Review of Load Balancing in Cloud Computing

Suriya Begum
Research Scholar.
Visvesvaraya Technical University
Belgaum, India

Dr. Prashanth C.S.R
Prof. and Head of Department
Dept. of Computer Science and Engineering
New Horizon College of Engineering
Bangalore, India

***Abstract*— With the exponential rise in the demands of the clients worldwide, a large scale distributed systems have been introduced as a computing environment. Cloud computing has paved a revolutionary path in this direction of distributed environment for accomplishing optimized performance, shortest response time, network resource utilization, and adaptability of service level agreement. Cloud computing has multiple benefits as well as it is also accompanied with certain serious technical loopholes. The proposed paper has focused on one such issue of load balancing. The consequences of inefficient load balancing may lead to detritions of an organization business performance on cloud environment. Hence this paper illustrates various aspects pertaining to domain of cloud computing, its evolution, its generic issues, and particularly to issues related to load balancing. Various techniques adopted in the past research work have been analyzed and the findings were illustrated in this paper.**

Keywords-; Cloud computing, Grid Computing, Load balancing

I. INTRODUCTION

Cloud computing can be illustrated as the operation of heavy computing resources that includes hardware and software package that are delivered to clients as a service over an outsized scale network [1]. The name „cloud" has been originated from the employment of a cloud-shaped representation as associate abstraction for the advanced infrastructure that is contained in a very system diagrams. The origination of the term „cloud computing" is very vague, however it appears to be derived from the practice of using schematic diagrams of computing and communications systems. The word cloud is employed as a allegory for the big scale network that is supported by the standardized use of a cloud-like form to denote a network on telephone schematics associated later to depict the web in network diagrams as an abstraction of the underlying infrastructure it represents. The cloud image [2] was used to represent the web as early as 1994. The elemental thought of cloud computing essentially dates back to the Nineteen Fifties once there was associate availableness of huge scale mainframe in establishments and companies. Owing to the pricey nature of mainframe, there arise a necessity to search out another answer for allowing the multiple users for accessing and sharing equivalent central processor time thereby truncating the chance of periods of inactivity (also termed as time-sharing) [3]. As computers became additionally prevailing, scientists and technologists explored ways to create large-scale computing power on the market to cater additional users through sharing, experimenting with algorithms to produce the best use of the infrastructure, platform and applications with prioritized access to the computer hardware and potency for the tip users [4]. The high value of those powerful computing systems has forced several prime organizations to require Associate in Nursing initiative to explore higher value effective answer exploitation sharing. The prime organization can embody IBM, GE, National CSS etc who took the initiative to launch and marketed time sharing. With the ascension of net technology and standards, varied merchandise and demands of distributed computing were on high increase. The presence of pervasive high computing network, value effective computing devices, storage devices together with massive scale use of virtualization of hardware, service headed design has paved the trail for top demands in new technology, therefore known as „cloud computing."

The domain of cloud computing is still surfaced by many issues which will be discussed in this paper in later section. The prime focus of the paper will be to analyze the research issues in load balancing protocols or understanding its requirement in cloud platform. Since, with the scene of rapid use of cloud computing resources, the demand along with provisioning of the cloud resources has to be effectively design to claim better SLA (Service Level Agreement) with zero downtime claims. Currently, there is a presence of multiple vendors offering cloud services (Amazon, Microsoft, IBM, Google, Salesforce, HP, Oracle, Citrix, EMC etc.) and there are growing numbers of clientele too. Hence, it is quite obvious that catering the massive and dynamic requirements of such exponentially growing clients will become one of the most challenging issues. And to mitigate this issue, an effective load balancing technique should be explored. The proposed paper will introduce a thorough analysis of the evolution of cloud platform right from the origination of the initial distributed computing system. The paper will mainly focus on the research issues of load balancing and will attempt to analyze the prior work done in this field. Section 2 will highlight about the evolution of cloud computing followed by Section 3 for Issues in cloud platform. Section 4 will discuss about the load balancing issues along with the review of the past research work focused on load balancing in cloud platform. Section 5 will discuss about the research issues finally followed by concluding remarks in Section 6

II. Evolution of Cloud Computing

A distributed computing system enables the sharing, selection, and aggregation of distributed heterogeneous computational and storage resources, which are under the control of different sites or domains. The Idea-phase of cloud computing started in 1960 followed by pre-Cloud phase in and around 1999-2006 and final cloud phase was seen on 2007. The evolution of cloud is discussed in this paper from cluster computing, grid computing and then cloud computing analyzing the merits and demerits of every constituent.

