IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 26, Issue 4, Ser. 1 (Jul. – Aug. 2024), PP 56-68
www.iosrjournals.org

Cluster Computing and Networking: A Panacea to Network Load Management and Throughput Optimization.

Ikporo, Stephen C. (Ph. D)
Department of Computer Science, Ebonyi State University, Abakaliki – Nigeria
chibeze2010@yahoo.com

ABSTRACT
The increasing rate of the Internet services and Web users in the recent time has led to the wide adoption of distributed Web Servers in form of clusters by Internet Service Providers. This is to enable them service the increasing number of client requests, as stand-alone servers has proven inefficient and can no longer handle the workloads. Stand-alone server allows requests to be processes in turn which causes delay in service delivery and response time, Denial of Service, computational overhead and non-availability in case of server failure. In dealing with these problems, the paper proposes cluster computing and networking technology as an alternative. Cluster computing and networking technology is a system where many servers are connected through Local Area Network or wirelessly and configured to form server farm to behave like a pool of single server with one configured as master. This allows incoming requests to be distributed using the specified algorithm among the servers thereby balancing workloads among them which improves performance, increase availability and throughput. Cluster networking being the interconnection of many network clusters into one cluster ensure that complex operations are solved more efficiently with much faster processing speed and better data integrity with overall throughput optimizing and cost minimization. With this technology, the master server is made to gather node status information with which load distribution and transfer is made. It is as well more scalable as it provides failover server in case of server down time.

Keywords: *Server-Cluster, Failover Server, Cluster Computing, Load Management, Server Nodes.*

Date of Submission: 01-07-2024 Date of Acceptance: 11-07-2024

I. INTRODUCTION

The invention of highly cost and branded super computers has also brought about several changes in computer world. The mainframe computers have also seen some changes and improvements in terms of performance and speed. Past decade witnessed the development of cluster technologies which allow multiple low cost computers to work in a coordinated and controlled fashion to process applications. Despite these improvements notwithstanding, the heterogeneity of the servers and increased web users demands a more flexible and effective computing environments to handle the ever growing loads. Clustering enables a group of independent servers to be managed as a single system for higher availability, easier manageability, and greater scalability [1]. By this, all incoming user's requests are distributed to the servers using the specified load balancing technique to ensure that they are processed as quickly as required. The Distributed Web Servers (DWSs) can either be organized and integrated into a cluster of web servers and linked through local area network to act as a powerful web server or be deployed at different sites over the internet. The additional servers incorporated by the DWSs provide high scalability in response to the growing demands for web services as well as supporting fault tolerant service. This ensures that services are sustained by the functioning servers each time any of the servers fails [2]. A cluster computing can be viewed as the connection of two or more computing devices together for common computing purposes. It can be made of multiple standalone machines connected by a network. The advantages of this are based on the parallel computing power of the computing devices that forms the cluster. This is in addition to the increased processing power as well as high scalability, availability and failover capabilities provided by the network should one of the computing devices which shares computing resources in the cluster network fails [3]. Usually, workloads are distributed to these computing devices to avoid long wait by the central server even without the knowledge of the client/user. This ensures that the clients don't experience any downtime during the operation [4].

The cost effectiveness, performance and flexibility of cluster computing make it an attractive alternative to centralized computing models which are known for their inflexibility, and scalability flaws. Cluster computing utilizes a group of processing speed of the connected computers by cooperatively completing the assigned job in a minimum amount of time. It uses the strategy of transferring the extra loads from busy servers to idle servers. Many enterprises are now looking at clusters of high-performance, low cost computers to provide increased application performance, high availability, and ease of scaling within the data center [4].

II. REVIEW OF RELATED LITERATURE

2.1 Background

Clustered server environment is usually composed of powerful desktop computers with high-speed connectivity connected together to form server cluster/farm. This enables smaller institutions lacking financial resources to harness the equivalent computing power of a multi-processing supercomputer [5]. The clustering algorithm directly affects the performance of the overall system and has become the High Performance Computer (HPC) engine of choice for many industries seeking to process complex operations with much efficiency and greater flexibility, reliability, scalability and performance over traditional workstations or super computers [6] and [7]. Cluster computing system has one master server and one or more computer nodes. The performance of cluster server could be increased by adding more nodes to the cluster or connecting it to another cluster(s) [8].

Before now, there are usually two factors that determine the overall efficiency of cluster network systems; the complexity in its manageability when the node size increases from dozens to hundreds and secondly, how to keep the hardware and software monitoring software up to date. This is even as performing routine maintenance requires significant effort from users [9]. Interestingly, Scalable Cluster Environment (SCE) and Linux Network X projects developed a software suite that includes tools that installs batch scheduler computer node software to manage and monitor computer nodes, and a batch scheduler to address the difficulties in deploying and maintaining clusters so as to operate seamlessly [10] and [11].

Parallel Virtual Machine (PVM) is a popular open-source software system that enables a collection of heterogeneous computers to be used as a coherent and flexible concurrent system. It is a software package for implementing cluster and grid computing. The package is installed on each node in a cluster, and is composed of a daemon that runs in the background, a command-line user-interface, and a communications library that is used to support inter process communication in parallelized applications. XPVM is a GUI interface that can be used to augment the command-line console [12].

2.2 Server Cluster

Server Cluster is defined as the combination of two or more servers which are interconnected to appear as one, thereby creating virtual resource that enhances availability, scalability, robustness, and so on. Server clustering helps the network to operate and be managed as a single entity. The interconnected servers access one another’s disk data. Special software such as Microsoft cluster server is used to manage the network, which automatically detect the failure of any of the servers and quickly provide failover/recovery [12].

According to [13]. Server clusters can be referred to as server farms. In some cases, the interconnected servers have individual operating system, while in others, they share same operating system. Cluster servers increases the availability of the network by ensuring that if a server becomes unavailable due to failure or planned downtime, another server in the cluster takes over the workload of the failed server. Cluster network eliminates loss of service and allows the users or applications that access the cluster to run transparently without the user’s knowledge.

According to [14], a Cluster Server can either be Symmetric or Asymmetric:

i. Symmetric Cluster Server: This is a cluster server in which each server serves as a primary server for a particular set of application. This ensures that every server in the cluster performs a useful function and whenever one server fails; the remaining servers continue to process their own assigned set of applications and that of the failed server. It also allows the connected servers to work together on a program and serves as a good solution for businesses with high processing requirements. Common examples of symmetric cluster servers include web servers, media servers, Virtual Private Network servers, etc. They are more cost effective because they use more of the cluster’s resources more often and as well more scalable, available and flexible.

