Cloud Load Balancing Techniques : A Step Towards Green Computing

Nidhi Jain Kansal[1], Inderveer Chana[2]

[1] **Computer Science and Engineering Department, Thapar University**
Patiala-147004, Punjab, India

[2] **Computer Science and Engineering Department, Thapar University**
Patiala-147004, Punjab, India

Abstract

Cloud computing is emerging as a new paradigm of large-scale distributed computing. It is a framework for enabling convenient, on-demand network access to a shared pool of computing resources. Load balancing is one of the main challenges in cloud computing which is required to distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed. It helps in optimal utilization of resources and hence in enhancing the performance of the system. The goal of load balancing is to minimize the resource consumption which will further reduce energy consumption and carbon emission rate that is the dire need of cloud computing. This determines the need of new metrics, energy consumption and carbon emission for energy-efficient load balancing in cloud computing. This paper discusses the existing load balancing techniques in cloud computing and further compares them based on various parameters like performance, scalability, associated overhead etc. that are considered in different techniques. It further discusses these techniques from energy consumption and carbon emission perspective.

Keywords: *Cloud computing, Load balancing, Energy efficiency, Green computing.*

1. Introduction

Cloud computing is emerging as a new paradigm of large-scale distributed computing. It has moved computing and data away from desktop and portable PCs, into large data centers [1]. It has the capability to harness the power of Internet and wide area network (WAN) to use resources that are available remotely, thereby providing cost-effective solution to most of the real life requirements [26]. It provides the scalable IT resources such as applications and services, as well as the infrastructure on which they operate, over the Internet, on pay-per-use basis to adjust the capacity quickly and easily. It helps to accommodate changes in demand and helps any organization in avoiding the capital costs of software and hardware [2] [3]. Thus, cloud computing is a framework for enabling a suitable, on-demand network access to a shared pool of computing resources (e.g. networks, servers, storage, applications, and services). These resources can be provisioned and de-provisioned quickly with minimal management effort or service provider interaction. This further helps in promoting availability [4]. Due to the exponential growth of cloud computing, it has been widely adopted by the industry and there is a rapid expansion in data-centers. This expansion has caused the dramatic increase in energy use and its impact on the environment in terms of carbon footprints. The link between energy consumption and carbon emission has given rise to an energy management issue which is to improve energy-efficiency [19] in cloud computing to achieve Green computing [6]. Besides this, there are many other existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation etc. that have not been fully addressed [5] [19]. Virtual Machine Migration enabled by virtualization can help in balancing load, enabling highly responsive provisioning and avoiding hot-spots in data centers thereby reducing power consumption. Server Consolidation helps in improving resource utilization by consolidating various VMs residing on multiple under-utilized servers onto a single server, thereby turning off unused servers, hence reducing energy consumption in a cloud computing environment [19]. Load balancing can help in reducing energy consumption by evenly distributing the load and minimizing the resource consumption. This paper focuses on the prevalent load balancing techniques in cloud computing environment.

Load balancing is one of the central issues in cloud computing [5]. It is a mechanism that distributes the dynamic local workload evenly across all the nodes in the whole cloud to avoid a situation where some nodes are heavily loaded while others are idle or doing little work. It helps to achieve a high user satisfaction and resource utilization ratio, hence improving the overall performance and resource utility of the system. It also ensures that every computing resource is distributed efficiently and fairly [13]. It further prevents bottlenecks of the system which may occur due to load imbalance. When one or more components of any service fail, load balancing helps in continuation of the service by implementing fair-over, i.e. in provisioning and de-provisioning of instances of applications without fail. It also ensures that every computing resource is distributed efficiently and fairly [5] [9]. Consumption of resources and conservation of energy are not always a prime focus of discussion in cloud computing. However, resource consumption can be kept to a minimum with proper load balancing which not only helps in reducing costs but making enterprises greener [6] [13]. Scalability which is one of the very important features of cloud computing is also enabled by load balancing. Hence, improving resource utility and the performance of a distributed system in such a way will reduce the energy consumption and carbon footprints to achieve Green computing [6] [10] [12].

