explained in this paper. That is an Infrastructure-as-a-Service (IaaS) system, where IT infrastructure is deployed as that in virtual machines. The tools and technologies are emerging at a growing popularity which will transform into an infrastructure of an organization into a Cloud. OpenNebula and Haizea are providing features which are not found in any other Cloud software (Sotomayor, B., et al., 2009).

In paper (Maurer, M., et al., 2013), the author tried to decrease the cost of the SLA violations, improves the performance, reduce the energy consumptions and allocate the resources. To do that, a hierarchically structured resource allocation is designed and implemented using knowledge management techniques. The Virtual Machines (VM) reconfiguration is applied each time of resource allocation and it takes more time and reduces the availability of the resources. To improve this, MADRA approach is applied for dynamic resource allocation with improved QoS.

Proposed Approach:

Multi-Agent based Dynamic Resource Allocation (MADRA):

A Cloud needs to focus on the following issues in order to move towards Green Cloud.

1. Reducing the energy consumption
2. Speed up the process to reduce power drain.
3. Optimization of security algorithms for reduced power.

Cloud helps to migrate, share and compute the data from a remote location with less impact on system performance. Resource allocation is a challenging issue in Cloud computing since, a remote user can access resources from anywhere at any time. It is impossible to access resources directly from remote, but it is possible by using various interfacing protocols such as SOAP, WEB-APIs and virtualization. Resources can be allocated statically or dynamically to user request demand.

A dynamic resource allocation technique is used to reduce the time and cost. The proposed solution improves the energy by choosing best resource and allocate to the client. Currently, Cloud users request the resources using REQ-RES. It is necessary to change the unpredictable requests of resource allocation. The main objective of the resource allocation in Cloud is either optimized or improved QoS in terms of time, cost and communication delay. Most of the resource allocation challenges are related to energy efficiency.

1. Resource allocation depends on the work assignment among Cloud users.
2. Resource allocation should be power consumption based, cost effective and time effective.
3. Overall performance of the Cloud environment including clients and servers, should be increased and it should be flexible to the other applications.

The main objective of this paper is to provide a fast, efficient dynamic resource allocation in Cloud environment. Figure 2, shows the proposed dynamic resource allocation model in Cloud environment.

Cloud users submit their requests to the server from anywhere in the world. It is very difficult to access the number of request coming to the server at a particular instant of time from various locations for various resources. It is important to record the Cloud user requests and resource needed and request deployment details. Multi-Agent based Dynamic Resource Allocation acts as an interface among the Cloud environment and the Cloud users. It necessitates the interaction of the following agents to improve the QoS. The following agents are designed in the proposed technique, for efficient resource allocation.

- Green Navigator Agent (GNA)
- Service Analyzer Agent (SAA)
- User Profile Agent (UPA)
- Pricing Agent (PA)
- QoS Parameter Monitoring Agent (QPMA)
- Service Scheduling Agent (SSA)

Green Navigator Agent assigns priority for the Cloud user requests. In case of web applications, 95% of the Cloud user requests are served within 3 seconds. The Service Analyzer Agent interprets and analyzes the incoming service requirement from Cloud users. It decides whether to accept or reject the submitted request. To provide accurate decision, SAA should know the latest information of load, energy from the VM manager and Energy monitor. User Profile Agent collects the behavior and characteristics of Cloud user. Important Cloud request are granted special privileges and prioritized over the other users. Pricing Agent decides about the cost for incoming services based on the time, demand and supply needed for the computing resources with all facilities. PA assigns priority for the services and allocates cost for the resources efficiently.

The Monitoring Agent analyzes the services and its needs in terms of energy, memory, time and throughput and then determines changing the states i.e. on/off, load/unload of the physical machines. Service Scheduling Agent assigns requests to VMs and determines resource entitlements for allocated VMs.

It also decides when VMs are to be added or removed to meet demand. The virtual machine manager maintains the availability of the virtual machines and the resources. It also takes care of the virtual migration across physical machines.