However, the additional load on the remaining servers whenever any of the servers failed causes system failure. Also, this arrangement can lead to increased infrastructure complexity. By this, it means that some clustering designs can significantly increase the complexity of the network which may affect operational and support requirements [15].

ii. Asymmetric Cluster Server: This is the type of cluster network where there is a standby (redundant) server designed to take over from any server that may fail during operation. This helps provide the needed high availability and scalability for read/write stores such as databases, messaging system, etc. Here, if one server is unavailable due to failure or planned/unplanned downtime, the standby server takes over immediately from the failed server. The standby server performs no other function and maybe as capable or less capable as the primary servers. The standby servers are commonly known as failovers [14].

According to [15], the basic advantages of cluster server network include:

a. **Easy and flexible load management**: Workloads are moved from one particular server to another without affecting the clients' accessibility to data and services.
b. **Uninterrupted availability and fault tolerance**: Cluster server software detects failed server(s) and then transfers the load to the remaining server (failover).
c. **Better scalability**: Workloads are scaled across multiple servers in the cluster. This allows for scaling of network services in conjunction with workload so that additional servers can be added to the cluster as network traffic increases.

2.3 Network Load balancing/Management in a Server Cluster

According to [17], Network load balancing is the process of distributing clients' requests/workloads among various servers in a cluster network in order to improve both job response time and resource utilization. This ensures that no server is heavily overloaded while some are less loaded or idle. It is the process of making all the servers within the cluster to participate in the processing of the client's requests. It ensures even distribution of workloads amongst all the serving entities. It provides failover and high scalability for Internet Protocol (IP) based services that provides support for Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and General Routing Encapsulation (GRE) traffic.

Load balancing is the best way of load distribution over a network, which ensures that the workload on every host is within some small range of the workload available on all the servers in the network. It is always an important activity in the distributed/cluster network which helps tackle the inherent problem of scalability, heterogeneity and adaptability associated with them [17]. Load Balancing can be achieved using specialized software or protocol and also ensuring that specialized gateway or load balancers are installed. This load balancing software can detect link failure and reconfigure the load balancing device in real time to adapt for the failed link. It can also detect interface and cable failure between the server and the switch and automatically redirects inbound and outbound traffic [18].

Network Load Balancing (NLB) in server clusters are addressed by the same set of cluster IP addresses. However, each server in a NLB cluster also maintains its own set of unique and dedicated IP addresses. The implementation of network load balancing allows all servers in the cluster to function collectively and if any one of them fails or goes offline, the load is automatically redistributed among the servers that are up [18].

Every server in a single cluster runs a separate copy of the desired server applications like Web, File Transfer Protocol, and Telnet servers applications. NLB distributes incoming client requests across the servers in the cluster in such a way that loads or traffics to be handled by each server can be defined as necessary. Additionally, more servers can be added dynamically to the cluster in case of increase in load/traffic in future. NLB simply decides which member will respond to the request based on its load balancing and affinity rules [19].

It's not unusual for multiple members to respond to what might seem to be a single request. For example, when a Web browser loads a Web page, it opens multiple connections. One connection retrieves the Hyper Text Markup Language for the page, while additional connections retrieve graphics and other page elements. These requests are sent in parallel and can be handled in parallel by multiple cluster members. NLB does not communicate between the clusters members at all times. Instead, it builds an algorithm table that allows each member to independently determine which requests to handle. This enables the members to act independently for maximum performance [20] and [21].

When a server fails or goes offline unexpectedly, active connections to the failed or offline server are lost. However, to maintain any server within the cluster, one can use the drain stop command to service all active connections prior to bringing the server offline. Once the offline server is back online, it can transparently rejoin the cluster and regain its share of the workload, thereby allowing the other servers in the cluster to handle less traffic thereafter, [16].

According to [18], all servers in an NLB cluster exchange heartbeat messages to maintain consistent communication between them. By default, when any server fails to send heartbeat messages within five seconds, it has failed. Members can change status—such as failing or being taken offline for maintenance from time to time. So, NLB periodically performs a process known as convergence. Convergence always occurs whenever a cluster member's heartbeat fails; it also happens whenever a member is added to, or removed from, the cluster. In the convergence process, NLB uses a multicast transmission to contact each cluster member. In such a scenario, the remaining servers in the cluster converge and do the following:

a. Establish which servers are still active within the cluster.
b. Elect the server with the highest priority as the new default host.

c. Ensure that all new client requests are handled by the remaining active servers.

Network Load Balancing scales the performance of a server-based program, such as Web server, by distributing its client requests among multiple servers within the cluster. With Network Load Balancing, each incoming Internet Protocol packet is received by each host, but only accepted by the intended recipient. The cluster hosts concurrently respond to both different client requests and multiple requests from the same client. For example, a Web browser may obtain the various images within a single Web page from different hosts in a load-balanced cluster. This speeds up processing and shortens the response time to clients [22].

According to [23], Each Network Load Balancing host can specify the load percentage that it will handle, or the load can be equally distributed among all of the hosts. Using these load percentages, each Network Load Balancing server selects and handles a portion of the workload. Clients are statistically distributed among cluster hosts so that each server receives its percentage of incoming requests. In doing this, the balanced loads do not change in response to varying server components (such as Central Processing Unit (CPU) or memory usage). For applications, such as Web servers, which have numerous clients and relatively short-lived client requests, the ability of Network Load balancing algorithm to distribute workload through statistical mapping can effectively distributes loads and provides fast response to cluster changes [24].

Clustered servers emit a *heartbeat messages* to other nodes in the cluster, and listen to the heartbeat from other nodes. If a server in a cluster fails, the remaining nodes adjust and redistribute the workload while maintaining continuous service to their clients. Although existing connections to an offline host are lost, the Internet services nevertheless remain continuously available. In most cases (especially in Web servers), client software automatically retries the failed connections, and the clients experience only a few seconds' delay in receiving a response [25].

2.4 Load Balancing Steps in a Server Cluster.

The balancing of workload in a cluster network is normally through a decentralized controller.
[26], Posited that, the centralized processor distributes all the incoming loads to the servers in the network.
This necessitated the need for the knowledge of the state of each server by the controller. This knowledge helps for judicious assignment of the incoming computational requests to suited servers, using any of the load balancing policies [27]. The necessary information needed by the centralized server includes:

i. Incoming request and throughput
ii. Computational requirements of each workload component
i. Delays imposed by servers and the network on the exchange of measurements and information.
ii. Delays and bandwidth constraints of servers and network components involved in the workload exchange.

The processing capabilities of servers differ from another due to the variation in the architecture, operating system, CPU speed, memory size, and available disk space. And above all, fault tolerance and fault recovery. These form the major challenges in load balancing [28].
In a network, the primary duty of a server includes:

i. Receiving all clients' requests and validating them
ii. Processing all the clients' request.
iii. Sending all the clients' responses back to the client

According to [29], it is common in today's high availability computer systems, that the incoming network traffic is distributed on network level using any one of the common network load balancing algorithms such as random allocation, round robin allocation, weighted allocation, etc.
All the incoming loads to the network cluster from the various clients are terminated at the central server that hosts the load balancer as shown in Figure 1, which in turns forwards the request to the various servers based on the load balancing techniques and mechanism employed [30].