The motivation of the survey of existing load balancing techniques in cloud computing is to encourage the amateur researcher to contribute in developing more efficient load balancing algorithms. This will benefit interested researchers to carry out further work in this research area. This paper is organized as follows: Section II discusses Green computing in clouds, Section III presents the need of load balancing in clouds, Section IV shows the study and analysis of the existing load balancing techniques in cloud computing, Section V identifies the metrics considered in the existing load balancing techniques and carries out the comparison between them based on those identified metrics and Section VI concludes the paper. To the best of our knowledge, none of the techniques has focused on energy consumption and carbon emission factors that is a dire need of cloud computing.

2. Green Computing in Clouds

Green Computing [6], or Green IT, is the practice of implementing policies and procedures that improve the efficiency of computing resources in such a way as to reduce the energy consumption and environmental impact of their utilization [10] [12].

As High Performance Computing (HPC) is becoming popular in commercial and consumer IT applications, it needs the ability to gain rapid and scalable access to high-end computing capabilities. This computing infrastructure is provided by cloud computing by making use of data-centers. It helps the HPC users in an on-demand and payable access to their applications and data, anywhere from a cloud [7]. Cloud computing data-centers have been enabled by high-speed computer networks that allow applications to run more efficiently on these remote, broadband computer networks, compared to local personal computers. These data-centers cost less for application hosting and operation than individual application software licenses running on clusters of on-site computer clusters [11]. However, the explosion of cloud computing networks and the growing demand drastically increases the energy consumption of data-centers, which has become a critical issue and a major concern for both industry and society [8]. This increase in energy consumption not only increases energy cost but also increases carbon-emission. High energy cost results in reducing cloud providers' profit margin and high carbon emission is not good for the environment [7]. Hence, energy-efficient solutions that can address the high energy consumption, both from the perspective of the cloud provider and the environment are required. This is a dire need of cloud computing to achieve Green computing. This whole scenario is depicted in Fig. 1. Load balancing can be one such energy-saving solution in cloud computing environment.

3. Need of Load Balancing in Cloud Computing

Load balancing in clouds is a mechanism that distributes the excess dynamic local workload evenly across all the nodes. It is used to achieve a high user satisfaction and resource utilization ratio [15], making sure that no single node is overwhelmed, hence improving the overall performance of the system. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption. It also helps in implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning, reducing response time etc.

Apart from the above-mentioned factors, load balancing is also required to achieve Green computing in clouds which can be done with the help of the following two factors:

• *Reducing Energy Consumption* - Load balancing helps in avoiding overheating by balancing the workload across all the nodes of a cloud, hence reducing the amount of energy consumed.

• *Reducing Carbon Emission* - Energy consumption and carbon emission go hand in hand. The more the energy consumed, higher is the carbon footprint. As the energy consumption is reduced with the help of Load balancing, so is the carbon emission helping in achieving Green computing.