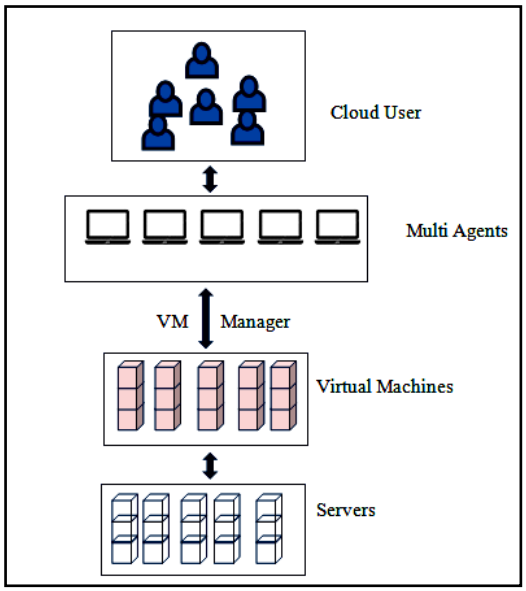

Fig. 2: Proposed Multi Agent based Model for Resource Allocation

Virtual Machines:

In Cloud, Large number of virtual machines are dynamically connected and disconnected to handle single request. It provides more flexibility for partitioning the resources in a single machine into different requirement for the service requests. Several virtual machines can concurrently access by different OS based environments dynamically. This process saves energy, time and cost.

In our proposed work, the entire process is decomposed into several sub-process is assigned to several agents in the Cloud. Then the resources are ranked in terms of time, cost, quality and availability. The chosen resource should be matched with Cloud user requirement. Finally in terms of QoS, all the resources are evaluated by assigning a score. From which the best score is selected and allocated for the appropriate incoming request dynamically shown in Figure 3.

The entire task ***T*** is divided into several sub-tasks such as

$[ST_1, ST_2, ST_3, \ldots . ST_n]$.

The quality scores $[QS_1, QS_2, QS_3, \ldots, QS_n]$

$$QS_i = QS_i(time) + QS_i(cost) + QS_i(availability) \quad (1)$$

Where, T represents the task,

$ST \rightarrow$ Denotes the sub-task

$QS \rightarrow$ Denotes the quality score

$QS_i \rightarrow$ is the ith job based resource quality score, where it can be calculated by adding the quality score in terms of time, cost, resource availability.

The Quality Score can be obtained using the following equation as:

$$\text{If the } \max(QS) = \max\left\{ \sum_{i=1}^{n} QS(i) \right\} \quad (2)$$

max(QS) →is the highest quality score among various resource request sent from Cloud user. Maximum value is obtained by comparing all the quality score values and selecting the highest score value.

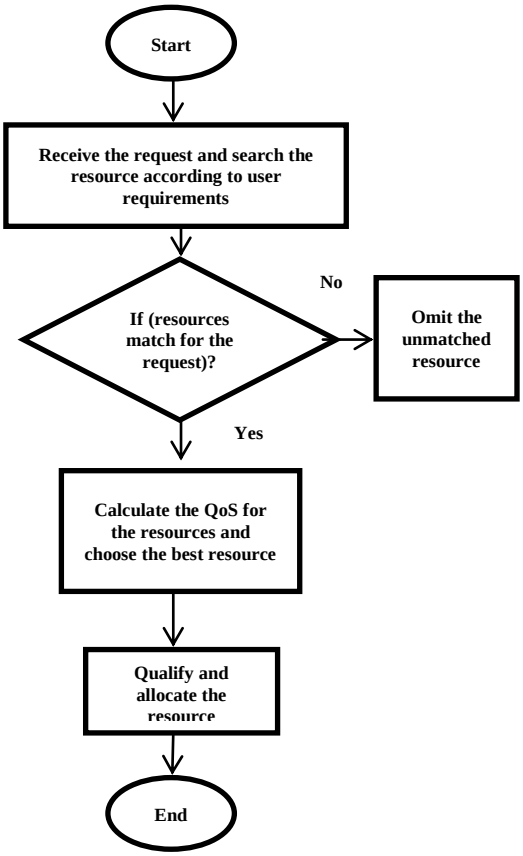

Fig. 3: Steps Involved in the Proposed Resource Allocation Technique

Simulation and Result:

The proposed Multi-Agent based Dynamic Resource Allocation (MADRA) strategy is simulated in CloudSim. The objective of the CloudSim tool is to provide a simulation framework which enables modeling, simulation and experimenting of emerging Cloud computing IAAS, SAAS, and PAAS based applications, application services. It allows the users to focus on a specific process and resource can be investigated without any programming issues. In the simulation, a data center with 50 heterogeneous physical nodes having CPU with performance equivalent to 1000, 2000 or more than 2000 million instructions per second, 8-GB RAM and 1 TB of storage is created. When the user submits a request it is processed and the capacity of the resource is simulated under the data center. Different kinds of web applications are running on each VM with different workloads. This can be used as model for creating the utilization of CPU according to the requirement randomly. Each experiment is simulated more than 10 times and the results are analyzed.

Table 2: Simulation Setting

Entity	Tool/Software
Operating System	Windows 8
Simulation Engine	CloudSim 3.0
Front-end IDE	NetBeans 8.0.3
Programming Language	Java
Performance Monitor	Cloud Analyst
Results Analysis	Cloud Report
Real Test-Bed	Xen-Server Based Cloud

The results obtained using simulations are shown in Figure 4, 5 and 6. Figure-4 shows the numbers of resources chosen for the best resource for the user requests. Since the numbers of Cloud requests are more at the same time, choosing the resources is also more in terms of quality score computed by the agents. The number of matching (selected) resources is not constant for all the iteration. It depends on the availability of the resources and Quality Score (QS).

MADRA utilizes the agents in various stages to reduce the request completion time. Also, one of the agents is investigating the resources, which resource is matched accurately for the user request. The quality (quality-score) of the resource can be calculated by investigating the resource name, time taken to allocate the resource, resource availability and etc. Using this quality score the appropriate resource can be chosen, which can provide better performance in the Cloud. By this way MADRA is decided as a better approach for resource allocation effectively. For example in Figure-4, more number of requests is entering into Cloud for requesting resources. But MADRA can allocate only relevant, best resources to the request. When 50 requests comes into the Cloud our MADRA can provide resources only for 23 request, since only 23 resources are the qualified resources can be allocated efficiently. Else, if the Cloud tries to allocate 50 resources for 50 requests, it cannot be confirmed that all the resources are the relevant resources to the requests.

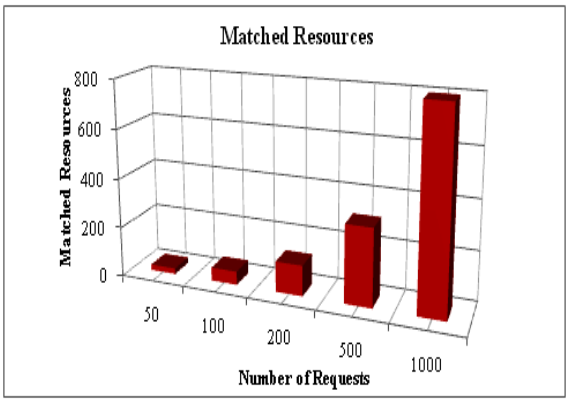

Fig. 4: Number of Request vs. Number of Selected Resources

Figure-5. Illustrates the time taken for resource allocation. A user request is not unique and does not require same resource by all the users. Thus, investigating and selecting a matched resource for one request in terms of QoS takes some time interval. From our obtained results, it is clear that, when request increases the time taken for resource allocation also increases. The time taken to allocate the resources depends on the resource availability, distance among the resources and the client (request place). For example, time taken to complete the resource allocation for 50 resources is 40 seconds, so it is taking 1.1 second for completing one resource allocation. Hence MADRA is efficient in terms of time

.