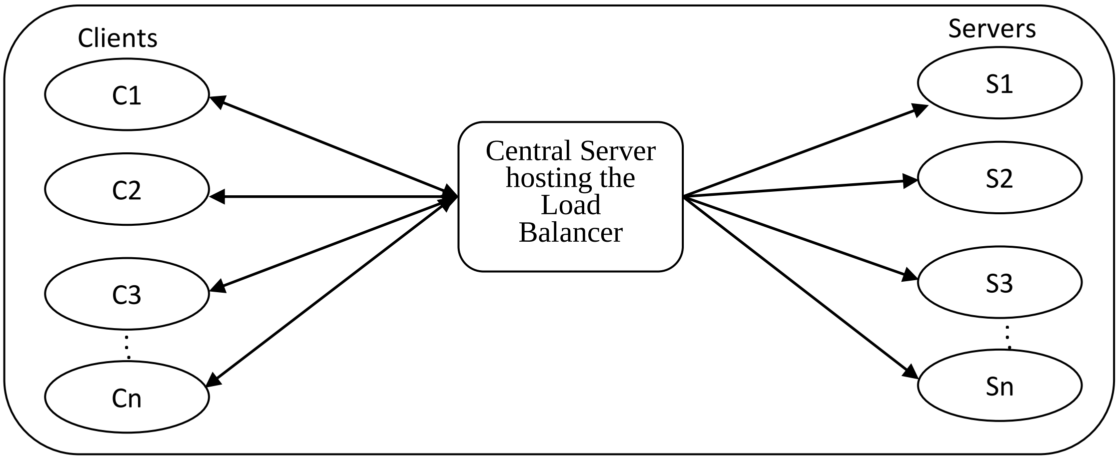

Figure 1: Distribution of loads by Load Balancer to server in a network [30].

2.5. Advantages of Network Load Balancing in a Clustered Network

According to [31], Network Load Balancing provides special value to enterprises that deploys TCP/IP services, such as e-commerce applications, Web, Terminal Services, proxy, Virtual Private Networking (VPN), and streaming media services, etc. These special values are extended advantages of network load balancing and its superiority over other distribution software solutions such as Round Robin- DNS (RR-DNS) which distribute work load amongst multiple server but do not provide a mechanism for server availability. This therefore ensures that if a server within a cluster fails, it will automatically be detected and there will be no further sending of load to it but rather such requests will be redistributed to other available servers. This is not the same with RR-DNS which does not detect easily, failed servers and therefore continue to send clients' requests to them until the network administrator detects and removes the failed server.

Therefore, according to [31], network load balancing provide the following advantages:

a. Scalability: This is the measure of how well a service, application or computer can grow to meet with the increasing demands in the system. In network cluster, scalability entails the ability of the system to incrementally add one or more servers to the existing ones whenever the overall load of the clusters exceeds its capabilities. The network load balancer scales the performance of servers-based program such as web server, by distributing its client requests across the servers within the cluster. This makes it possible to add up to 32 servers within one cluster network.

Scalability also allows a network cluster to remove server from a cluster whenever the load goes down. It as well enables pipelining (i.e. allows for request to be sent to the cluster without waiting for response to the previous sent requests)

b. High Availability: Cluster NLB has advantage of automatically detecting any server that fails or goes offline within the network. It also can recover and redistribute the workload among the remaining server within ten (10) seconds. This ensures that users are provided with continues service with minimal downtime and as well balances the workloads whenever servers are added or removed.

[32], posited that network Load Balancing transparently divides the workloads among the servers and allows the clients access the cluster using one or more "virtual" IP addresses. This makes the cluster servers appear as a single server in the view of the client. Each server runs a copy of an IP-based service, such as Internet Information Services (IIS), and Network Load Balancing distributes the workload among the servers. This speeds up normal processing so that Internet clients see faster turnaround on their requests. For added system availability, the back-end application (eg. database) may operate on a two-server cluster running Cluster service.

c. Manageability: NLB manages and configures multiple NLB clusters and cluster servers from a single computer. It also allows clients' requests to be directed to a single host by using optimal single –host rules especially when a particular server is running specific application. It also enables the definition of different port rules for each website when using the same set of load-balanced servers for multiple applications or websites. This is because port rules are based on the destination virtual IP address in a virtual cluster [33].

d. User Friendliness: Network load balancing enables clients access the cluster using a single, logical Internet name and virtual IP address-known as the cluster IP address. This allows multiple virtual IP addresses for multi-homed servers. And it does not require any hardware change to run.

III. DISCUSSIONS

3.1 Cluster Computing and Networking

Cluster computing and networking is best described as the integration of a number of off-the-shelf commodity computers and resources integrated through hardware, networks, and software to behave as a pool of single computer. Server Cluster can be seen as the interconnection of many nodes (servers) of varied composition for act form and act as one server while Cluster computing is the techniques used to program and drive the resources of the connected servers seamlessly as one single server. Cluster computing and high performance computing were considered as the same before now because of their efficiency. However, modern technologies have redefined cluster computing beyond parallel computing to incorporate load-balancing clusters (e.g., web clusters) and high availability clusters. Clusters are usually employed to address load balancing, parallel processing, systems management, and scalability.

Today, clusters are made up of commodity computers usually restricted to a single switch or group of interconnected switches operating at Layer 2 and within a single virtual local-area network (VLAN). Each server (node) may have different characteristics such as single processor or symmetric multiprocessor design, varied processing speed, and access to various types of storage devices. The underlying network is a dedicated network made up of high-speed, low-latency switches that may be of a single switch or a hierarchy of multiple switches.

Most times, cluster servers are heterogeneous (varied composition) in nature and so differ in price-per-port, bandwidth, latency, processing speed, memory size and throughput. Clusters are not commodities in themselves, although they may be based on commodity hardware. A number of decisions need to be made (for example, what type of hardware the server run on, which interconnect to use, and which type of switching architecture to build on) before assembling a cluster range. Selecting the right cluster elements involves an understanding of the application and the necessary resources that include, but are not limited to, storage, throughput, latency, and number of nodes.

