EnhancedQoS in Distributed System Using Load Balancing Approach

Arju Malik [1], PankajPratap Singh[2]

arzoomalik17@gmail.com, pankajpratapsinghcs@gmail.com

[1, 2] *Department of Computer Science & Engineering and Information Technology,*

[1, 2] *Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh, India*

Abstract:*A load balancing approach is undertaken to methodically characterize the average overall conclusion time in a distributed system to improve the quality of services. This comes close toconsider the delays imposed by the communication medium. The load balancing approach is developed to improve the certain parameters of quality of services that dynamically reallocate the incoming external loads at each server. One of the most important issues in distributed systems is to design of aproficient dynamic load balancing algorithm that improves the taken as a whole performance of the distributed systems. Scheduling and resource management plays animportant role in achieving high utilization of the resource in grid computing environments. Load balancing is the procedure of distributed the load betweendiverse nodes of a distributed method to find improved both job response time withreservedeployment while also avoiding a situation where altered of the nodes are deeply loaded while other nodes are idle or lightly loaded. The performance the load balancing approach is shown in a simulated way using progress bars that shows how progress bar at each server performs before using load balancing approach and after using load balancing approach.*

Keywords: *Load balancing, Distributed system, Internet, Server, QoS.*

1. INTRODUCTION

Load balancing is dividing the total of work that a computer has to do among two or more computers so that more effort gets done in the same quantity of time and, in general, all users obtain served nearer. Load balancing can be implemented with hardware, software, or a combination of both. Typically, load balancing is the key reason for computer server clustering[3]. On the Internet, companies whose Web sites acquire a great deal of traffic frequently use load balancing. For load balancing Web traffic, there are several approach. For Web serving, one approach is to transmit each request in turn to a different server host address in a domain name system (DNS) table, round-robin fashion. Frequently, if two servers are used to balance a work load, a third server is required to determine which server to assign the effort to. Since load balancing requires multiple servers, it is regularlyshared with failover and backup services. In some approaches, the servers are distributed greater than different geographic locations[1].The most favourable one-shot load balancing policy is developed and subsequently extensive to develop an autonomous and distributed load-balancing strategy that can dynamically reallocate received external loads at each node. This adaptive and dynamic load balancing strategy is implemented and evaluated in a two-node distributed system. The performance of the projected dynamic load-balancing policy is compare to that of static policies as well as existing dynamic load-balancing policies by allowing for the average completion time per task and the system handing out rate in the presence of random arrivals of the external loads[10].

1.1 LOAD BALANCING

Load balancing is a computer networkingtechnique to allocate workload across multiple computers or a computer cluster network links[2], central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, reduce response time, and avoid overload. Using multiple components with load balancing, instead of a single component, can increase reliability duringredundancy. The load balancing service is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server. This might include forwarding to a backup load balancer, or displaying a message about the outage [3]. Load balancing gives a chance to achieve a significantly higher fault tolerance. It can repeatedly provide the amount of capacity needed to respond to any increase or decrease of application traffic.