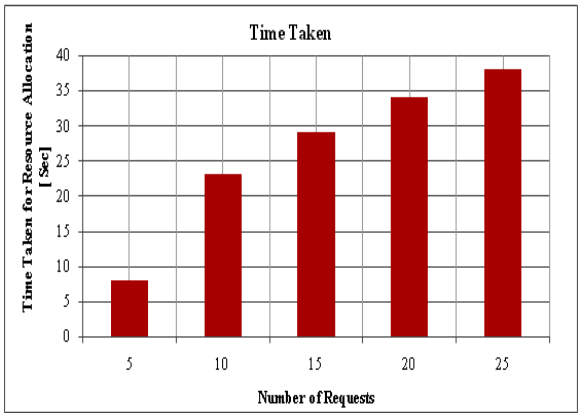

Fig. 5: Number of Request vs. Time

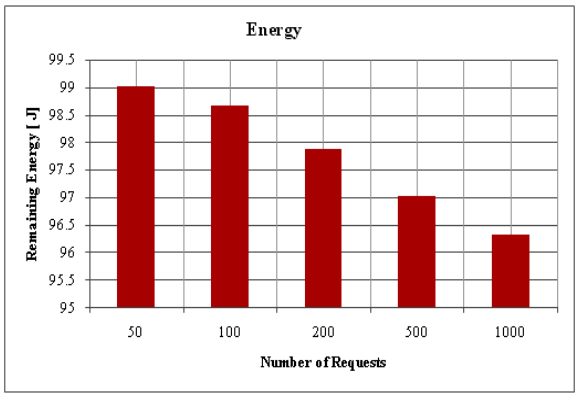

Fig. 6: Number of Request vs. Energy Consumption

Energy consumption proportionally increases according to the increased number of resources. The energy consumption can be obtained by calculating the remaining energy by subtracting the consumed energy from the initial energy. From the remaining energy the amount of energy consumed can be identified.

Figure-6 shows the remaining energy in joules. To allocate the resources and time taken to allocate the resources Cloud entity needs some amount of energy. From the Figure-6, it is clear that according to the number of resources, the energy is consumed. Figure-6, illustrates the energy consumption for resource allocation obtained in the simulation based experiment. A resource needs certain amount of energy for its each state like active, service, and live etc. When a resource is investigated, it should be active and involved in job, so some amount of energy is spent for the resources. While number of resource and time increases, the energy consumption also increases. By reducing the resource active state, processing speed of the energy can be saved. Most of the results obtained from the CLOUD-SIM tool is in the form of online reports. It is also simulated for large sized problems such as 50, 100, 500, 1000 and 5000 requests and the performance is evaluated. When a request increases the number of resources matched for the requests also increases. The resources availability can be verified dynamically while evaluating the Cloud user request by comparing the request based resources by the multiple agents. Since multiple agents are working simultaneously the time complexity is not affecting the Cloud computing based resource allocation.

The simulation results illustrates that our approach is efficient and potential for large data center based Cloud computing with centralization. Any CPU deployment in Cloud does not going to affect the migration with the Cloud server. The multi agents smoothly verify and distribute the overall resource utilization at the maximum less than or equaling to the user request.

Conclusion:

Our main objective is to allocate the resource speedily and dynamically. It is concerned that the QoS parameters should be improved during resource allocation. For this, MADRA approach is proposed using a number of agents involved in the resource investigation and allocation process in the server side. Each agent has their own assigned job and works simultaneously and individually, but the agents are correlated with one another. From the simulation results, it is obtained that our proposed MADRA approach outperforms well in terms of resource allocation, request completion time and

reduced energy consumption as 99.75%, 92% and 96.34% respectively than the existing approach. It is proved that MADRA is a good approach and it is efficient for resource allocations dynamically even the number of user requests are high. Our future work, the MADRA is verified and evaluated in terms of all the QoS parameters in a large environment in real time Laboratory.