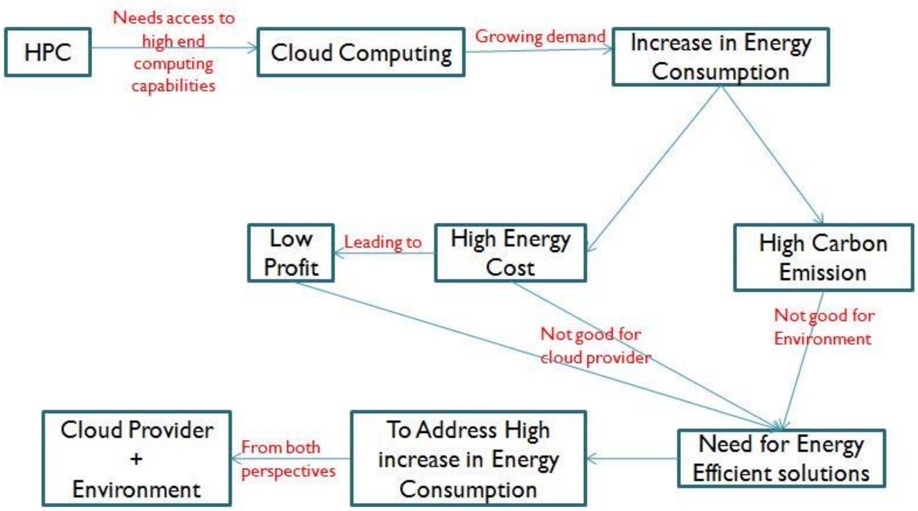

Fig. 1: Green Computing in Clouds

4. Existing Load Balancing Techniques in Cloud Computing

Following load balancing techniques are currently prevalent in clouds:-

• *Decentralized content aware load balancing* - H. Mehta et al. [21] proposed a new content aware load balancing policy named as workload and client aware policy (WCAP). It uses a unique and special property (USP) to specify the unique and special property of the requests as well as computing nodes. USP helps the scheduler to decide the best suitable node for the processing the requests. This strategy is implemented in a decentralized manner with low overhead. By using the content information to narrow down the search, this technique improves the searching performance and hence overall performance of the system. It also helps in reducing the idle time of the computing nodes hence improving their utilization.

• *Server-based load balancing for Internet distributed services* - A. M. Nakai et al. [22] proposed a new server-based load balancing policy for web servers which are distributed all over the world. It helps in reducing the service response times by using a protocol that limits the redirection of requests to the closest remote servers without overloading them. A middleware is described to implement this protocol. It also uses a heuristic to help web servers to endure overloads.

• *Join-Idle-Queue* - Y. Lua et al. [23] proposed a Join-Idle-Queue load balancing algorithm for dynamically scalable web services. This algorithm provides large-scale load balancing with distributed dispatchers by, first load balancing idle processors across dispatchers for the availability of idle processors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. By removing the load balancing work from the critical path of request processing, it effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time.

• *A Lock-free multiprocessing solution for LB* - X. Liu et al. [24] proposed a lock-free multiprocessing load balancing solution that avoids the use of shared memory in contrast to other multiprocessing load balancing solutions which use shared memory and lock to maintain a user session. It is achieved by modifying Linux kernel. This solution helps in improving the overall performance of load balancer in a multi-core environment by running multiple load-balancing processes in one load balancer.

• *Scheduling strategy on load balancing of virtual machine resources* - J. Hu et al. [25] proposed a scheduling strategy on load balancing of VM resources that uses historical data and current state of the system. This strategy achieves the best load balancing and reduced dynamic migration by using a genetic algorithm. It helps in resolving the issue of load-imbalance and high cost of migration thus achieving better resource utilization.

• *Central load balancing policy for virtual machines* - A. Bhadani et al. [26] proposed a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load evenly in a distributed virtual machine/cloud

computing environment. This policy improves the overall performance of the system but does not consider the
systems that are fault-tolerant.

• *LBVS: Load Balancing strategy for Virtual Storage* - H. Liu et al. [27] proposed a load balancing virtual storage strategy (LBVS) that provides a large scale net data storage model and Storage as a Service model based on Cloud Storage. Storage virtualization is achieved using an architecture that is three-layered and load balancing is achieved using two load balancing modules. It helps in improving the efficiency of concurrent access by using replica balancing further reducing the response time and enhancing the capacity of disaster recovery. This strategy also helps in improving the use rate of storage resource, flexibility and robustness of the system.

• *A Task Scheduling Algorithm Based on Load Balancing* - Y. Fang et al. [28] discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain a high resource utilization. It achieves load balancing by first mapping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.

• *Honeybee Foraging Behavior* - M. Randles et al. [14] investigated a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. It achieves global load balancing through local server actions. Performance of the system is enhanced with increased system diversity but throughput is not increased with an increase in system size. It is best suited for the conditions where the diverse population of service types is required.

• *Biased Random Sampling* - M. Randles et al. [14] investigated a distributed and scalable load balancing approach that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system. The performance of the system is improved with high and similar population of resources thus resulting in an increased throughput by effectively utilizing the increased system resources. It is degraded with an increase in population diversity.

• *Active Clustering* - M. Randles et al. [14] investigated a self-aggregation load balancing technique that is a self-aggregation algorithm to optimize job assignments by connecting similar services using local re-wiring. The performance of the system is enhanced with high resources thereby increasing the throughput by using these resources effectively. It is degraded with an increase in system diversity.