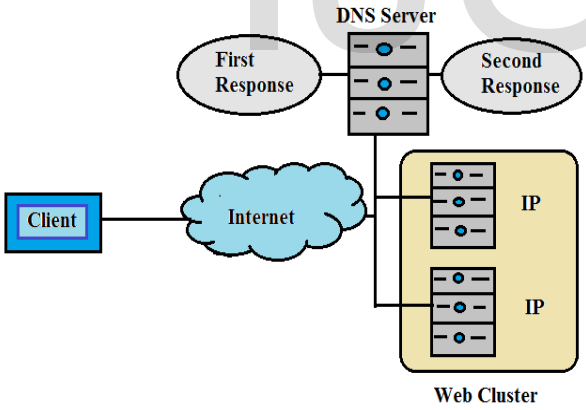

Fig 1: Basic DNS Response for Redundancy

1.2 DISTRIBUTED SYSTEM

Distributed computing is a field of computer scienceto studies distributed systems. A distributed system consists of numerouscomputers that communicate through a computer network. The computers work together with each other in order to achieve a common goal[4]. A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of symbols such programs. Distributed computing also refers to the use of distributed systems to resolve computational problems. In distributed computing, a problem is separated into many tasks, each of which is solved by one or more computers which commune with each other by message passing. The structure of the system (network topology, network latency, number of computers) is not known in progress, the system may consist of different kind of computers and network links, and the system may change through the execution of a distributed program. Each computer has only a partial, incomplete view of the system. Each computer may know only one part of the input. Distributed systems are groups of networked computers, which have the same goal for their work[5]. The terms concurrent computingparallel computing and distributed computing have a group of overlap, and no clear distinction exists between them. The situation is further difficult by the traditional use of the terms parallel and distributed algorithm that do not quite match the above definitions of parallel and distributed systems.

1.3 DISTRIBUTED NETWORK

Distributed Networking is a distributed computing network system, said to be "distributed" when the computer programming and the data to be worked on are spread out over further than one computer. Usually, this is implemented over a network. Prior to the appearance of low-cost desktop computer power, computing was generally centralized to one computer[6]. Although such centre still exist, distribution networking applications and data operate more efficiently over a mix of desktop workstations, local area network servers, regional servers, Web servers, and other servers. One popular trend is client/server computing. This is the standard that a client

computer can provide certain capabilities for a user and request others from other computers that provide services for the clients.

1.4 QUALITY OF SERVICES

The quality of service (QoS) refers to a number of related aspects of telephony and computer networks that permit the transport of traffic with special needs. In particular, much technology has been developed to allow computer networks to be converted into as useful as telephone networks for audio conversation, as well as following new applications with even stricter service demands. In the field of telephony, quality of service was defined by the ITU in 1994. Quality of service comprises needs on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, cross-talk, echo, interrupts, frequency response, loudness levels, and so on. During the session it might monitor the achieved level of performance, for example the data rate and delay, and dynamically control scheduling priorities in the network nodes[7].

1.5 QUALITIES OF TRAFFIC

In packet-switched networks, quality of service is exaggerated by various factors, which can be divided into “human” and “technical” factor. Human factors include: constancy of service, availability of service, delays, user information. Technical factors include: reliability, scalability, effectiveness, maintainability, grade of service etc.New Networking technologies, Service and Application are all arriving on the market, each with anplan to deliver a Quality of Service (QoS) that is equal to or improved than the legacy equipment, Service providers and network operators have trusted brands, the maintenance of which is critical to their business[8]. There are manykind of network services such as leased line service, IP-VPN service, xDSL services, Frame relay service, etc. Also, QoS parameters are the target of SLA monitoring and NPMs are required to measure the network performance and guarantee QoS parameters. QoS parameter is the example to represent the quality of service to customers. It should be easy for customers to recognize the degree of assuring the service[9]. QoS parameters can be different according to the type of services.We categorize the NPMs into four types: Availability, Loss, Delay and Utilization. The meaning of each NPM is as follows.

1.5.1 LOW THROUGHPUT

Due to varying load from other users input the same network resources, the bit rate (the ceiling throughput) that can be provided to a definite data stream may be too low for real time multimedia services if all data streams acquire the same scheduling priority.

1.5.2 DROPPED PACKETS

The routers might fail to deliver (*drop*) a number of packets if their data is corrupted or they arrive when their buffers are already full. The receiving function may ask for this information to be retransmitted, possibly causing severe delays in the overall transmission.

1.5.3 ERRORS

Sometimes packets are corrupted due to bit errors cause by noise and interference, especially in wireless communications and long copper wires. The receivers have to detect this and just as if the packet was dropped, may ask for this information to be retransmitted.

1.5.4 LATENCY

It may take a long time for each packet to reach its destination, because it gets held up in long queues, or takes a less direct route to evade congestion. This is different from throughput, as the delay can build up over time, even if the throughput is almost normal. In some

cases, unnecessary latency can render an application such as VoIP or online gaming unusable.