REFERENCES

Abdul Hameed, Alireza Khoshkbar foroushha, Rajiv Ranjan, Prem Prakash Jayaraman, Joanna Kolodziej, Pavan Balaji, SheraliZeadally, Qutaibah Marwan Malluhi, Nikos Tziritas, Abhinav Vishnu, Samee U. Khan, Albert Zomaya, 2014. "A Survey and Taxonomy on Energy Efficient Resource Allocation Techniques for Cloud Computing Systems", Journal on Computing.

Aftab Ahmed Chandio, Kashif Bilal, Nikos Tziritas, Zhibin Yu, Qingshan Jiang, Samee U, Cheng-ZhongXu, 2014. "A Comparative Study on Resource Allocation and Energy Efficient Job Scheduling Strategies in Large-scale Parallel Computing Systems", Journal of Cluster Computing, 17(4): 1349-1367.

Cumulus services [online]. Available: http://www.nimbusproject.org/docs/ current/admin/ reference.html#cumulus

DeepaVerma, R.K., Somani, 2014."Tune the Resource Provisioning and Virtual Machine Migration for Cloud Environment", Journal of Engineering Science and Innovative Technology (IJESIT) 3(6): 193-200.

GuiyiWeiAthanasios V. Vasilakos YaoZheng NaixueXiong, 2010. "A Game-Theoretic Method of Fair Resource Allocation for Cloud Computing Services", The Journal of Supercomputing, 54(2): 252-269.

Maurer, M., I. Brandic and R. Sakellariou, 2013. "Adaptive Resource Configuration for Cloud Infrastructure Management", Future Generation Computer Systems, 29(2): 472-487.

Nimbus services [online]. Available: http://www.nimbusproject.org/docs/current/admin/reference.html#nimbusweb-config.

Pragya, ManjeetGupta, 2015. "A Review on Energy Efficient Techniques in Green Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering, 5(3): 550-554.

Priyanka Mod, Professor Mayank Bhatt, 2014. "A Survey on Dynamic Resource Allocation Technique in Cloud Environment", International Journal of Advanced Research in Computer and Communication Engineering, 3(8): 7815-7818.

Sabarish, B.A., S. Shanmugapriya, 2012."Improved Data Discrimination in Wireless Sensor Networks", Scientific Research, Wireless Sensor Network, 4(4): 117-119.

Sotomayor, B., R.S. Montero, I.M. Llorente and I. Foster, 2009. "Virtual Infrastructure Management in Private and Hybrid Clouds", IEEE Journal on Internet Computing, 13(5): 14-22.

Wanneng Shu, Wei Wang and Yunji Wang, 2014. "A Novel Energy-efficient Resource Allocation Algorithm based on Immune Clonal Optimization for Green Cloud Computing", EURASIP Journal on Wireless Communications and Networking, arXiv:1405.4618, 64.

Wuhib, F., R. Yanggratoke and R. Stadler, 2015. "Allocating Compute and Network Resources under Management Objectives in Large-Scale Clouds", Journal of Network and Systems Management, 23(1): 111-136.

YagızOnatYazır, Chris Matthews, RoozbehFarahbod, Stephen Neville, Adel Guitouni, SudhakarGanti and Yvonne Coady, 2010. "Dynamic Resource Allocation in Computing Clouds using Distributed Multiple Criteria Decision Analysis", IEEE 3rd International Conference on Cloud Computing, pp: 91-98.

Yegnanarayana, B., 1994. "Artificial neural networks for pattern recognition", Sadhana, 19(2): 189-238.

Zhen Xiao, WeijiaSong and Qi Chen, 2013."Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment", IEEE Transactions on Parallel and Distributed systems, 24(6): 1107-1117.