• *ACCLB (Load Balancing mechanism based on ant colony and complex network theory)* - Z. Zhang et al. [15] proposed a load balancing mechanism based on ant colony and complex network theory in an open cloud computing federation. It uses small-world and scale-free characteristics of a complex network to achieve better load balancing. This technique overcomes heterogeneity, is adaptive to dynamic environments, is excellent in fault tolerance and has good scalability hence helps in improving the performance of the system.

• *Two-phase load balancing algorithm (OLB + LBMM)* - S.-C. Wang et al. [18] proposed a two- phase scheduling algorithm that combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms to utilize better executing efficiency and maintain the load balancing of the system. OLB scheduling algorithm, keeps every node in working state to achieve the goal of load balance and LBMM scheduling algorithm is utilized to minimize the execution time of each task on the node thereby minimizing the overall completion time. This combined approach hence helps in an efficient utilization of resources and enhances the work efficiency.

• *Event-driven* - V. Nae et al. [20] presented an event-driven load balancing algorithm for real-time Massively Multiplayer Online Games (MMOG). This algorithm after receiving capacity events as input, analyzes its components in context of the resources and the global state of the game session, thereby generating the game session load balancing actions. It is capable of scaling up and down a game session on multiple resources according to the variable user load but has occasional QoS breaches.

• *CARTON* - R. Stanojevic et al. [17] proposed a mechanism for cloud control named as CARTON that unifies the use of LB and DRL. LB (Load Balancing) is used to equally distribute the jobs to different servers so that the associated costs can be minimized and DRL (Distributed Rate Limiting) is used to make sure that the resources are distributed in a way to keep a fair resource allocation. DRL also adapts to server capacities for the dynamic workloads so that performance levels at all servers are equal. With very low computation and communication overhead, this algorithm is simple and easy to implement.

• *COMPARE AND BALANCE* - Y. Zhao et al. [16] addressed the problem of intra-cloud load balancing amongst physical hosts by adaptive live migration of virtual machines. A load balancing model is designed and implemented to reduce virtual machines' migration time

Table 1: Existing Load Balancing techniques

Techniques	***Environment***	***Description***	***Findings***
Decentralized content aware [21]	Distributed computing	1. Uses a unique and special property(USP) of requests and computing nodes to help scheduler to decide the best node for processing the requests 2. Uses the content information to narrow down the search	1. Improves the searching performance hence increasing overall performance 2. Reduces idle time of the nodes
LB for Internet distributed services [22]	Distributed web servers	1. Uses a protocol to limit redirection rates to avoid remote servers overloading 2. Uses a middleware to support this protocol 3. Uses a heuristic to tolerate abrupt load changes	1. Reduces service response times by redirecting requests to the closest servers without overloading them 2. Mean response time is 29% smaller than RR(Round Robin) and 31% smaller than SL(Smallest Latency)
Join-Idle-Queue [23]	Cloud data centers	1. First assigns idle processors to dispatchers for the availability of the idle processors at each dispatcher 2. Then assigns jobs to processors to reduce average queue length of jobs at each processor	1. Effectively reduces the system load 2. Incurs no communication overhead at job arrivals 3. Does not increase actual response times
Lock-free multiprocessing[24]	Multi-core	1. Runs multiple load-balancing processes in one load balancer	1. Improves overall performance of load balancer
Scheduling strategy on LB of VM resources [25]	Cloud Computing	1. Uses Genetic algorithm, historical data and current state of system to achieve best load balancing and to reduce dynamic migration	1. Solves the problems of load imbalance and high migration cost
Central LB policy for VMs [26]	Cloud Computing	1. Uses global state information to make load balancing decisions	1. Balances the load evenly to improve overall performance 2. Up to 20% improvement in performance 3. Does not consider fault tolerance
LBVS: LB strategy for Virtual Storage [27]	Cloud Storage	1. Uses Fair-Share Replication strategy to achieve Replica Load balancing module which in turn controls the access load balancing 2. Uses writing balancing algorithm to control data writing load balancing	1. Enhances flexibility and robustness 2. Provides large scale net data storage and storage as a service
Task Scheduling Based on LB [28]	Cloud Computing	1. First maps tasks to virtual machines and then virtual machines to host resources	1. Improves task response time 2. Improves resource utilization
Honeybee Foraging Behavior [14]	Large scale Cloud Systems	1. Achieves global load balancing through local serve actions	1. Performs well as system diversity increases 2. Does not increase throughput as system size increases
Biased Random Sampling [14]	Large scale Cloud Systems	1. Achieves load balancing across all system nodes using random sampling of the system domain	1. Performs better with high and similar population of resources 2. Degrades as population diversity increases
Active Clustering [14]	Large scale Cloud systems	1. Optimizes job assignment by connecting similar services by local re-wiring	1. Performs better with high resources 2. Utilizes the increased system resources to increase throughput 3. Degrades as system diversity increases
ACCLB (Ant Colony and Complex Network Theory) [15]	Open Cloud Computing Federation	1. Uses small-world and scale-free characteristics of complex network to achieve better load balancing	1. Overcomes heterogeneity 2. Adaptive to dynamic environments 3. Excellent in fault tolerance 4. Good scalability
Two-phase scheduling(OLB + LBMM) [18]	Three-level Cloud Computing Network	1. Uses OLB (Opportunistic Load Balancing) to keep each node busy and uses LBMM(Load Balance Min-Min) to achieve the minimum execution time of each task	1. Efficient utilization of resources 2. Enhances work efficiency