A. Cluster computing

The concept of clustering in computer system is the use of multiple computers, typically PCs or UNIX workstations, multiple storage devices, and redundant interconnections, to form what appears to users as a single highly available system [5]. Cluster computing can be used for load balancing as well as for high availability. Advocates of clustering suggest that the approach can help an enterprise achieve 99.99% availability in some cases. One of the main ideas of cluster computing is that, to the outside world, the cluster appears to be a single system. The standard cluster architecture as defined in the work of Buyya [5] is as shown below:

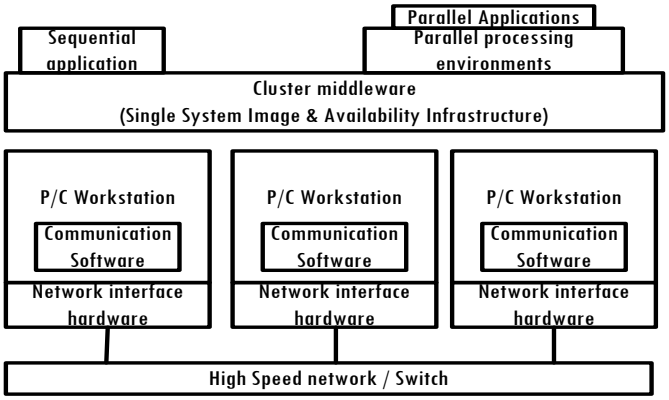

Figure 1 Cluster Computing Architecture

A frequent use of cluster computing is to load balance traffic on high-traffic internet sites. An internet page request is forwarded to a "manager" server, that then determines that of many identical or extremely similar internet servers to forward the request to for handling. Having an internet farm (as such a configuration is usually called) permits traffic to be handled additional quickly. clump has been offered since the Nineteen Eighties once it absolutely was employed in DEC's VMS systems [6]. Cluster computing may also be used as a comparatively inexpensive kind of multiprocessing for scientific and different applications that lend themselves to parallel operations. associate early and well-known example was the Beowulf project [7] within which variety of off-the-peg PCs were wont to kind a cluster for scientific applications

- *Fail-over Cluster Computing*: In this type of cluster computing, the machines‟ work is continuously monitored and when one of the two host stops working the other machine takes over. The aim is to ensure a continuous service.
- *Cluster with load balancing*: In this type of cluster computing, the work requests are sent to the machine with fewer loads.
- *HPC Cluster Computing*: In this type of cluster computing, the computers are configured to provide extremely high performance. The machines break down the processes of a job on multiple machines in order to gain in performance.

The main precincts of Cluster Computing are:

- Complicated to manage and organize a large number of computers
- Poor performance in the case of non-parallelizable applications.
- Physical space requirement is considerably greater than that of a single server
- Maximized energy consumption compared to a single server

B. Grid Computing

Grid computing permits virtual organizations to share geographically distributed resources as they follow universal goals, forward the absence of central location, central management, state, associate degreed an existing trust relationship [8]. Grid computing may be utilized in a spread of how to handle varied styles of application necessities. Often, grids square measure categorized by the sort of solutions that they best address. The 3 primary styles of grids square measure summarized below [9]. Of course, there aren't any hard boundaries between these grid varieties and infrequently grids could also be a mixture of 2 or a more of those. However, as one think about developing applications which will run in an exceedingly grid atmosphere, keep in mind that the sort of grid atmosphere that one can simply are exploitation can have an effect on several of one‟s selections.

- Computational grid: A computational grid is focused on setting aside resources specifically for computing power. In this type of grid, most of the machines are high-performance servers.
- Scavenging grid: A scavenging grid is most commonly used with large numbers of desktop machines. Machines are scavenged for available CPU cycles and other resources. Owners of the desktop machines are usually given control over when their resources are available to participate in the grid.
- Data grid: A data grid is responsible for housing and providing access to data across multiple organizations. Users are not concerned with where this data is located as long as they have access to the data. For example, you may have two universities doing life science research, each with unique data. A data grid would allow them to share

their data, manage the data, and manage security issues such as who has access to what data.

A Computational grid states that it is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. Whereas, [10] Desktop grid refers to a grid infrastructure that is confined to an institutional boundary, where the spare processing capacity of an enterprise‟s desktop PCs are used to support the execution of the enterprise‟s applications.

Table 1: Key components of Computational and Desktop Grids

KEY COMPONENTS	
Computational Grids	Desktop Grids
Infrastructure	Physical node Management
Dependable service	Resource Scheduling
Consistency of service	Job scheduling
Pervasive access	

Computational and Desktop Grids can be classified on the basis on Infrastructure, Models and Software applications and platforms etc. some of them are given in Table 2.

Table 2: Classifications of Computational and Desktop Grids

CLASSIFICATIONS	
Computational Grids	Desktop Grids
gLite	Distributed.net
NorduGrid/ARC	Entropia
UNICORE	SETI@home
MiG	Bayanihan
WebCom-G	Condor
Office Grid	BOINC

However, the concept and utilization of grid computing is also explored with multiple limitations:

- For the application that cannot properly use Message Passing Interface (MPI), the user may by bound to work on large symmetric multiprocessor (SMP).
- The network connection requirement is quite higher (minimum GB Ethernet)
- Some specific application running on grid environment may need to be tweaked in order to completely utilize the new model.
- Licensing in grid across multiple servers will make it eventually resistive operationally for certain application running on grid environment.
- Although Grid environments offer the privilege of many smaller servers across various administrative domain, but it becomes tremendously challenging task to manage efficient tools for change and controlling server configuration in proper synchronization.
- Standard and benchmarked resource sharing policy with better SLA and resource provisioning is still missing

C. Cloud Computing

A Cloud is a form of parallel and distributed system possessing a group of inter-connected and virtualized computers that are dynamically scheduled and highlighted as one or more unified computing resources based on service-level agreements established through conciliation between the service provider and consumers. [11]. The cloud service model is shown below.

