International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Load Balancing Techniques in Cloud Computing: An Overview

Sheetanshu Rajoriya

Research Scholar, Department of Computer Science and Applications, SunRise University, Alwar, Rajasthan, India

Abstract: ***When you store your photos online instead of on your home computer, or use web-mail or a social networking site, you are using a "cloud computing" service. 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. This study discusses the concept of load balancing technique in cloud computing, the existing load balancing techniques and also discusses the different qualitative metrics or parameters like performance, scalability, associated overhead etc.***

Keywords: Cloud Computing; Parallel and Distributed Computing; Load Balancing; Static Load Balancing; Dynamic Load Balancing; ACCLB

1. Introduction

Cloud computing means that instead of all the computer hardware and software we are using sitting on our desktop, or somewhere inside our company's network, it's provided for us as a service by another company and accessed over the Internet, usually in a completely seamless way. Exactly where the hardware and software is located and how it all works doesn't matter to us, the user – it's just somewhere up in the nebulous "cloud" that the Internet represents.

As described by Sidhu and Kinger [1], cloud computing is an emerging computing paradigm. It aims to share data, calculations and service transparently over a scalable network of nodes. Since Cloud computing stores the data and disseminated resources in the open environment. So, the amount of data storage increases quickly.

Generally, there is no standard definition of cloud computing. Although, it consists of a bunch of distributed servers known as masters, providing demanded services and resources to different clients known as clients in a network with scalability and reliability of datacenter. The distributed computers provided on-demand services. Services may be of software resources (e.g. Software as a Service, SaaS) or physical resources (e.g. Platform as a Service, PaaS) or hardware/infrastructure (e.g. Hardware as a Service, HaaS or Infrastructure as a Service, IaaS).

Consider a scenario where we only have one web server in operation to handle all incoming requests to your company website. When the business is being established, it may be possible to handle the volume of traffic your site receives with one web server [2]. However, as the business grows, the one server will no longer be sufficient. If we don't add new web server at that instances then our web pages load slowly and users will have waiting till the server is free to process client requests.

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 [3], making sure that no single node is overwhelmed, hence improving the overall performance of the system. Kansal and Chana [4] stated that 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.

2. Concept of Cloud Computing

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. End users access cloud-based applications through a web browser or a light-weight desktop or mobile app while the business software and user's data are stored on servers at a remote location.

Cloud computing is one of the newest technology. Today lots of business organizations and educational institutions use cloud environment [5]. Cloud computing is the most recent topic in IT industry due to its flexibility in using the computing system. It provides everything as a service [6]. In general, cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. Doing so may give rise to certain privacy implications. In other words, cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing data storage, processing and bandwidth.

A simple example of cloud computing is Yahoo-mail, G-mail or Hotmail etc. All you need is just an internet connection and you can start sending e-mails. The server and e-mail management software is all on the cloud (internet) and is totally managed by the cloud service provider like Yahoo, Google etc. The consumer gets to use the software alone and enjoy the benefits. The analogy is, "if you need

milk, would you buy a cow?" All the users or consumers need is to get the benefits of using the software or hardware of the computer like sending emails etc. Just to get this benefit (milk) why should a consumer buy a (cow) software/hardware?

3. Parallel and Distributed Computing

The advents in the present micro-electronic technology have resulted in the availability of fast, inexpensive processors and advancement in the communication technology has resulted in the availability of cost-effective and highly efficient computer networks. The net result of the advancement in these two technologies is that the price/performance ratio has now changed to favour the use of interconnected multiple hosts instead of single high-speed processor. These interconnected multiple hosts either loosely or tightly coupled constitutes distributed and parallel computing environment respectively [7].

Today's need for parallel and distributed systems is speedily increasing because of increasing advance in scientific endeavor and the necessity of high-speed processing which may even tend toward the mode of distribution. In many systems there are number of servers but may be the probability of a processor being idle in the system and other processors having a queue of tasks at hand is very high. So there is necessity of the uniform distribution of workload among these servers [8].