Event-driven [20]	Massively Multiplayer Online Games	1. Uses complete capacity event as input, analyzes its components and generates the game session load balancing actions	1. Capable of scaling up and down a game session on multiple resources according to the variable user load 2. Occasional QoS breaches as low as 0.66%
Carton(LB + DRL) [17]	Unifying framework for cloud control	1. Uses Load Balancing to minimize the associated cost and uses Distributed Rate Limiting for fair allocation of resources	1. Simple 2. Easy to implement 3.Very low computation and communication overhead
Compare and Balance [16]	Intra-Cloud	1. Based on sampling 2. Uses adaptive live migration of virtual machines	1. Balances load amongst servers 2. Reaches equilibrium fast 3. Assures migration of VMs from high-cost physical hosts to low-cost host 4. Assumption of having enough memory with each physical host
VectorDot [29]	Datacenters with integrated server and storage virtualization	1. Uses dot product to distinguish node based on the item requirement	1. Handles hierarchical and multidimensional resource constraints 2. Removes overloads on server, switch and storage

by shared storage, to balance load amongst servers according to their processor or IO usage, etc. and to keep virtual machines' zero-downtime in the process. A distributed load balancing algorithm COMPARE AND BALANCE is also proposed that is based on sampling and reaches an equilibrium very fast. This algorithm assures that the migration of VMs is always from high-cost physical hosts to low-cost host but assumes that each physical host has enough memory which is a weak assumption.

• *VectorDot* - A. Singh et al. [29] proposed a novel load balancing algorithm called VectorDot. It handles the hierarchical complexity of the data-center and multi-dimensionality of resource loads across servers, network switches, and storage in an agile data center that has integrated server and storage virtualization technologies. VectorDot uses dot product to distinguish nodes based on the item requirements and helps in removing overloads on servers, switches and storage nodes.

These existing techniques have been summarized in Table 1.

5. Metrics for Load Balancing in Clouds

The existing load balancing techniques in clouds, consider various parameters like performance, response time, scalability, throughput, resource utilization, fault tolerance, migration time and associated overhead. But, for an energy-efficient load balancing, metrics like energy consumption and carbon emission should also be considered.

• *Overhead Associated* - determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a load balancing technique can work efficiently.

• *Throughput* - is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.

• *Performance* – is used to check the efficiency of the system. It has to be improved at a reasonable cost e.g. reduce response time while keeping acceptable delays.
• *Resource Utilization* - is used to check the utilization of resources. It should be optimized for an efficient load balancing.

• *Scalability* - is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.

• *Response Time* - is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.

• *Fault Tolerance* - is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique.

• *Migration time* - is the time to migrate the jobs or resources from one node to other. It should be minimized in order to enhance the performance of the system.

• *Energy Consumption (EC)* - determines the energy consumption of all the resources in the system. Load balancing helps in avoiding overheating by balancing the workload across all the nodes of a Cloud, hence reducing energy consumption.