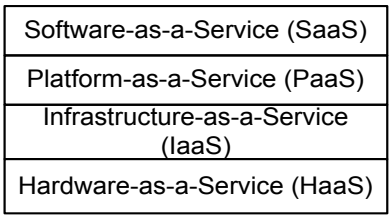

Figure 2 Cloud Service Models

Cloud schematic design is that the blueprint of the software package applications that uses internet-accessible on demand service. Cloud architectures are underlying on infrastructure that is used only when it is required that draw the necessary resources on-demand and perform a specific job, then relinquish the unneeded resources and often dispose them after the job is done. SaaS is a multi-tenant platform that uses common resources and a single instance of both the object code of an application as well as the underlying database to support multiple customers simultaneously. The prime global players of SaaS are SalesForce, IBM, Microsoft, and Oracle. PaaS provides developers with platform including all the systems and environments comprising the end-to-end life cycle of developing, testing, deploying, and hosting of complex web applications. Microsoft Azure is one example of PaaS. Iaas Is the delivery of computer infrastructure as a service. The usage based payment factor is one key benefit of PaaS. GoGrid, Flexiscale, Rackspace etc are to name a few examples for PaaS.

Table 3 Cloud Deployment models

CLOUD COMPUTING DEPLOYMENT MODELS	
Private Cloud	The cloud infrastructure is owned or leased by a single organization and is operated solely for that organization.
Community Cloud	Several organizations that have similar polices, objectives, aims and concerns share the cloud infrastructure.
Public Cloud	A large organization owns the cloud infrastructure and sells cloud services to industries or public.
Hybrid	It is combination of two or more clouds. It

Cloud	enables data and application probability.

Each deployment model is either internal or external. Internal clouds are covered under organizations security policy but external clouds are not. Cloud computing offers lots of advantages [12]:

- *Cost*- As in the clouds the user need not own the resources, it just need to pay as per the usage in terms of time, storage and services. This feature reduces the cost of owning the infrastructure [13], [14].
- *Performance*-the performance is improved because the cloud is not a single computer but a large network of powerful computers resulting in high processing power [15], [11], [16].
- *Freedom from up gradation and maintenance*- the cloud infrastructure is maintained and upgraded by the cloud service provider [14], [11].
- *Scalability*- The user is can request to increase the resources if the area of application grows or new functionality is added. On the other hand if requirement shrinks the user can request to reduce the resources as well [11], [15].
- *Speedy Implementation*- Time of Implementation of cloud for an application may be in days or sometimes in hours. You just need a valid credit card and need to fulfill some online registration formalities [16].
- *Green*- The cloud computing is a green technology since it enable resource sharing among users thus not requiring large data centers that consumes a lot of power [16].
- *Mobility*- We don"t need to carry our personal computer, because we can access our documents anytime anywhere [11].
- *Maximized Storage Capacity*- In Cloud computing we have extreme resources for storing data because our storage consists of many bases in the Cloud. Another thing about storing data in the Cloud is that, because of our data in the Cloud can automatically duplicated, they will be more safety [11].

Along with list of advantages, various constraints found in the usage of cloud computing are as follows:

- When the applications, processes and data are tightly coupled or interdependent.
- When there are not well defined points to share the data, process and behavior within an application.
- When the application require a very high level of security.
- When you want total control on your processes and data and thus cannot outsource your application or its critical components.
- When the core internal architecture of the organization is not functioning well, then first make it strong so that it can be easily mapped to cloud architecture.
- When one need native APIs, since the cloud does not provide native APIs.
- When someone is already using a legacy system, since older systems posses number of difficulties to move to cloud architecture.

Table 4 Evidence of Cloud Service issues

Service Issue	Duration	Date
Microsoft Azure: Malfunction in Windows Azure [17]	22 Hrs	March, 2008
Gmail and Google Apps Engine [18]	2.5 Hrs	July, 2009
Google search outage: Programming error [19]	40 min	Jan, 2009
Gmail: Site unavailable due to outage in contacts system [20]	1.5 Hrs	Aug, 2008
Google AppEngine partial outage programming Error [21]	5 Hrs	June, 2008
S3 outage: authentication service overload leading to unavailability [22]	2 Hrs	Feb, 2008
FlexiScale: core network failure [23]	18 Hrs	Oct, 2008
Salesforce.com: zero connectivity due to server disruption stopping all data from Japan, Europe, & North America [24]	1-2 Hrs	Jan 2009
Microsoft: zero connectivity for design issue [25]	2 Hrs	Sep-2010
Skype: network outage due to unstable and overloaded clusters [26]	1 day	Dec-2010
Amazon Web Service: zero connectivity due to incorrect traffic shift [27]	4 days	April-2011
Amazon EC2 Cloud: Transformer exploded, caught fire in datacenter and caused connection outage [28]	2 days	Aug, 2011
Amazon: Connectivity Issues in EC2 cloud computing service [29]	3 Hrs	June, 2012

The above table represents only a few evidence of the connection outage issue from some of the reputed Cloud service providers.