An important problem here is to decide how to achieve a balance in the load distribution between processors so that the computation is completed in the shortest possible time. In parallel and distributed systems the concept of load balancing is used for distribution of workload among servers.

4. Load Balancing

Load balancing is the process of improving the performance of a parallel and distributed system through a redistribution of load among the processor [9]. The main goal of load balancing is to equalize the workload among the nodes by minimizing execution time, minimizing communication delays, maximizing resource utilization and maximizing throughput.

A basic example of load balancing in our daily life can be related to websites. Without load balancing, users could experience delays, timeouts and possible long system responses. Load balancing solutions usually apply redundant servers which help a better distribution of the communication traffic so that the website availability is conclusively settled [10].

In a distributed computer system environment, as described in [11], where two or more autonomous computers are connected via a communication network, resource sharing is a most desirable feature. Apart from sharing data and I/O devices, nodes of a distributed system could further improve system performance by sharing their computational power. Load balancing is a mechanism that enables jobs to move from one computer to another within the distributed system. This creates faster job service e.g., minimize job response time and enhances resource utilization. Various studies have shown that load balancing among nodes of a distributed system highly improves system performance and increases resource utilization.

A. Issues of Load Balancing and Scheduling

The load balancing mechanism in distributed systems has more issues as there is no centralized authority to allocate the work load among multiple processors. Some of the issues are described below:

- A good load balancing scheme needs to be general, stable, scalable, and to add a small overhead to the system. These requirements are interdependent [12].
- Load balancing is critical because processes may migrate from one node to another even in the middle of execution to ensure equal workload [9].
- An important problem is to decide how to achieve a balance in the load distribution between processors so that the computation is completed in the shortest possible time.
- Algorithms for load balancing have to rely on the assumption that the on hand information at each node is accurate to prevent processes from being continuously circulated about the system without any progress [9].
- Load sharing struggle to avoid the unshared state in processors which remain idle while tasks compete for service at some other processor [9].
- One of the crucial aspects of the scheduling problem is load balancing [13]. The challenge for a scheduling algorithm is that the requirements of fairness and data locality often conflict.
- Load balancing and task scheduling in distributed operating systems is a critical factor in overall system efficiency because the distributed system is non-uniform and non-preemptive, that is, the processors may be different [13].

B. Goals of Load Balancing

Some of the main goals of a load balancing algorithm as pointed out by [14] are:

- To achieve a greater overall improvement in system performance at a reasonable cost.
- To treat all jobs in the system equally regardless of their origin.
- To have a fault tolerance.
- To have the ability to modify itself in accordance with any changes or expand in the distributed system configuration.
- To maintain system stability.

Some other pros of load balancing are as follows:

- Ensures that connections are not directed to a server that is down.
- Good for scaling out for multiple clusters on different segments.
- Is highly configurable, with rules allowing for client affinity, weighting, filtering, availability etc.
- Works as a driver rather than as a service.

- Allows for mixed-version clusters.
- Manages resources efficiently.
- Utilizes all the systems resources as efficiently as possible.
- Improves the application response time by sending traffic round robin.
- If we have two members in load balance pool, with priority function we can send all the traffic to one node and keep other node as a backup.
- Helps with disaster recovery.
- OS and application patching is made easier by routing traffic to different during change windows (less customer downtime).

C. Load Balancing Algorithms

Load balancing on multi computers is a challenge due to the autonomy of the processors and the interprocessor communication overhead incurred in the collection of state information, communication delays, redistribution of load etc. Parallel and distributed computing environment is inherently best choice for solving/running distributed and parallel program applications. In such type of applications, a large process/task is divided and then distributed among multiple hosts for parallel computation. Livny and Melman [15] has pointed out that in a system of multiple hosts the probability of one of the hosts being idle while other host has multiple jobs queued up can be very high. Here load balancing is likely to improve performance. Such imbalances in system load suggest that performance can be improved by either transferring jobs from the currently heavily loaded hosts to the lightly loaded ones or distributing load evenly/fairly among the hosts .The algorithms known as load balancing algorithms, helps to achieve the above said goal(s).