Table 2: Metrics considered by existing Load Balancing techniques

Techniq- -ues	*Perform- -ance*	*Response Time*	*Scalability*	*Over- head*	*Through -put*	*Resource utilization*	*Fault tolerance*	*Migration time*	*EC*	*CE*
T1 [21]	✓	✓	✓	✓	×	✓	×	×	×	×
T2 [22]	✓	✓	×	×	×	×	×	×	×	×
T3 [23]	✓	✓	×	✓	×	×	×	×	×	×
T4 [24]	✓	×	×	×	✓	×	×	×	×	×
T5 [25]	×	×	×	✓	×	✓	×	×	×	×
T6 [26]	✓	✓	×	×	✓	✓	×	×	×	×
T7 [27]	✓	✓	✓	×	×	×	✓	×	×	×
T8 [28]	✓	✓	×	×	×	✓	×	×	×	×
T9 [14]	✓	×	✓	×	✓	×	×	×	×	×
T10 [14]	✓	×	✓	×	✓	×	×	×	×	×
T11 [14]	✓	×	✓	×	✓	×	×	×	×	×
T12 [15]	✓	×	✓	×	×	✓	✓	×	×	×
T13 [18]	✓	×	×	×	×	✓	×	×	×	×
T14 [20]	×	×	✓	×	×	✓	×	×	×	×
T15 [17]	✓	×	×	✓	×	✓	×	×	×	×
T16 [16]	×	×	×	✓	×	✓	×	✓	×	×
T17 [29]	×	×	×	×	×	✓	×	×	×	×

• *Carbon Emission (CE)* - calculates the carbon emission of all the resources in the system. As energy consumption and carbon emission go hand in hand, the more the energy consumed, higher is the carbon footprint. So, for an energy-efficient load balancing solution, it should be reduced.

Based on the above metrics, the existing load balancing techniques have been compared in Table 2.

6. Conclusions

Cloud Computing has widely been adopted by the industry, though there are many existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation, Energy Management, etc. which have not been fully addressed. Central to these issues is the issue of load balancing, that is required to distribute the excess dynamic local workload evenly to all the nodes in the whole Cloud to achieve a high user satisfaction and resource utilization ratio. It also ensures that every computing resource is distributed efficiently and fairly. Existing Load Balancing techniques that have been studied, mainly focus on reducing overhead, service response time and improving performance etc., but none of the techniques has considered the energy consumption and carbon emission factors. Therefore, there is a need to develop an Energy-efficient load balancing technique that can improve the performance of cloud computing along with maximum resource utilization, in turn reducing energy consumption as well as carbon emission to an extent that will help achieve Green Computing.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", EECS Department, University of California, Berkeley, Technical Report No., UCB/EECS-2009-28, pages 1-23, February 2009.

[2] R. W. Lucky, "Cloud computing", IEEE Journal of Spectrum, Vol. 46, No. 5, May 2009, pages 27-45.

[3] M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Journal of Internet Computing, Vol. 13, No. 5, September/October 2009, pages 10-13.

[4] G. Pallis, "Cloud Computing: The New Frontier of Internet Computing", IEEE Journal of Internet Computing, Vol. 14, No. 5, September/October 2010, pages 70-73.

[5] B. P. Rima, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, August 2009, pages 44-51.

[6] R. Mata-Toledo, and P. Gupta, "Green data center: how green can we perform", Journal of Technology Research, Academic and Business Research Institute, Vol. 2, No. 1, May 2010, pages 1-8.

[7] S. K. Garg, C. S. Yeob, A. Anandasivamc, and R. Buyya, "Environment-conscious scheduling of HPC applications on