III. ISSUE IN CLOUD PLATFORM

Apart from the issues specified in the previous section, the other set of the technical issues in cloud computing will include load balancing, security, reliability, ownership, data back-up, data portability, multiplatform support, and intellectual property. Here is a rundown on most of the current issues concerning cloud computing:

- *Load balancing* [30]: Load leveling is usually mechanized to implement failover-the continuance of a service when the failure of 1 or additional of its parts. The components are monitored frequently and once one becomes non-responsive, the load balancer is up on and now not sends traffic to it. This can be an inherent feature from grid-based computing for cloud-based platforms. Energy conservation and resource consumption don't seem to be continuously attentiveness once discussing cloud computing; but with correct load leveling in place of resource consumption are often unbroken to a minimum. This is not only serves to keep cost low and enterprise „greener", it also puts less stress on the circuits of each individual design making them more potentially last longer.
- *Security* [31]: While a leading edge cloud services provider will employ data storage and transmission encryption, user authentication, and authorization (data access) practices, many people worry about the vulnerability of remote data to such criminals as hackers, thieves, and disgruntled employees. Cloud providers are enormously sensitive to this issue and apply substantial resources to mitigating concern.
- *Reliability* [32]: Some people worry also about whether a cloud service provider is financially stable and whether their data storage system is trustworthy. Most cloud providers attempt to mollify this concern by using redundant storage techniques, but it is still possible that a service could crash or go out of business, leaving users with limited or no access to their data. A diversification of providers can help alleviate this concern, albeit at a higher cost.
- *Ownership* [33]: Once data has been relegated to the cloud, some people worry that they could lose some or all of their rights or be unable to protect the rights of their customers. Many cloud providers are addressing this issue with well-crafted user-sided agreements. That said, users would be wise to seek advice from their favorite legal representative. Never use a provider who, in their terms of service, lays any kind of ownership claim over your data.
- *Data Backup* [34]: Cloud providers employ redundant servers and routine data backup processes, but some people worry about being able to control their own backups. Many providers are now offering data dumps onto media or allowing users to back up data through regular downloads.
- *Data Portability and Conversion* [35]: Some people are concerned that, should they wish to switch providers, they may have difficulty transferring data. Porting and converting data is highly dependent on the nature of the cloud provider"s data retrieval format, particular in cases where the format cannot be easily discovered. As service competition grows and open standards become established, the data portability issue will ease, and conversion processes will become available supporting the more popular cloud providers. Worst case, a cloud subscriber will have to pay for some custom data conversion.
- *Multiplatform Support* [36]: More an issue for IT departments using managed services is how the cloud-based service integrates across different platforms and operating systems, e.g. OS X, Windows, Linux and thin-clients. Usually, some customized adaption of the service takes care of any problem. Multiplatform support requirements will ease as more user interfaces become web-based.
- *Intellectual Property* [37]: A company invents something new and it uses cloud services as part of the invention. Is the invention still patentable? Does the cloud provider have any claim on the invention? Can they provide similar services to competitors? All good questions and answerable on a case-by-case basis.

Once someone understands that cloud computing potentially suffers from much of the same fate as proprietary systems, the question becomes "do the advantages of using the cloud outweigh my concerns?" For low-risk operations and for insensitive information, the answer can easily be "yes." Realize that cloud-based services can be backed-up, verified, double-checked, and made more secure by combining them with traditional non-cloud IT processes.

IV. LOAD BALANCING IN CLOUD PLATFORM

Load balancing [38] is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded. A load balancing protocol is dynamic in nature doesn't contemplate the previous state or behavior of the system, that is, it depends on the current behavior of the system. It is common these days in redundant high-availability computer systems that incoming network traffic is distributed on network level by deploying one of the frequently used network load balancing algorithms like:- random-allocation, round-robin allocation, weighted round-robin allocation, etc). These algorithms use solely network parameters of incoming traffic to create selections wherever to forward traffic, with none data from different elements of database system, like current load of application or info servers. Since these days it is extremely common to possess internet servers acting as application servers, it is usual that load balancers use session-switching technique, which suggests that once a user opens website on one server, it will stay on it server whereas the session lasts.

Depending on who initiated the process, load balancing algorithms can be of five categories:

- Sender Initiated: If the load balancing algorithm is initialized by the sender
- Receiver Initiated: If the load balancing algorithm is initiated by the receiver
- Symmetric: It is the combination of both sender initiated and receiver initiated
- Static: It doesn"t depend on the current state of the system. Prior knowledge of the system is needed.
- Dynamic: Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach.

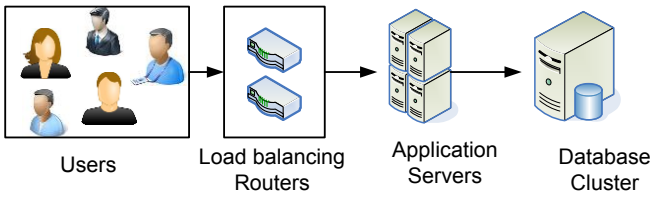

Figure 3 Schematics of typical high-availability computer system with hardware load balancers.

Table 5 Metrics in existing LB techniques in cloud computing

LOAD BALANCING METRICS	
Metric	Illustration
Throughput	It is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system
Overhead	It 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.
Fault Tolerance	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.
Response Time	It is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.
Resource Utilization	It is used to check the utilization of re-sources. It should be optimized for an efficient load balancing.
Scalability	It is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.
Performance	It is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays

Central to many other issues lies the establishment of an effective load balancing algorithm. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. This technique can be sender initiated, receiver initiated or symmetric type (combination of sender initiated and receiver initiated types).