During cluster computing and networking, the MAC addresses of the network adapter and the server are configured into the same subnet is such a way that of the adapter enables communication between cluster servers while that of the server enables communications with servers outside the network. This enhances effective load balancing and increases throughput. The cluster configuration can be in form of one of the following modes:

- **Unicast with Single Network Adapter** – here, the MAC address of network adapter is set at disabled mode and the cluster servers' MAC address are used. All loads from each server in the cluster are collected and filtered by the load balancing driver. The basic feature here is that servers in the cluster are able to communicate with addresses outside the cluster subnet, but node to node communication within cluster subnet is not possible.
- **Unicast with Multiple Network Adapters** - here, the MAC address of network adapter is set at disabled mode and the cluster servers' MAC address are used. All loads from each server in the cluster are collected and filtered by the load balancing driver. The basic feature here is that servers in the cluster are able to communicate with addresses outside the cluster subnet, and also with addresses outside the subnet.
- **Multicast with Single Network Adapters** - here, both the MAC addresses of the network adapter and cluster servers' MAC addresses are enabled. The basic feature here is that servers in the cluster are able to communicate with each other within the cluster subnet and also with addresses outside the subnet. However, where port rules are configured to direct significant levels of traffic to specific cluster servers, it is not recommended.
- **Multicast with Multiple Network Adapters** - here, both the MAC addresses of the network adapter and cluster servers' MAC addresses are enabled. The basic feature here is that servers in the cluster can communicate with each other within the cluster subnet and also with addresses outside the subnet. This configuration is idea for environments where there are significant levels of traffic directed to specific cluster servers.

In some occasions, cluster computers share the same characteristics (homogeneous) and in another varied characteristics (heterogeneous), but in both situations each server contains its own memory and copy of the operating system and application. Each server also communicates with the others via a high-speed interconnect thereby exhibiting a handshake and boosting throughput.

In cluster computing a group of independent servers are networked and managed as a single system for higher availability, easier manageability, and greater scalability. By this, all incoming user's requests are distributed to the cluster servers using the specified load balancing technique. This ensures that all requests are processed as quickly as possible. The distributed web servers are organized and integrated into a clusters and linked through local area network to act as a powerful web server or can been deployed at different sites over the internet. The additional servers enables the distributed web servers provide high scalability in response to the growing demands for web services as well as supporting fault tolerant service. This ensures that services are sustained in case any of the servers fails.

3.2 Types of Server Cluster/Cluster Computing

There are several types of clusters, each with specific design goals and functionality. They include:

i. Load Balancing Cluster

This is the type of cluster which distributes incoming requests among multiple nodes running the same programs or having the same content. Whenever a node fails, its requests are redistributed amongst the remaining available nodes. Load balancing Cluster is typically seen in a web-hosting environment and is a fully functional layer 4 application and employs round-robin load balancing algorithm. With this, the first load is to be sent to the first server, and second load to the second server in that order until server n had been served. They are capable of inspecting the destination Internet Protocol (IP) address of the incoming packet (request) and forwarding it to another server provided all the servers configured on the same subnet. Users' requests which come from devices such as phones, PDAs, Palmtops e.tc usually pass through the web server which serves as the load balancer see figure 2.

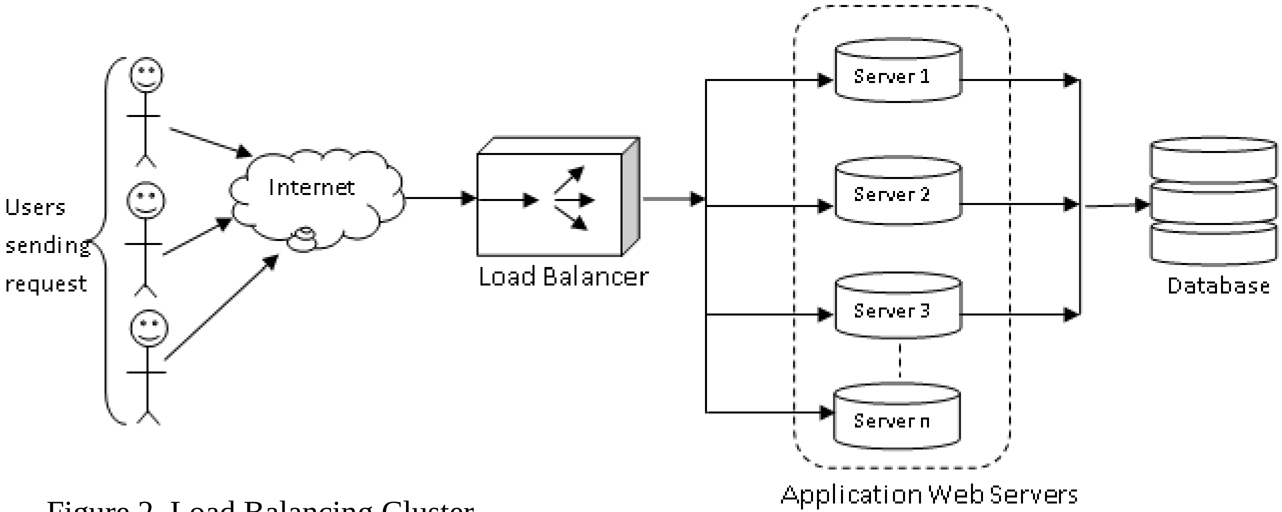

Figure 2. Load Balancing Cluster

Here, each of the web servers in the cluster hosts the floating IP address known to the NLB cluster IP address. This is why the Microsoft server starts load balancing by assigning a single virtual IP addresses to the multiple servers that can respond while Domain Name System (DNS) name is assigned to the virtual IP address.

Remote Desktop Protocol (RDP) clients connect to this DNS name, and the system responds by automatically connecting the user to the least-busy server. Under this architecture, all the configured servers on one single subnet detect incoming network traffic for the cluster's virtual IP address. On each Terminal Server in the cluster, the network load balancing driver acts as a layer residing between the cluster driver and the TCP/IP stack.

One can combine a high availability and load-balancing cluster technologies to increase the reliability, availability, and scalability of application and data resources that are widely deployed for web, mail, news, or FTP services.

Some issues with Load Balancing Cluster include that they can only determine which server is the least-busy based on network load. Secondly, if one server has failed from the cluster farm but is still responding to the network, the system will continue to send users to it with such user's request going to be delayed.

ii. Parallel/Distributed Processing Clusters

These are systems in which multiple processors share a single memory and bus interface within a single computer. With the advent of high speed, low-latency switching technology, computers can be interconnected to form a parallel-processing cluster. These types of cluster increase availability, performance, and scalability for applications, particularly computationally or data intensive tasks. A parallel cluster is a system that uses a number of nodes to simultaneously solve a specific computational or data-mining task.