The algorithm adopted for load balancing is closely related to the type and amount of load and job information assumed to be known to the decision-making modules. Load balancing algorithms can have three categories based on initiation of process as follows:

- **Sender Initiated:** In this type the load balancing algorithm is initialized by the sender. In this type of algorithm the sender sends request messages till it finds a receiver that can accept the load.
- **Receiver Initiated:** In this type the load balancing algorithm is initiated by the receiver. In this type of description algorithms the receiver sends request messages till it finds a sender that can get the load.
- **Symmetric:** It is the combination of both sender initiated and receiver initiated.

Depending on the current state of the system, load balancing algorithms can be divided into 2 categories as static and dynamic load balancing algorithms.

D. Static Load Balancing (SLB) Algorithms

Static load balancing policies are generally based on the information about the average behavior of system; transfer decisions are independent of the actual current system state. Static load balancing schemes use a priori knowledge of the applications and statistical information about the system.

In static load balancing, the performance of the processors is determined at the beginning of execution. Then depending upon their performance the work load is assigned by the master processor. The slave processors calculate their allocated work and submit their result to the master. A task is always executed on the processor to which it is assigned that is static load balancing methods are non pre-emptive. The goal of static load balancing method is to reduce the execution time, minimizing the communication delays [16].

A general disadvantage of static approaches is that the final selection of a host for process allocation is made when the process is created and cannot be changed during process execution to make changes in the system load.

E. Dynamic Load Balancing (DLB) Algorithms

In dynamic load balancing algorithms work load is distributed among the processors at runtime. The master assigns new processes to the slaves based on the new information collected [17, 18]. Unlike static algorithms, dynamic algorithms allocate processes dynamically when one of the processors becomes under loaded. Instead, they are buffered in the queue on the main host and allocated dynamically upon requests from remote hosts.

Dynamic load balancers continually monitor the load on all the processors, and when the load imbalance reaches some predefined level, the redistribution of work takes place. But as this monitoring steals CPU cycles so care must taken as when it should be invoked. This redistribution does incur extra overhead at execution time.

5. Comparative Parameters of Different Load Balancing Algorithms

The performance of various load balancing algorithms is measured by the following parameters:

F. Performance

Performance is used to check the efficiency of the system. This has to be improved at a reasonable cost, for example, reduce task response time while keeping acceptable delays.

G. Response Tiime

How much time a distributed system using a particular load balancing algorithm is taking to respond?

Static load balancing algorithms have shorter response time as one should not forget that in static load balancing there is lesser overhead as discussed earlier so emphasis is totally on executing jobs in shorter time rather than optimally utilizing the available resources.

Dynamic load balancing algorithms may have relatively higher response time as sometimes redistribution of processes takes place. Some time is being consumed during task migration.

H. Stabiliity

Stability can be characterized in terms of the delays in the transfer of information between processors and the gains in the load balancing algorithm by obtaining faster performance by a specified amount of time.

I. Overload Rejection

If Load balancing is not possible additional overload, rejection measures are needed. When the overload situation ends then first the overload rejection measures are stopped. After a short guard period load balancing is also closed down.

Static load balancing algorithms incurs lesser overhead as once tasks are assigned to processors, no redistribution of tasks takes place, so no relocation overhead. Dynamic load balancing algorithms incur more overhead relatively as relocation of tasks takes place.

J. Throughput

Throughput is the amount of data moved successfully from one place to another in a given time period.

K. Resource Utilization

Resource utilization includes automatic load balancing. A distributed system may have unexpected number of processes that demand more processing power. If the algorithm is capable to utilize resources, they can be moved to under loaded processors more efficiently.

Static load balancing algorithms have lesser resource utilization as static load balancing methods just tries to assign tasks to processors in order to achieve minimize response time ignoring the fact that may be using this task assignment can result into a situation in which some processors finish their work early and sit idle due to lack of work.

Dynamic load balancing algorithms have relatively better resource utilization as dynamic load balancing take care of the fact that load should be equally distributed to processors so that no processors should sit idle.