V. EXISTING TECHNIQUES IN LOAD BALANCING

This review aims at summarizing the current state of the art of existing load balancing techniques in cloud computing. Here inspite of quantity of work done, the focus in given to only names of distinctive techniques used to mitigate load balancing issue in cloud computing), (load balancing techniques in cloud computing), (load balancing in clouds) and (load balancing in datacenters). Only papers written in English were included. Following load balancing techniques are currently prevalent in clouds:

A. Technique-1: Event-Driven

V. Nae et al. [39] bestowed event-driven load balancing algorithmic for real time Massively Multiplayer on-line Games (MMOG) that is characterized by a new type of large-scale distributed applications with a real-time virtual world entertaining massive volumes of players in the network. In order to cater up the variable computational and latency-aware resource demands, the MMOG machinist over-provision an own multi-server infrastructure with adequate competence for assuring the Quality of Service (QoS) requirements and a smooth game play at all times. This statically scheduled infrastructure has two prime challenges: it has maximum functional costs and is susceptible to capacity shortages in case of unanticipated increases in demand. In contrast to uniform provisioning, the new cloud computing technology based on resource virtualization has the potential to provide an on-demand infrastructure for MMOGs, where resources are provisioned and paid for only when they are actually needed. Conversely, this technology can introduce virtualization overheads which may cancel out the benefits. In prior work, the authors have studied the outcomes of deploying virtualized resources regarding the technology-incurred overheads and their impact on the QoS offered to the clients and the cost-effective results, while considering ideal resources in terms of accessibility. It consists of three modules with unique and distinct roles, e.g.1) *Client*: it is the module that connects to

the game operator‟s sessions. It can join the sessions offered by game machinist on the basis of its subscriptions. A subscription represents a contract between a client and a game machinist based on which the client is allowed, under certain terms and with certain QoS guarantees, to join a session managed by the game machinist. 2) *Machinist*: This module provisions resources from the resource provider and ensures the proper execution of the sessions. The game machinists interact with clients and offer them a selection of games, usually by contracting new games from game development companies. The machinists execute distributed sessions with guaranteed QoS comprising interconnected application servers. 3) *Resource Provider*: This module offers the physical or virtualized machines on which the game servers will run. The resource providers lease virtual machines with fuzzy definitions of their characteristics, but with much more precise guaranties in terms of resource availability. This algorithmic once receiving capability events as input, analyzes its elements in context of the resources and also the global state of the game session, thereby generating the game session load reconciliation actions. It is capable of scaling up and down a game session on multiple resources in keeping with the variable user load however it also has occasional QoS breaches. The findings of the work are i) it is capable of scaling a game session on multiple resources in keeping with the variable user load and ii) it conjointly provides rise to occasional QoS breaches

B. Technique-2: VectorDot

A. Singh et al. [40] planned a unique load balancing rule referred to as VectorDot. In this work, the author has described a novel VectorDot algorithm that takes into account challenging hierarchical and multi-dimensional constraints while load balancing a system. The VectorDot algorithm is inspired by the Toyoda method for multi-dimensional knapsacks [41] and has several interesting aspects. While the original Toyoda heuristic was limited to selecting items into a single knapsack (item-comparison only), here we extend it to a collection of knapsacks (node-comparisons also) and dynamic load balancing among them in hierarchical SAN topologies. It was also seen that another use of Toyoda heuristic for resource management in storage systems, however it neither addresses the hierarchy, nor has been evaluated for performance and quality. It handles the hierarchal quality of the data-center and multidimensionality of resource hundreds across servers, network switches, associate degreed storage in an agile knowledge center that has integrated server and storage virtualization technologies. VectorDot mechanizes dot product to differentiate nodes supported by the item needs and assists in removing overloads on servers, switches and storage nodes. The findings of the work are i) it handles hierarchal and third-dimensional resource constraints and ii) data size need more validation testing for proving reliability of the model.

C. Technique-3: LBVS

H. Liu et al. [42] projected a load balancing on virtual memory strategy (LBVS) that gives an outsized scale web information storage model and Storage as a Service model supported Cloud Storage. This work basically highlights a load balancing virtual storage strategy and use Fair-Share Replication (FSR) and a executing a balancing algorithm to archive it. Since bandwidth and CPU speed are usually expensive to change, the another alternative way is by placing replicas of data objects closer to clients is the cheapest way. The main idea of FSR is to identify best candidate nodes for replica placement primarily based on access load. Storage virtualization is achieved using a schema of three-layered and load balancing is achieved using 2 load balancing modules. It helps in rising the potency of coincident access by using duplicate balancing any reducing the interval and enhancing the capability of disaster recovery. This strategy additionally helps in raising the utilization rate of storage resource, flexibility and strength of the system. The findings of the work are i) it improves task interval and ii) the process components & cumulative data center load are not considered throughout the work.