1.5.5 JITTER

Packets from the source will reach the target with different delays. A packet's delay varies with its position in the queues of the routers along the path connecting source and destination and this position can vary unpredictably. This variation in delay is known as jitter and can seriously affect the quality of streaming audio and/or video.

1.5.6 OUT-OF-ORDER DELIVERY

When a collection of connected packets is routed through a network, different packets may take different routes, each resulting in a different delay. The effect is that the packets arrive in a different order than they were sent. This problem requires special additional protocols responsible for rearranging out-of-order packets to an isochronous status once they reach their destination.

2. COMPONENT QUALITY MODELING LANGUAGE (CQML)

CQML includes four types of specification constructs. The basic construct is QoS characteristic. QoS characteristics are defined as user-defined types. QoS characteristics arethen grouped and restricted into specifications of QoS. For this, QoS statements that constraineach constituent QoS characteristic within specific ranges of values are defined. These twoconstructs focus on specification of QoS independent of what interface it annotates and howthe QoS mechanism is implemented[10]. A third construct, QoS profiles, relate QoS statements tospecific components or parts thereof. Finally, QoS categories are used to group any of thethree aforementioned concepts.

2.1 THE .NET FRAMEWORK

The .NET Framework is computing platforms that simplify application development in the highly distributed environment of the Internet. To provide a reliable object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.Eliminate the performance problems.There are different types of application, such as Windows-based applications and Web-based applications[7]. To make communication on distributed environment to ensure that code be accessed by the .NET Framework can combine with any other code.

2.2 COMPONENTS OF .NET FRAMEWORK

2.2.1 THE COMMON LANGUAGE RUNTIME (CLR):

The common language runtime is the establishment of the .NET Framework. It manages code at execution time, providing vital services such as memory management, threads management, and remote and also ensures more security and robustness. The concept of code organization is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while codes that do not target the runtime are known as unmanaged code.

2.2.2 THE .NET FRAME WORK CLASS LIBRARY:

It is a comprehensive, object-oriented set of reusable types used to develop applications ranging from traditional command-line or graphical user interface applications to applications based on the latest innovations provided by ASP.NET, such as Web Form and XML Web service. The .NET Framework can be hosted by unmanaged components that load the common language runtime into their process and initiate the carrying out of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but

moreover supports the development of third-party runtime hosts.

2.3 FEATURES OF ASP.NET

ASP.NET is the version of Active Server Pages (ASP); it is a unified Web development platform that provides the services for developer to build enterprise-class Web applications. While ASP.NET is largely syntax compatible, it also provide a programming model and infrastructure for more secure, scalable, and stable applications. ASP.NET is a compiled, NET-based environment, we can creator applications in any .NET compatible language, including Visual Basic .NET, C#, and J Script .NET. Additionally, the entire .NET Framework is available to any ASP.NET application. Developers can simply access the benefits of these technologies, which include the managed common language runtime environment, type safety, inheritance, and so on.ASP.NET has been considered to work seamlessly with WYSIWYG HTML editors and other programming tools, with Microsoft Visual Studio .NET. Not only does this make Web development easier, except it also provides all the benefits that these tools have to offer, including a GUI that developers can use to drop server controls onto a Web page and fully integrated debugging support.

2.4 MODULES

- Authentication Module.
- IP Address Representation Module.
- Load Servers Module.
- Load Balancing Module.
- Report Module

2.4.1 AUTHENTICATION MODULE:

The authentication module is to register the new user and previously registered users can enter into our project. The administrator only can enter and do the uploading files into the servers. After login by every user and the admin the SQL server checks the login id and password is valid or not. If the login is not valid it displays that the login is not correct.

2.4.2 IP ADDRESS REPRESENTATION MODULE:

The IP Address depiction module is to give the IP addresses which we are going to allocate those as servers. We can enter and view IP addresses from the module. In load balancing system we can connect the three servers [system]. The connection has to be represented by the IP Address representation only.