distributed Cloud-oriented data centers", Journal of Parallel and Distributed Computing, Elsevier, Vol. 70, No. 6, May 2010, pages 1-18.
[8] K. M. Nagothu, B. Kelley, J. Prevost, and M. Jamshidi, "Ultra low energy cloud computing using adaptive load prediction", Proceedings of IEEE World Automation Congress(WAC) , Kobe, September 2010, pages 1-7.
[9] B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy, Survey, and Issues of Cloud Computing Ecosystems, Cloud Computing: Principles, Systems and Applications", Computer Communications and Networks, Chapter 2 , pages 21-46, DOI 10.1007/978-1-84996-241-42, Springer – V erlagLondonLimited, 2010.
[10] S. Kabiraj, V. Topka, and R. C. Walke, "Going Green: A Holistic Approach to Transform Business", International Journal of Managing Information Technology (IJMIT), Vol. 2, No. 3, August 2010, pages 22-31.
[11] K. T. Rao, P. S. Kiran, and L. S. S. Reddy, "Energy Efficiency in Datacenters through Virtualization: A Case Study", Global Journal of Computer Science and Technology, Vol. 10, No. 3, April 2010, pages 2-6.
[12] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport", Proceedings of the IEEE, Vol. 99, No. 1, January 2011, pages 149-167.
[13] A. M. Alakeel, "A Guide to dynamic Load balancing in Distributed Computer Systems", International Journal of Computer Science and Network Security (IJCSNS), Vol. 10, No. 6, June 2010, pages 153-160.
[14] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, April 2010, pages 551-556.
[15] Z. Zhang, and X. Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pages 240-243.
[16] Y. Zhao, and W. Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Republic of Korea, August 2009, pages 170-175.
[17] R. Stanojevic, and R. Shorten, "Load balancing vs. distributed rate limiting: a unifying framework for cloud control", Proceedings of IEEE ICC, Dresden, Germany, August 2009, pages 1-6,.
[18] S. Wang, K. Yan, W. Liao, and S. Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Chengdu, China, September 2010, pages 108-113.
[19] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges", Journal of Internet Services and Applications, Vol. 1, No. 1, April 2010, pages 7-18.
[20] V. Nae, R. Prodan, and T. Fahringer, "Cost-Efficient Hosting and Load Balancing of Massively Multiplayer Online Games", Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (Grid), IEEE Computer Society, October 2010, pages 9-17.
[21] H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), February 2011, pages 370-375.
[22] A. M. Nakai, E. Madeira, and L. E. Buzato, "Load Balancing for Internet Distributed Services Using Limited Redirection Rates", 5th IEEE Latin-American Symposium on Dependable Computing (LADC), 2011, pages 156-165.
[23] Y. Lua, Q. Xiea, G. Kliotb, A. Gellerb, J. R. Larusb, and A. Greenber, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services", An international Journal on Performance evaluation, In Press, Accepted Manuscript, Available online 3 August 2011.
[24] Xi. Liu, Lei. Pan, Chong-Jun. Wang, and Jun-Yuan. Xie, "A Lock-Free Solution for Load Balancing in Multi-Core Environment", 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA), 2011, pages 1-4.
[25] J. Hu, J. Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2010, pages 89-96.
[26] A. Bhadani, and S. Chaudhary, "Performance evaluation of web servers using central load balancing policy over virtual machines on cloud", Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE), January 2010.
[27] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, "LBVS: A Load Balancing Strategy for Virtual Storage", International Conference on Service Sciences (ICSS), IEEE, 2010, pages 257-262.
[28] Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.
[29] A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: integration and load balancing in data centers", Proceedings of the ACM/IEEE conference on Supercomputing (SC), November 2008.

Nidhi Jain Kansal received her Bachelor's Degree (Electronics) from Kurukshetra University, Haryana, India in 1997, and completed her MCA in 2000 from Guru Jambheshwar University, Hisar, Haryana, India. At present, she is a Ph.D. Candidate in Computer Science & Engineering Department at Thapar University. Her research interests lie in Cloud Computing, Green Computing, Load balancing and Resource utilization techniques.

Dr. Inderveer Chana completed her B.Tech in Computer Science (1997) and M.E. in Software Engineering from TIET (2002) and Ph.D. in Resource Management in Grid Computing from Thapar University, Patiala (2009) and has over thirteen years of teaching and research experience. She is working as Associate Professor in Computer Science and Engineering Department of Thapar University, Patiala. Her research interests include Grid computing, Cloud Computing and resource

management challenges in Grids and Clouds. She has over 30 publications in International Journals and Conferences of repute. More than 20 Masters have been completed so far under her supervision and is currently supervising 9 Doctoral candidates in the area of Grid and Cloud Computing.