Unlike the load balancing or high-availability clusters that distribute requests/tasks to nodes and the node processes the entire request, parallel processing divides the request into multiple sub-tasks that are distributed to multiple nodes within the cluster for processing. Parallel clusters are typically used for CPU-intensive analytical applications, such as mathematical computation, scientific analysis (weather forecasting, seismic analysis, etc.), and financial data analysis.

iii. High Availability or Failover Clusters

These are clusters designed to provide uninterrupted availability of data or services usually in web services to the end-users. The technique in failover cluster is to ensure that a single instance of an application is ever available and running at least in one server at a time such that if one cluster node fails and it is offline, the applications will automatically failover to another cluster node. With a high-availability cluster, nodes can be taken out-of-service for maintenance or repairs with little or no harm. Additionally, if a node fails, the service can be restored without affecting the availability of the services provided by the cluster see figure 3.

The processing of customers' requests are shared and performed by the different servers and if one of the nodes fails, all the ongoing processes in the failed server/node are transferred to another available server which also has all the required resources for such operations. This transfer process is called failover. The reverse process, failback, occurs when the failed node becomes active again and the groups that were failed over to the other nodes are transferred back to the original node.

High-availability clusters implementations are best for mission-critical applications or databases, mail, file and print, web, or application servers etc.

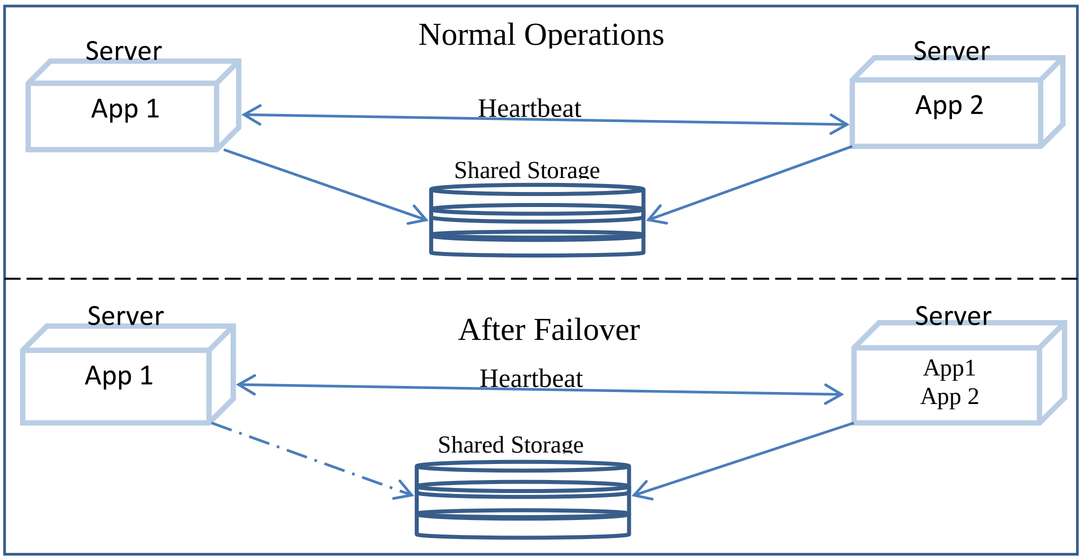

Figure 3: High Availability/Failover Server

High-availability clusters seamlessly and transparently integrate existing standalone, non-cluster aware applications together into a single virtual machine necessary to allow the network to effortlessly grow to meet increased business demands.

The ability to initiate and sustain a handshake/heartbeat amongst servers in cluster network is always the function of an application called **Cluster-Aware and Cluster-Unaware Applications.** Cluster-aware applications are designed specifically for use in clustered environment. This application (e.g Clustered database) knows about the existence of all nodes and is able to communicate with them. Instances of clustered database run in different nodes and have to notify other instances if they need to lock or modify some data. Cluster-unaware applications on the other hand do not know if they are running in a cluster or on a single node. The existence of a cluster is completely transparent for such applications (e.g Web Server), and some additional software is usually needed to set up a cluster. All servers in the cluster have the same content, and the client does not care from which server it provides the requested content.

iv. High Availability + Load Balancing

This is the integration of high availability and load balancing cluster together to provide an extended performance with no complicated breakdowns. The combination of the performances of the two clustering techniques provides the required solution for the network applications and for ISPs too. This integrated technique provides enhanced levels of service quality for conventional network activities, an extensively scalable architecture context and transparent assimilation for stand-alone functionalities and non-clustered collectively in a single virtual system.

3.3 COMPONENT OF NETWORK CLUSTERS

Every computer cluster is composed of five basic building blocks known as cluster components. These components work together to ensure efficient connection and transmission of signals among the connecting devices. Cluster components include the following: cluster application, cluster middleware, cluster operating system, network switching/interconnect hardware and cluster nodes, see figure 4. Significant improvements have been made over the past years to improve the performance of both the computer nodes as well as the underlying switching infrastructure.

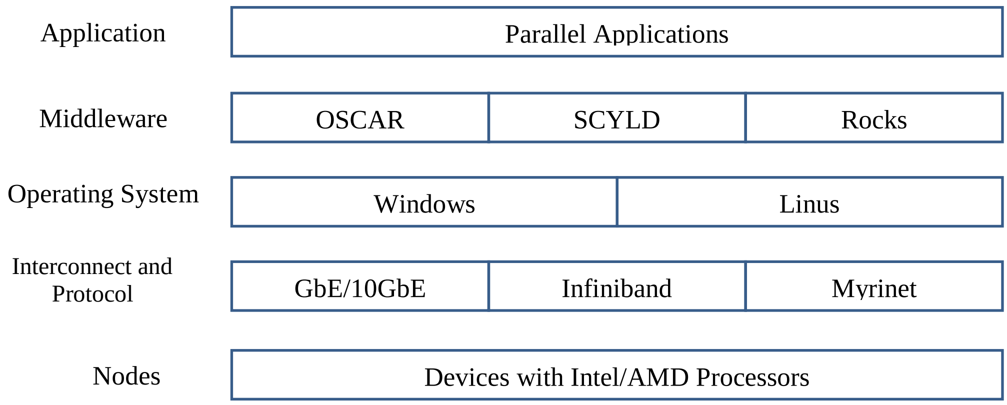

Figure 4: Components of Cluster

Application: This includes all the various applications that are running in group of networked computers. These applications run in parallel and include various queries running on different nodes of the cluster. It can be referred to as the input part of the cluster where users submit their requests.
Middleware: These are software packages which allow the user to interact with the operating system especially in cluster computing. In other words we can say that these are the layers of software between applications and operating system. Middleware provides various services required by an application to function correctly. The software that are used as middleware are: **OSCAR SCYLD ROCK**
Operating System: Clusters can be supported by various operating systems which include Windows, Linux.etc. They are the program that enables interactions between users, application and the hardware.
Interconnect: This includes all the devices that are used to connect the various cluster nodes to one another. Most cluster nodes interconnections are done using 10GbE, Myrinet, Infiniband, etc. In case of small cluster system these and be connected with the help of ample switches

Cluster Node/Server

Nodes refer to devices that house an operating system and a processor which is used to process all the available client requests. Node can as well be referred to as a server. As shown in the figure 6, connected servers in the cluster fall into one of two responsibilities: master (or head) node and compute (or slave) nodes. The master node is the unique server in a cluster system which is responsible for running the file system and also serves as the key system for clustering middleware to route processes, and monitor the health and status of each slave node. A computer (or slave) node within a cluster is responsible for providing the cluster a computing and data storage capability.