L. Fault Tolerant

This parameter gives that algorithm is able to tolerate tortuous faults or not. It enables an algorithm to continue operating properly in the event of some failure. If the performance of algorithm decreases, the decrease is proportional to the seriousness of the failure, even a small failure can cause total failure in load balancing.

M. Migration Time

It 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.

6. Distributed Load Balancing for the Clouds

In complex and large systems, there is a tremendous need for load balancing. For simplifying load balancing globally (e.g. in a cloud), one thing which can be done is, employing techniques would act at the components of the clouds in such a way that the load of the whole cloud is balanced.

In a distributed system, dynamic load balancing can be done in two different ways: distributed and non-distributed. In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. The interaction among nodes to achieve load balancing can take two forms: cooperative and non-cooperative [19]. Dynamic load balancing algorithms of distributed nature, usually generate more messages than the non-distributed ones because, each of the nodes in the system needs to interact with every other node. A benefit, of this is that even if one or more nodes in the system fail, it will not cause the total load balancing process to halt; it instead would affect the system performance to some extent. Distributed dynamic load balancing can introduce immense stress on a system in which each node needs to interchange status information with every other node in the system. In non-distributed type, either one node or a group of nodes do the task of load balancing. Non-distributed dynamic load balancing algorithms can take two forms: centralized and semi-distributed. In the first form, the load balancing algorithm is executed only by a single node in the whole system – the central node. This node is solely responsible for load balancing of the whole system. The other nodes interact only with the central node. In semi-distributed form, nodes of the system are partitioned into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in each cluster by appropriate election technique which takes care of load balancing within that cluster. Hence, the load balancing of the whole system is done via the central nodes of each cluster [19].

7. Why is Dynamic Load Balancing Required for a Distributed Cloud Environment?

Dynamic load balancing is a major key for a successful implementation of cloud environments. The main goal of cloud based architecture is to provide elasticity, the ability to expand and contract capacity on demand. Sometimes additional instances of an application will be required in order for the architecture to scale and meet demand. That means there is a need for a mechanism to balance requests between two or more instances of that application. The mechanism most likely to be successful in performing such a task is a load balancer [20].

There's no other way to assume increased load other than adding new instances and distributing that load with software or hardware. Similarly, when the additional instances of that application are deprovisioned, the changes to the network configuration need to be reversed, but software and hardware load balance is easy to scale up or scale down.

Obviously a manual process would be time consuming and inefficient, effectively erasing the benefits gained by introducing a cloud based architecture in the first place. The below is an example of how dynamic load balancing can be implemented [20].

- Let's assume that cloud management console, or a custom developed application or cloud tool kit, triggers an event that indicates a new instance is required to maintain availability. How it determines capacity limitations may be based on VM (Virtual Machines) status via VMware APIs (VMware is a virtualization and cloud computing software provider for x86compatible computers.) or data received from the load balancer, or a combination both.
- A new instance is launched in the cloud environment for same application. This is accomplished via the cloud management console or cloud tool kit.
- The cloud management console or tool kit grabs the IP address of the newly launched instance and instructs the load balancer to add it to the configuration as new resources for same application. This is accomplished by the standards based API which presents the configuration and management control plane of the load balancer to external consumers as services.
- The load balancer adds the new application instance to the appropriate configuration and as soon as it has confirmation that the instance is available and responding to requests, begins to direct traffic to that new instance without disturbing existing instances.

This process should be easily reversed upon termination of an instance, load balancer should be able to release termination instance IP. Note that, there may be other infrastructure components that are involved in this process that must also be considered on launch and decommission, but for this discussion we're just looking at the load balancing piece as it's critical to the concept of auto-scaling.

8. Load Balancing Algorithms for Distributed Systems

Here we will discuss three types of load balancing algorithms which can be applied to a distributed system [21]: honeybee foraging algorithm, a biased random sampling on a random walk procedure and active clustering.

N. Honeybee Foraging Algorithm

Randles et al. [22] 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.