D. Technique-4: Server-based LB for Internet distributed services

A. M. Nakai et al. [43] projected a novel server-based load balancing policy for internet servers that are distributed everywhere the world. The main contributions of this work are: (i) the evaluation of client-based server selection schemes in scenarios where several clients use the same schemes; and (ii) the proposal of a new solution that outperforms existing ones by dynamically adapting the fraction of load each client submits to each server. In order to evaluate the solution, the author have implemented in a discrete event simulator framework using Java. The author has used the PackMime Internet traffic model [44] to generate HTTP traffic in the simulations. PackMime allows the generation of both HTTP/1.0 and HTTP/1.1 traffic. The intensity of the traffic is controlled by the rate parameter, which is the average number of new connections started per second. The implementation provides a set of random variable generators that drive the traffic generation. Each random variable follows a specific distribution. In this paper, we argue that if clients collaborate in order to balance server load they can obtain better response times. The solution adaptively changes the fraction of load each client sends to each server giving higher priorities to nearby servers. Although this less greedy strategy of sending fractions of the load to worsen servers seems to be counterintuitive, our experiments have shown that our solution overcomes the two types of policies proposed so far, even an in scenario that favors one type or another. It assists in reducing the service response times by employing a protocol that limits the redirection of requests to the closest remote servers while not overloading them. A middleware is represented to implement this protocol. It conjointly uses a heuristic to assist net servers to endure overloads. The finding of the work can embody the actual fact that reliability of the simulation result doesn‟t return up with real time situation.

E. Technique-5: Fuzzy Logic

Srinivas Sethi et. al. [45] has introduced the novel load equalization technique using fuzzy logic in cloud computing, within which load equalization could be a core and difficult issue in Cloud Computing. In this work, the authors have designed a new load balancing algorithm based on round robin in Virtual Machine (VM) environment of cloud computing in order to achieve better response time and processing time. The load balancing algorithm is done before it reaches the processing servers the job is scheduled based on various parameters like processor speed and assigned load of Virtual Machine (VM) and etc. It maintains the information in each VM and numbers of request currently allocated to VM of the system. It identify the least loaded machine, when a request come to allocate and it identified the first one if there are more than one least loaded machine. The authors have tried to implement the new load balancing technique based on Fuzzy logic. Where the fuzzy logic is natural like language through which one can formulate their problem. In this architecture, the fuzzifier performs the fuzzification process that converts two types of input data like processor speed and assigned load of Virtual Machine (VM) and one output like balanced load which are needed in the inference system. The design also considers the processor speed and load in virtual machine as two input parameters to make the better value to balance the load in cloud using fuzzy logic. These parameters are taking as inputs to the fuzzifier, which are used to measure the balanced load as the output. Two parameters named as the processor speed and assigned load of Virtual Machine (VM) of the system are jointly used to evaluate the balanced load on data centers of cloud computing environment through fuzzy logic. The results obtained with performance evaluation can balance the load with decreases the processing time as well as improvement of overall response time, which are leads to maximum use of resources. The processor speed and allotted load of Virtual Machine (VM) are deployed to balance the load in cloud computing through fuzzy logic. The finding of the work can embrace the very fact that solely Round Robin is taken into account that doesn't cater the important real time demand of the many large files over network.

F. Technique-6: Task Scheduling

Tayal [46] has proposed an optimized protocol based on Fuzzy-GA improvement that makes a programming call by evaluating the whole cluster of task within the job queue. In this work, the authors have described and assessed Fuzzy sets to model inexact scheduling parameters and also to symbolize satisfaction grades of each objective. Genetic algorithms with different components are designed on the based technique for task level scheduling in Hadoop MapReduce. In order to achieve a better balanced load across all the nodes in the cloud environment, the scheduler is enhanced by forecasting the execution time of tasks assigned to certain processors and making an optimal decision over the entire group of tasks. This work also assumes centralized scheduling policy where a master processor unit in cloud, collecting all tasks, will take charge of dispatching them to other process units. The system architecture illustrates the data store and computing cluster that jobs could be allocated to the cluster includes machines arranged in a general tree-shaped switched network. The nodes are commodity PCs. Data are distributed through these nodes. There are several replicas for each data block in the distributed file system. By default, the number of replicas is set as 3 in Hadoop. Map tasks generate the intermediate data stored the same node. The design also assumes the communication overhead exits when the data does not locate in the same node as the computing node. However it will require further improvement as this whole rule is predicated on the accuracy of the anticipated execution time of every task. The work is sensible however includes advanced rule and incorporates a tendency to extend network overhead too.

G. Technique-7: Particle Swarm Optimization

Wu et. al [47] has experimented with a set of workflow applications by varying their data communication costs and computation costs according to a cloud price model. First, the algorithm starts with swarm initialization using greedy randomized adaptive search procedure to guarantee each particle in the initial swarm is a feasible and efficient solution. Then, compute the potential exemplars, pbest and gbest, for particles to learn from while they are moving. The stop condition is considered as the user‟s QoS requirements, such as deadline, the budget for computation cost or data transfer cost. The particle‟s new position generation procedure has three steps: 1) select elements from the promising set of pairs with larger probability, that is, the particle learns from gbest and pbest; 2) due to the discrete property of scheduling, there are usually not enough feasible pairs in gbest to generate new position, so the particle will learn from its previous position; 3) all the unmapped tasks should choose resources from other feasible pairs. Finally, gbest will be return as optimal solution. The authors have also compared the total computation cost optimization ratio by varying the tasks number. The result shows that when the task number of the workflow becomes large, their technique optimization ratio increases relatively dramatic. It means the technique can actually achieve lower cost for executing the workflow. Experimental results show that the proposed algorithm can achieve much more cost savings and better performance on makes pan and cost optimization. Result could be better if SLA was considered. The goal of this study was to determine whether the literature on load balancing techniques in cloud computing provides a uniform and rigorous base. The papers were initially obtained in a broad search in four databases covering relevant journals, conference and workshop proceedings. Then an extensive systematic selection process was carried out to identify papers describing load balancing techniques in cloud computing. The results presented here thus give a good picture of the existing load balancing techniques in cloud computing.