Cluster networking and computing architecture usually consist of n number of clients (nodes) connected with cloud service providers via internet and service provider and consisting of m pool of shared cluster servers. The shared pool of cluster servers are configured in such a way that they communicate and share server status and load information to enable the manager server make appropriate decision for load balancing as well as load transfer initiation when necessary, see figure 5.

Normally, load balancing in a cluster is usually done by either a load balancer as found in Window Server OS or a software agent (mobile agent) whose job is to gather server information by constantly checking the status of the I/O networks, memory and CPU utilization so as to have real-time information of every server in the cluster. Window based load balancing techniques in a cluster server are usually static (ie load balancing decisions are taken before load processing). Mobile agents based in the other hand, is usually dynamic (ie load balancing decisions are taken during the loads process). The mobile agents can either reside in the manager server or in slave server. When it resides in a manager server, the agent ensures that loads are transferred or migrated from the overloaded server to less loaded server once transfer initiation is made. The manager server can also initiate an inter node load transfer by instructing the mobile agents to migrate a load from the overloaded to the lesser loaded node. This makes the sending and receiving servers to establish communication so as to commence load migration. If the manager server did not establish that there is an overloaded node, load transfer will not be initiated. Also, the agent which resides in the slave server is usually responsible for gathering server information which is usually communicated to the manager server to enable it know the server status and in return take transfer decisions.

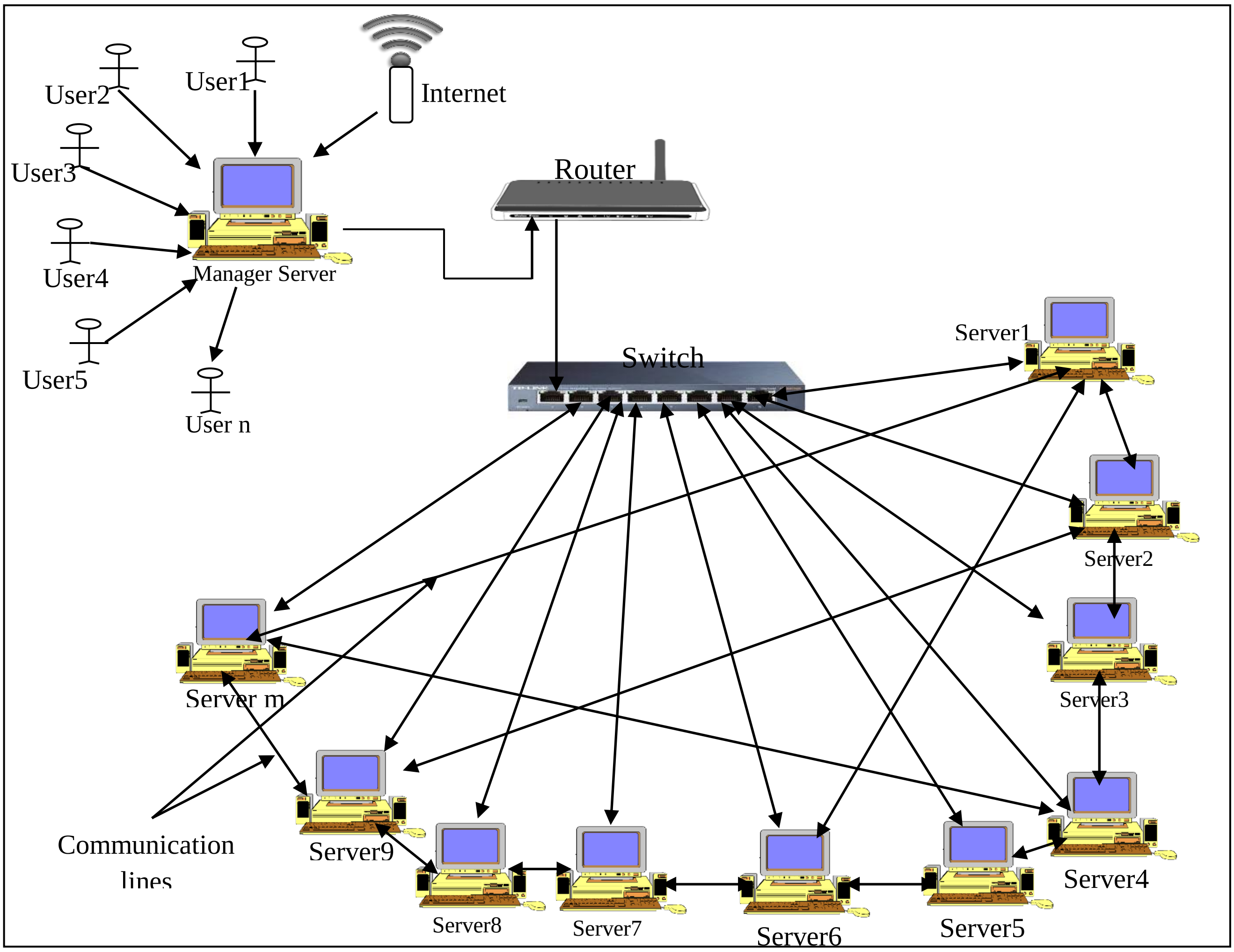

Fig. 5: Network Server Cluster Nodes and information passing mechanism

Characteristics of Cluster Nodes:

Network Cluster nodes have the following characteristics:

- They access all cluster configuration data.
- They communicate with each other using one or more physically independent networks (sometimes referred to as interconnects).
- They are usually aware when another node joins or leaves the cluster.
- They are usually aware of the resources that are running locally as well as the resources that are running the other cluster nodes.
- They are grouped under a common name, the cluster name, which is used for accessing and managing the cluster.

Whenever a node is connected to a cluster, it will first search for active nodes on the networks responsible for internal communication. When found, it will try to join the node's cluster. But, if it cannot find an existing cluster, it will try to form a cluster by taking control of the quorum resource. The quorum resource is where the most current version of the cluster database is stored, and it contains cluster configuration and state data. A server cluster maintains a consistent, updated copy of the cluster database on all active nodes.

Classifications of Network Cluster

Network Clusters can be classified into two categories; Open and Close Cluster.