This algorithm is derived from the behaviour of honey bees for finding and reaping food. There is a class of bees called the forager bees which forage for food sources, upon finding one, they come back to the beehive to advertise this using a dance called waggle dance. The display of this dance, gives the idea of the quality or quantity of food and also its distance from the beehive. Scout bees then follow the foragers to the location of food and then began to reap it. They then return to the beehive and do a waggle dance, which gives an idea of how much food, is left and hence results in more exploitation or abandonment of the food source [23].

The considered honeybee based load balancing technique uses a collection of servers arranged into virtual servers, each serving a virtual service queue of requests. Each server processing a request from its queue calculates a profit or reward, which is analogous to the quality that the bees show in their waggle dance. One measure of this reward can be the amount of time that the CPU spends on the processing of a request. The dance floor in case of honey bees is analogous to an advert board here. This board is also used to advertise the profit of the entire colony.

In load balancing operation [22], each server takes a particular bee role with probabilities px or pr. These values are used to mimic the honeybee colony whereby a certain number of bees are retained as foragers (to explore (px); rather than as harvesters) to exploit existing sources. A server successfully fulfilling a request will post on the advert board with probability pr. A server may randomly choose a virtual server's queue with probability px (exploring), otherwise checking for an advert (watching a waggle dance). In summary, idle servers (waiting bees) follow one of two behaviour patterns: a server that reads the advert board will follow the chosen advert, and then serve the request; thus mimicking harvest behaviour. A server not reading the advert board reverts to forage behaviour; servicing a random virtual server's queue request. An executing server will complete the request and calculate the profitability of the just-serviced virtual server.

O. Biased Random Sampling

Randles et al. [22] 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.

In this second load balancing approach, the load on a server is represented by its connectivity in a virtual graph. A full analysis of this mechanism is found in [24], with this section providing a brief overview. Here a virtual graph is constructed, with the connectivity of each node (a server is treated as a node) representing the load on the server. Each node is represented as a vertex in a directed graph and each in-degree represents free resources of that node.

Whenever a client sends a request to the load balancer, the load balancer allocates the job to the node which has at least one in-degree. Once a job is allocated to the node, the in-degree of that node is decremented by one. After the job is completed, the node creates an incoming edge and increments the in-degree by one. The addition and deletion of processes is done by the process of random sampling.

Each process is characterized by a parameter know as threshold value, which indicates the maximum walk length. A walk is defined as the traversal from one node to another until the destination is found. At each step on the walk, the neighbour node of current node is selected as the next node.

In this algorithm, upon receiving the request by the load balancer, it would select a node randomly and compares the current walk length with the threshold value. If the current walk length is equal to or greater than the threshold value, the job is executed at that node. Else, the walk length of the job is incremented and another neighbour node is selected randomly. The performance is degraded as the number of servers increase due to additional overhead for computing the walk length.

P. Active Clustering

Randles et al. [22] 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. Active clustering works on the principle of grouping similar nodes together and working on these groups. The process involved is:

- A node initiates the process and selects another node called the matchmaker node from its neighbours satisfying the criteria that it should be of a different type than the former one.
- The so called matchmaker node then forms a connection between a neighbour of it which is of the same type as the initial node.
- The matchmaker node then detaches the connection between itself and the initial node.

The above set of processes is followed iteratively.

9. Additional Algorithms

Here we describe some other load balancing algorithms used in distributed systems.

Q. Compare and Balance

This algorithm [25] uses the concept of compare and balance to reach an equilibrium condition and manage unbalanced system's load on the basis of probability (number of virtual machine running on the current host and whole cloud system). The current node selects randomly a node and compares the load with itself.

R. ACCLB (Load Balancing mechanism based on Ant Colony and Complex network theory)

Zhang and Zhang [26] 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.

S. Join--Idle Queue

Join-idle queue [27] is basically used for large-scale systems. It uses distributed dispatchers by first load balancing the idle processors across dispatchers and then assigning jobs to processors to reduce average queue length at each processor. The disadvantage of this algorithm is that it is not scalable. Lua et al. [28] proposed this load balancing algorithm for dynamically scalable web services. It effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time. It can perform close to optimal when used for web services. However, it cannot be used for today's dynamic-content web services due to the scalability and reliability.