VI. CONCLUSION

The random arrival of load in such an environment can cause some server to be heavily loaded while other server is idle or only lightly loaded. Equally load distributing improves performance by transferring load from heavily loaded server. Efficient scheduling and resource allocation is a critical

characteristic of cloud computing based on which the performance of the system is estimated. The considered characteristics have an impact on cost optimization, which can be obtained by improved response time and processing time. 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 re-source utilization ratio by making sure that every computing re-source is distributed efficiently and fairly. With proper load balancing, resource consumption can be kept to a minimum which will further reduce energy consumption. Therefore, there is a need to develop an energy-efficient load balancing technique that can improve the performance of cloud computing by balancing the workload across all the nodes in the cloud along with maximum resource utilization, in turn reducing energy consumption and carbon emission to an extent which will help to achieve Green computing.

REFERENCES

[1] http://www.ibm.com/cloud-computing/us/en/. Accessed on 26th Sep, 2012

[2] Steve Gibbs, Cloud Computing, International journal of Innovative Research in Engineering & Science, issue 1, Volume 1, July, 2012

[3] C. L. Liu, Computing and Communication: There's so much that we share, ICESA 2007

[4] Corbato, Fernando J., "An Experimental Time-Sharing System". SJCC Proceedings. MIT. AIEE-IRE '62 (Spring) Proceedings of the May 1-3, 1962, spring joint computer conference, Pages 335-344, 1962.

[5] Mark Baker, Amy Apon, Rajkumar Buyya, Hai Jin, Cluster Computing and Applications, Encyclopedia of Computer Science and Technology, Allen Kent and Jim Williams (eds), pp.87-125 (Chapter 4), Volume 45 (Supplement 30), ISBN: 0-8247-2298-1, Marcel Dekker, Inc., New York, USA, Jan. 2002.

[6] David Cepeda, Ran Ding, Mohammed Ahmed, Ruth Diaz, Clustered Environments, Report, 2010.

[7] Stephanie Tanner, Todd Frederick, Jeremy Gustafson, Richard Brown, Beowulf cluster computing for all: The HiPerCiC project, 2009

[8] Komal Vashisht, Shefali, Karishma Shukla, A Survey On Grid Computing Approach, IJCST Vol. 1, Iss ue 2, December 2010

[9] Bart Jacob, Grid computing: What are the key components?, ITSO Redbooks, 2003

[10] Bill Allcock, Joe Bester, John Bresnahan, Ann L. Chervenak, Ian Foster1, Carl Kesselman, Sam Meder, Veronika Nefedova, Darcy Quesnel1 Steven Tuecke1, Data Management and Transfer in High-Performance Computational Grid Environments, DOI:10.1016/S0167-8191(02)00094-7

[11] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal, Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities, HPCC 2008: 5-13

[12] Rajnish Choubey, Rajshree Dubey, Joy Bhattacharjee, A Survey on Cloud Computing Security, Challenges and Threats, International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397, Vol. 3 No. 3 Mar 2011

[13] David S. Linthicum, Cloud Computing and SOA Convergence in your Enterprise, Pearson, 2010.

[14] Mehrdad Mahdavi Boroujerdi, Soheil Nazem, Cloud Computing: Changing Cogitation about Computing, World Academy of Science, Engineering and Technology 58 2009.

[15] Top Threats to Cloud Computing V1.0, Cloud Security Alliance, March 2010.

[16] Amazon web service, [Online]. Available: http://aws.amazon.com/

[17] Mary Jo Foley, Microsoft's Windows Azure has a meltdown, February 29, 2012. Retrieved from http://www.zdnet.com/blog/microsoft/microsofts-windows-azure-has-a-meltdown/12076

[18] Chris Talbot, Google Apps Cloud Outage Lessons Learned, Published on 20th July, 2011. Retrieved from http://www.channelinsider.com/c/a/Cloud-Computing/Google-Apps-Cloud-Outage-Lessons-Learned-392917/

[19] Nick Antonopoulos, Lee Gillam, Cloud Computing: Principles, Systems and Applications, Springer, 03-Aug-2010 - 379 pages

[20] http://blog.cloudimpact.com/?p=78. Accessed on 25th Sep, 2012

[21] Z. Kalbarczyk, Toward Resilient Cloud Environment: Virtualization of Error Monitoring and Recovery, March 2012

[22] Neha Thakur, Performance Testing in Cloud: A pragmatic approach, White Paper Submitted for STC 2010

[23] Anastasia Tubanos, FlexiScale Suffers 18-Hour Outage, theWHIR.com on October 31, 2008. Retrieved from http://www.thewhir.com/web-hosting-news/flexiscale-suffers-18-hour-outage

[24] Tim Ferguson, Salesforce.com outage hits thousands of businesses, January 8, 2009. Retrieved from http://news.cnet.com/8301-1001_3-10136540-92.html

[25] Mary Jo Foley, Microsoft apologizes for spate of recent Online Services outages, September 9, 2010. Retrieved from http://www.zdnet.com/blog/microsoft/microsoft-apologizes-for-spate-of-recent-online-services-outages/7337