Open Cluster:

Here, all nodes need Internet Protocols (IPs), which can be accessed through internet/web. Its disadvantages usually are that it can cause more security concern.

Close Cluster:

Nodes connected in a Close Cluster usually hide behind the gateway node and provide better security.

Advantages of Cluster computing and networking

1. **Cost efficiency**: In a Cluster computing Cost efficiency is the ratio of cost to output, which is the connecting group of the computer as computer cluster much cheaper as compared to mainframe computers.
2. **Processing speed**: Cluster computing provides for high processing speed of workloads in the network cluster. This helps eliminate bandwidth utilization and reduces downtime in the network.
3. **Scalability**: The best benefit of cluster computing is that it can be expanded easily by adding the additional node to the cluster. This ensures that at no time will there be much workload that cannot be processed by the nodes within the cluster.
4. **High availability of resources**: In cluster computing, once a node fails, another node within the cluster continues to provide uninterrupted processing. This makes cluster computing to be highly available

Cluster Computing and Grid Computing; what is the difference?

Cluster computing allows computers (nodes) to be connected in a tightly-coupled fashion in such a way that they are all on the same subnet in the same domain, and often with very high bandwidth connections. Nodes in cluster can be homogeneous; they all use the same hardware, run the same software, and are generally configured identically or heterogeneous; have diverse hardware capabilities and software configurations. Each node in a cluster is a dedicated resource--generally only the cluster applications run on a cluster node. One advantage available to clusters is the Message Passing Interface (MPI) which is a programming interface that allows the distributed application instances to communicate with each other and share information. Dedicated hardware, high-speed interconnects, and MPI provide clusters with the ability to work efficiently on “fine-grained” parallel problems, including problems with short tasks, some of which may depend on the results of previous tasks.

In contrast, grid computing allows servers (nodes) to be loosely-coupled; they may exist across domains or subnets. The nodes can be heterogeneous; they can include diverse hardware and software configurations. Grid computing is a dynamic system that can accommodate nodes coming in and dropping out over time. This ability to grow and shrink where necessary contributes to grid’s ability to scale applications easily. In Grid computing, there is no need for high-performance interconnects; rather, they are usually configured to work with existing network connections. As a result, Grid computing are better suited to relatively “coarse-grained” parallel problems, including problems composed primarily of independent tasks. There is no dominant programming paradigm in grid computing today, and a key challenge to increasing the acceptance of grid computing is creating grid-enabled applications with familiar programming models such as Digiped’s Object-Oriented Programming for Grid (OOP-G).

World over, Grid computing is the most distributed form of parallel computing which makes use of computers communicating over the Internet to work on a given problem. The low bandwidth and extremely high latency available on the Internet, the distributed computing typically deals only with parallel problems.

Most grid computing applications make use of middleware (software that sits between the operating system and the application to manage network resources and standardize the software interface). The most common distributed computing middleware is the Berkeley Open Infrastructure for Network Computing (BOINC). Often, distributed computing software makes use of "spare cycles", performing computations at times when a computer is idling.

Nonetheless, Grids computing and cluster computing can be incorporate together. Often the best way to make use of all available resources is to manage the cluster resources as part of a larger grid, assigning jobs and tasks to the resources best suited to those jobs and tasks. For example, jobs requiring MPI would be assigned exclusively to the cluster, while loosely-coupled jobs could be assigned to all grid nodes, including those in the cluster (when available). Indeed, cluster compute nodes make excellent grid nodes, and many grids are composed exclusively of dedicated servers. On the Windows operating system compute clusters are supported by Windows Compute Cluster Server 2003 and grid computing is supported by the Digipede Network™. The chart below gives an overview of the two solutions. Both systems use similar terminology to define submitted requests: A job defines the work submitted to the system which includes the required resources and the tasks to execute. A task is an individual unit of work that can be executed concurrently with other tasks.

Benefits of Clusters Computing and networking

The main benefits of clusters are scalability, availability, and performance. For scalability, a cluster uses the combined processing power of compute nodes to run cluster-enabled applications such as a parallel database server at a higher performance than a single machine can provide. Scaling the cluster's processing power is achieved by simply adding additional nodes to the cluster. Availability within the cluster is assured as nodes within the cluster provide backup to each other in the event of a failure. In high-availability clusters, if a node is taken out of service or fails, the loads transferred to another node (or nodes) within the cluster. To the user, this operation is transparent as the applications and data running are also available on the failover nodes. An additional benefit comes with the existence of a single system image and the ease of manageability of the cluster. From the users perspective the users sees an application resource as the provider of services and applications. The user does not know or care if this resource is a single server, a cluster, or even which node within the cluster is providing services. These benefits map to needs of today's enterprise business, education, military and scientific community infrastructures. In summary, clusters provide:

- Scalable capacity for computing, data, and transaction intensive applications, including support of mixed workloads
- Horizontal and vertical scalability without downtime
- Ability to handle unexpected peaks in workload
- Central system management of a single systems image
- 24 x 7 availability

IV. CONCLUSION

High-performance cluster Networking and Computing enables a new class of computationally intensive applications that are solving problems that were previously cost prohibitive for many enterprises. The use of high-speed commodity computers collaborating to resolve highly complex, computationally intensive tasks has broad application across several industry verticals such as chemistry or biology, quantum physics, petroleum exploration, crash test simulation, CG rendering, and financial risk analysis. They are also applied in some of the critical Applications such as Google Search Engine, Petroleum Reservoir Simulation, Earthquake Simulation and Weather Forecasting. Cluster computing helps solving complex operations more efficiently with much faster processing speed and with better data integrity than a single computer and are used for mission-critical applications.

However, cluster computing pushes the limits of server architectures, computing, and network performance. Due to the economics of cluster computing and the flexibility and high performance offered, cluster networking and computing has made its way into the mainstream enterprise data centers using cluster servers of various sizes. As cluster networking and computing become more popular and more pervasive, careful consideration of the application requirements and what that translates to in terms of network characteristics becomes critical to the design and delivery of an optimal and reliable performing solution.

Cluster computing and networking has made it possible for two or more computing devices to be connected together into a network and configured for common computing purposes. The various parallel computing powers of these devices are made to form a single system called cluster, thereby increasing its processing power. With the new system being made up of more than one server, workloads are distributed among the connected servers in the cluster. Extra loads from busy servers are transferred to idle servers, hence ensuring better load management. The new system is more scalable, available and provides failover capabilities should any of the computing devices fails.

The cost effectiveness, performance and flexibility of cluster computing and networking make it an attractive alternative to centralized computing models which are known for their inflexibility, and scalability flaws. Many enterprises are now looking at cluster computing and networking to provide increased application performance, high availability, and ease of scaling within the data center. A scalable and modular networking solution is critical, not only to provide incremental connectivity but also to provide incremental bandwidth options as the cluster grows. The technologies associated with cluster computing, including host protocol stack-processing and interconnect technologies, are rapidly evolving to meet the demands of current, new, and emerging applications.