10. Comparison of Existing Load Balancing Techniques Based on Various Metrics

Based on metrics discussed in section VI, the existing load balancing techniques have been compared in following table:

Metrics	*Honeybee Foraging*	*Biased Random Sampling*	*Active Clustering*	*Compare and Balance*	*ACCLB*	*Join-Idle Queue*
Performance	Yes	Yes	Yes	No	Yes	Yes
Response Time	No	No	No	No	No	Yes
Scalability	Yes	Yes	Yes	No	Yes	No
Overload	No	No	No	Yes	No	Yes
Throughput	Yes	Yes	Yes	No	No	No
Resource Utilization	No	No	No	Yes	Yes	No
Fault Tolerance	No	No	No	No	Yes	No
Migration Time	No	No	No	Yes	No	No

11. Conclusion

Load balancing is one of the main challenges in cloud computing. It is required to distribute the dynamic local workload evenly across all the nodes to achieve a high user satisfaction and resource utilization ratio by making sure that every computing resource is distributed efficiently and fairly. With proper load balancing, resource consumption can be kept to a minimum which will further reduce energy consumption and carbon emission rate which is a desire need of cloud computing.

This study explains the concept of load balancing, general idea about static and dynamic load balancing algorithms and explains various comparative parameters of different load balancing algorithms. This study also gives an overall description of various distributed load balancing algorithms that can be used in case of clouds. We have also compared these distributed load balancing algorithms and concluded that we can use a particular algorithm according to our requirement. But as we know that the cloud computing covers a very vast area, it is applicable to both small and large scale area but as we have concluded that none of the above algorithms satisfies the criteria. So there is a need to develop an adaptive algorithm which is suitable for heterogeneous environment and should also reduce the cost.