[26] Fahmida Y. Rashid, Skype Outage Caused by Overloaded Servers, Outdated Desktop Client, Posted 2010-12-29, Retrieved from http://www.eweek.com/c/a/IT-Infrastructure/Skype-Outage-Caused-by-Overloaded-Servers-Outdated-Desktop-Client-749097/

[27] Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region, Retrieved from http://aws.amazon.com/message/65648/

[28] Beth Bacheldor, Lightning takes down a data center. Are solar explosions next?, August 10, 2011, Retrieved from http://www.itworld.com/data-centerservers/192207/lightning-takes-down-data-center-are-solar-explosions-next

[29] Anthony Wing Kosner, Amazon Cloud Goes Down Friday Night, Taking Netflix, Instagram And Pinterest With It, 6/30/2012 , Retrieved from http://www.forbes.com/sites/anthonykosner/2012/06/30/amazon-cloud-goes-down-friday-night-taking-netflix-instagram-and-pinterest-with-it/

[30] Jaspreet kaur, Comparison of load balancing algorithms in a Cloud, International Journal of Engineering Research and Applications, Vol. 2, Issue 3, May-Jun 2012, pp.1169-1173

[31] Kuyoro S. O., Ibikunle F., Awodele O., Cloud Computing Security Issues and Challenges, International Journal of Computer Networks (IJCN), Volume (3) : Issue (5) : 2011

[32] Jing Deng Scott C.-H. Huang Yunghsiang S. Han and Julia H. Deng, Fault-Tolerant and Reliable Computation in Cloud Computing, GLOBECOM Workshops (GC Wkshps), 2010 IEEE

[33] Ritambhara Agrawal, Legal issues in cloud computing, IndicThreads.com conference on cloud computing, June 2011

[34] Clive Longbottom, The Promise and Problems of Cloud Backup and Restore, Quocirca Ltd, 2011

[35] Robert H. Carpenter, Walking from cloud to cloud: the portability issue in cloud computing, Washington journal of law, Technology & Arts Volume 6, Issue 1 Summer 2010

[36] Raksha Nawal, Aashish Gupta, Pragya Nagar, Rajesh Gurjar, Emergence, Performance And Issues Challenging Cloud Computing, Proceedings of the NCNTE-2012, Third Biennial National Conference on Nascent Technologies, 2012

[37] Brian E. Mitchell, Cloud Computing: Intellectual Property Legal Issues, Strafford Publishing Webinar, 2011

[38] K. Ramana, A. Subramanyam and A. Ananda Rao, Comparative Analysis of Distributed Web Server System Load Balancing Algorithms Using Qualitative Parameters, VSRD-IJCSIT, Vol. 1 (8), 2011, 592-600

[39] Vlad Nae, Alexandru Iosup, Radu Prodan, Dynamic Resource Provisioning in Massively Multiplayer Online Games, Transactions on Parallel and Distributed Systems, 2010

[40] Toyoda Y. A simplified algorithm for obtaining approximate solutions to zero-one programming problems. Management Science, vol-21(12): 1417–27, 1975

[41] Aameek Singh, Madhukar Korupolu, Dushmanta Mohapatra, Server-Storage Virtualization: Integration and Load Balancing in Data Centers, Journal of Research and Development Vol-52, 2008

[42] Hao Liu, Shijun Liu, Xiangxu Meng, Chengwei Yang, Yong Zhang, LBVS:A Load Balancing Strategy for Virtual Storage, IEEE International Conference on Service Sciences, 2010

[43] Alan Massaru Nakai, Edmundo Madeira, and Luiz E. Buzato, Improving the QoS of Web Services via Client-Based Load Distribution, XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (Aceito para apresentação). May, 2011

[44] Cao, J., Cleveland, W. Y. G., Jeffay, K., Smith, F., and Weigle, M. (2004). Stochastic models for generating synthetic http source traffic. In Proceedings of INFOCOM, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2004

[45] Srinivas Sethi, Anupama Sahu, Suvendu Kumar Jena, Efficient load Balancing in Cloud Computing using Fuzzy Logic, IOSR Journal of Engineering (IOSRJEN) ISSN: 2250-3021 Volume 2, Issue 7(July 2012), PP 65-71

[46] Sandeep Tayal, Task Scheduling Optimization for the Cloud Computing Systems, IJAEST, 2011

[47] Zhangjun Wu, Zhiwei Ni, Lichuan Gu, Xiao Liu, A Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling, IEEE 2010

Suriya Begum has completed her Bachelor of Engineering in Computer Science and Engg in 1995 from Bangalore University, India. She completed her Master of Technology in Computer Science in 2007 from Allahabad University, India and currently a research scholar in Visveswaraya Technical University, Belgaum, India. She is having almost 19 years of experience as an academician. Her research interest is cloud computing, networking and load balancing.

Dr. Prashanth C.S.R has completed Bachelor of Engineering in Computer Science and Engg from Bangalore University, India. M.S in Computer Science from University of Texas at Dallas and Completed his PhD in Computer Science, from Auburn University in 2006.His research interest is high performance computing, cloud computing, networks and Operating Systems. He is presently working as Professor and Head of Department of Computer Science and Engg, New Horizon College of Engineering, Bangalore. He has published many national and international papers.