REFERENCES

[1]. Safiriyu, E., Olatunde, A., Ganiyu A., Ayodeji, O., Clement, O. and Lawrence, K., (2010). A load Balancing Policy for Distributed Web Service. International Journal of Communication, Network and System Science, Vol. 3, pp. 645 – 654.

[2]. Ajay, A. J., Shriwastava, S. K., (2012). Design of an Optimized Virtual Server for Efficient management of Cloud Load in Multiple Cloud Environments. International Journal of Application or Innovation in Engineering & Management. Vol. 1, Issue 3.

[3]. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616.

[4]. Abdallah, M., Griwodz, C., Chen, K. T., Simon, G., Wang, P. C., & Hsu, C. H. (2018). Delay-sensitive video computing in the cloud: A survey. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14(3s), 1-29.

[5]. Schmeisser, M., Heisen, B. C., Luettich, M., Busche, B., Hauer, F., Koske, T., ... and Stark, H. (2009). Parallel, distributed and GPU computing technologies in single-particle electron microscopy. Acta Crystallographica Section D: Biological Crystallography, 65(7), 659-671.

[6]. Raj, P., Raman, A., Nagaraj, D., & Duggirala, S. (2015). High-performance big-data analytics. Computing Systems and Approaches, Springer, 2015, 1.

[7]. Rao, P. S., Jana, P. K., & Banka, H. (2017). A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks. Wireless networks, 23(7), 2005-2020.

[8]. Yang, C. T., & Chang, S. C. (2004). A parallel loop self-scheduling on extremely heterogeneous PC clusters. Journal of Information Science and Engineering, 20(2), 263-273.

[9]. Shalf, J., Dosanjh, S., & Morrison, J. (2010). Exascale computing technology challenges. In International Conference on High Performance Computing for Computational Science (pp. 1-25). Springer, Berlin, Heidelberg.

[10]. Harr, J., & Denault, G. (2002). Issues concerning Linux clustering: cluster management and application porting. In Proceedings 16th International Parallel and Distributed Processing Symposium (pp. 6-pp). IEEE.

[11]. Putchong, U., Paisitbenchapol, S., Angskun, T., & Maneesilp, J. (2000). System management framework and tools for beowulf cluster. In Proceedings Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region,Vol. 2, pp. 935-940. IEEE.

[12]. Yeo, C. S., Buyya, R., Pourreza, H., Eskicioglu, R., Graham, P., and Sommers, F. (2006). Cluster computing: High-performance, high-availability, and high-throughput processing on a network of computers. In Handbook of nature-inspired and innovative computing (pp. 521-551). Springer, Boston, MA.

[13] Heo, J., Yi, S., Cho, Y., and Hong, J. (2007). PEACH: Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks. Computer communications, 30(14-15), 2842-2852.

[15] Rajamani, K., and Lefurgy, C. (2003). On evaluating request-distribution schemes for saving energy in server clusters. In 2003 IEEE International Symposium on Performance Analysis of Systems and Software. ISPASS 2003. pp. 111-122. IEEE.

[16] Zhang, S. and Shufen Z., (2010). The Comparison between Cloud Computing and Grid Computing, International Conference on Computer Application and System Modelling (ICCASM), IEEE, pp. 72-75.

[17] Ramana, K. K. and Mahesh, V. G., (2005). Load Balancing of Services with Initiated Connections. ICPWC.

[18] Cardellini, V., et al, (2002). The State of the Art on Locally Distributed Web-Server System. ACM Computing Surveys, 35(4):1-49.

[19] Che-Lun, H., Hsiao-his, W. and Yu-Chen, H., (2012). Effective Load Balancing Algorithm for Cloud Computing Network. International Conference on Information Science and Technology. Pp. 251- 253.

[20] Mah, B. A. (1997). An empirical model of HTTP network traffic. In Proceedings of INFOCOM'97 (Vol. 2, pp. 592-600). IEEE.

[21] Rossberg, J., and Redler, R. (2004). Cluster Techniques. In Designing Scalable. NET Applications (pp. 59-98). Apress, Berkeley, CA.

[22]. Cardellini, V. Colajanni, M. and Yu, P. S., (1999). Dynamic Load Balancing on Web-

[23]. Dharmesh, K. and Jaydeep, V., (2014). A Survey of various Load Balancing Algorithms in Cloud Computing. International Journal of Scientific and technology Research. Vol. 3, Issue 11.

[24] Shkapenyuk, V., and Suel, T. (2002). Design and implementation of a high-performance distributed web crawler. In Proceedings 18th International Conference on Data Engineering (pp. 357-368). IEEE.

[25] Kambourakis, G., Klaoudatou, E., Konstantinou, E., and Gritzalis, S. (2010). A survey on cluster-based group key agreement protocols for WSNs. IEEE Communications surveys & tutorials, 13(3), 429-442.

[26] SrinivasaRao, P., (2017). A framework for Scalable Distributed Job Processing with Dynamic Load Balancing using Decentralized Approach. Laxmi Book Publication.

[27] Jyoti, V., Anant, K. J., (2013). Comparative Study o Load Balancing Algorithms. IOSR Journal of Engineering.Vol. 3, Issue 3, pp.45 – 50.

[28] Saber, R. O. and Murray, R. M., (2003). Consensus protocols for networks of dynamic agents. In American Control Conference. Proceedings, Vol. 2, pp. 951-956.

[29] Mendiola, A., Astorga, J., Jacob, E., & Higuero, M. (2016). A survey on the contributions of software-defined networking to traffic engineering. IEEE Communications Surveys & Tutorials, 19(2), 918-953.

[30]. Beaulah, S., Sandhya, R. A., Ritesh, k. S., Thriveni, J., Venugopal, K. R. and Patnaik, L. M. (2012). Comparative Study on Load Balancing Techniques in Distributed Systems. International Journal of Information Technology and Knowledge Management, Vol. 6, No.1, pp. 53-60.

[31] Iyengar, A. MacNair, A. and Nguyen, E., (2000). An Analysis of Web Server Performance, .IEEE GLOBECOM '97, Volume: 3, pp. 1943-1947.

[32] Firoj, M. and Rafiqul, Z. k., (2012). The Study of load Balancing Strategies in Distributed Computing System. International Journal of Computer Science and Engineering Survey. Vol. 3 No. 2.

[33]. Dhurandher, S. K., Mohammad, S. O., Isaac, W., Pragya, A., Abhishek, G. and Prateek G., (2014). A Cluster-based Load Balancing Algorithm in Cloud Computing. **In** Communications, IEEE International Conference. Pp. 2921 – 2925.