References

[1] Sidhu, A. K. and Kinger, S., "Analysis of Load Balancing Techniques in Cloud Computing", International Journal of Computers & Technology, 4, 2, March – April 2013.
[2] Kunamneni, V., "Dynamic Load Balancing for the Cloud", International Journal of Computer Science and Electrical Engineering (IJCSEE), 1, 1, 2012.
[3] Laha, J., Satpathy, R. and Dev, K., "Load Balancing Techniques : Major Challenges in Cloud Computing - A Systematic Review", International Journal of Computer Science and Network, 3, 1, 1-8, February, 2014.
[4] Kansal, N. J. and Chana, I., "Cloud Load Balancing Techniques: A Step Towards Green Computing", International Journal of Computer Science Issues (IJCSI), 9, 1, 238-246, January, 2012.
[5] Banerjee, S., Mandal, D., Adhikari, M. and Biswas, U., "Service Delivery Improvement for the Cloud Service Providers and Customers", International Journal of Computer Applications, 51, 5, 20-23, August, 2012.
[6] Kashyap, S. and Sharma, A. K., "Load Balancing Techniques in Cloud Computing Environment", International Conference on Electrical, Electronics & Computer Science Engineering, 94-97, 26th May, 2013.
[7] Chhabra, A., Singh, G., Waraich, S. S., Sidhu, B. and Kumar, G., "Qualitative Parametric Comparison of Load Balancing Algorithms in Parallel and Distributed Computing Environment", Proceedings of World Academy of Science, Engineering and Technology (PWASET), 16, November, 2006.
[8] Chopde, N. R. and Tijare, P. A., "Dynamic Load Balancing in Parallel Computing Based on Execution Time", International Journal of Advanced Research in Computer Science and Software Engineering, 2, 10, 372-375, October, 2012.
[9] Sharma, S., Singh, S. and Sharma, M., "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology, 38, 2008.
[10] Chaczko, Z., Mahadevan, V., Aslanzadeh, S. and Mcdermid, C., "Availability and Load Balancing in Cloud Computing", International Conference on Computer and Software Modeling, IACSIT Press, Singapore, 14, 134-140, 2011.
[11] Enslow P. H., "What is a "Distributed" Data Processing System ?", Computer, 11, 1, 13-21, January, 1978.
[12] Grosu, D. and Chronopoulos, A. T., "Noncooperative load balancing in distributed systems", Elsevier, Journal of Parallel and Distributed Computing, 65, 1022-1034, 2005.
[13] Nikravan, M. and Kashani, M. H., "A Genetic Algorithm for Process Scheduling in Distributed Operating Systems Considering Load Balancing", Proceedings 21st European Conference on Modelling and Simulation (ECMS), 2007.
[14] Madhu, K., M. and Shah, S., M., "Study on Dynamic Load Balancing in Distributed System", International Journal of Engineering Research & Technology (IJERT), 1, 9, November, 2012.
[15] Livny, M. and Melman, M., "Load balancing in homogeneous broadcast distributed systems", Proceedings of the Computer Network Performance Symposium, College Park, Maryland, United States, 47-55, 13th – 14th April, 1982,
[16] Rajguru, A. A. and Apte, S., S., "A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters", International Journal of Recent Technology and Engineering (IJRTE), 1, 3, 175-179, August, 2012.
[17] Malik, S., "Dynamic Load Balancing in a Network of Workstation", 95.515 Research Report, 19th November, 2000.
[18] Wang Y. and Morris, R., "Load balancing in distributed systems", IEEE Trans. Computing, C-34, 3, 204-217, March, 1985.
[19] Alakeel, A. M., "A Guide to Dynamic Load Balancing in Distributed Computer Systems", IJCSNS International Journal of Computer Science and Network Security, 10, 6, 153-160, June, 2010.
[20] Kunamneni, V., "Dynamic Load Balancing for the Cloud", International Journal of Computer Science and Electrical Engineering (IJCSEE), 1, 1, 2012.
[21] Randles, M., Odat, E., Lamb, D., Abu-Rahmeh, O. and Taleb-Bendiab, A., "A Comparative Experiment in Distributed Load Balancing", Second International Conference on Developments in eSystems Engineering, 2009.
[22] Randles, M., Lamb, D. and Taleb-Bendiab, A., "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", 24th International Conference on Advanced Information Networking and Applications Workshops, 551-556, 2010.

[23] Padhy, R. P., and Rao, P G. P., thesis entitled "Load balancing in cloud computing system", Department of Computer Science and Engineering, National Institute of Technology, Rourkela, Orissa, India, May, 2011.
[24] Foster, I., Yong, Z., Raicu, I. and Lu, S., "Cloud Computing and Grid Computing 360-Degree Compared", Grid Computing Environments Workshop, November, 2008.
[25] Zhao, Y. and Huang, W., "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud" Fifth International Joint Conference on INC, IMS and IDC, 6-9, 2009.
[26] Zhang, Z. and Zhang, X., "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), 240-243, May, 2010.
[27] Sran, N. and Kaur, N., "Comparative Analysis of Existing Load balancing techniques in cloud computing", International Journal of Engineering Science Invention, 2, 1, 2013.
[28] Lua Y., Xiea Q., Kliotb G., Gellerb A., Larusb J. R. and Greenber A., "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services", accepted in International Journal on Performance Evaluation, in press, 3rd August, 2011.

Authors Profile

Sheetanshu Rajoriya is the youngest author in the field of Computer Science and Applications. He was authored several books on this field. This name is synonym of mathematics as in his higher secondary examination, he scored cent percent marks in mathematics, for which he was awarded by many famous personalities. He then obtained his bachelor's degree in Computer Science from Dr. Hari Singh Gour Central University, Sagar, Madhya Pradesh in 2007 and attaining highest marks in the entire university. After that he completed MCA from the same university of repute in 2010 and again attaining highest marks in the entire university. He also obtained masters degree in mathematics. Now he is pursuing the PhD in Computer Applications from SunRise University, Alwar, Rajasthan. He has rich practical experience of 8 years in the field of Hardware & Networking and has been teaching to the students at various levels like diploma, PGDCA, BCA, MCA, etc.