2.4.3 LOAD SERVERS MODULE:

The Load Servermodules have the authentication for the administrator only can enter into this module. The administrator will perform the encryption of the text file and store into the server which we are assigned in IP representation module. This module will construct the both public and private key for the cryptography[11].

2.4.4 LOAD BALANCING MODULE:

The Load Balancing modules have the authentication for users can enter into the upload page and can view the file name which the administrator store into the servers. The user can select the file from the list and can download from the server which is in inactive state[16]. We will get the response time and from which server we are getting the file[12]. Finally we can get the decrypted file from the key pair.

2.4.5 REPORT MODULE

We will get the response time and as of which server we are getting the file. It compares the response time between

the servers and downloads the given file in the enhanced performance response time server[13].

3. PROPOSED MODEL

A regeneration-theory approach is undertaken to analytically characterize the normal overall completion time in a distributed system. The advance considers the heterogeneity in the handing out rates if the nodes as well as the arbitrariness in the delays imposed by the communication medium[14]. The optimal one-shot load balancing policy is developed and subsequently absolute to extend an autonomous and distributed load-balancing policy that can dynamically reallocate incoming peripheral loads at each node. This adaptive and dynamic load-balancing is implemented in addition toevaluate in a two-node distributed system. The performance of the proposed dynamic load-balancing policy is compared to that of static policies as well as existing dynamic load-balancing policies by allowing for the average completion time per task and the system processing rate in the existence of random arrivals of the external loads[15]. So Here by we conclude that with the increase in number of users there is problem of time delay that arises. We are investigating this time delay problem and improving it using the method called LOAD BALANCING method and algorithm[17].

Algorithm:

Step 1: Registration is done.

Step 2: Enter the Username and Password for Client

Step 3: The Username and Password is incorrect then a message is shown.

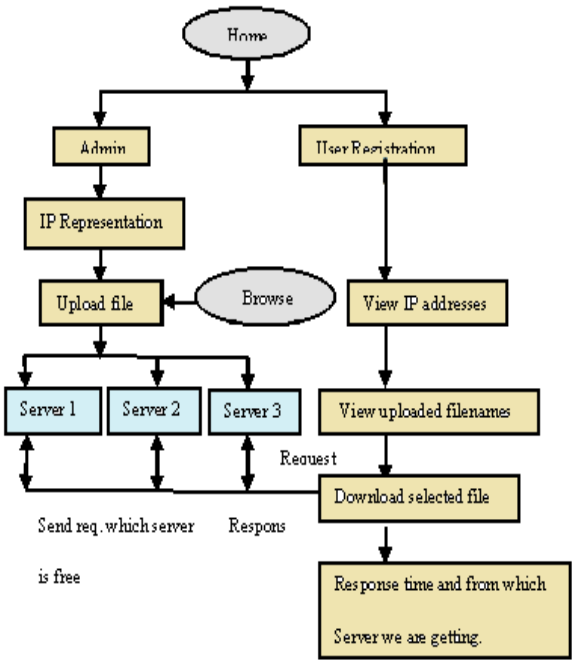

Fig 2: Process Diagram

Step 4: Login is correct then creates IP address

Step 5: Again enter Admin username and password

Step 6: View the existing IP address or server Name

Step 7: Select IP address and Upload the File

Step 8: Finally view the Network load balancing using progress bar.

4. RESULT AND CONCLUSION

In this paper we studied the load balancing strategies lucidly in detail. Distributed system using load balancing is the most propel area in research today as the demand of heterogeneous computing due to the wide use of internet. In this project, we have investigated the delay problem on load balancing for distributed system. Due to communication delay among servers, the load balancing process may be using outdate load information from local

servers to compute the balancing flows. As we have shown, this would significantly affect the concert of the load balancing algorithm. We have just reviewed this work for our understanding. We have calculated the average data rate for the servers i.e. time under which they have to accomplish their work. The load is assigned to them according to their bandwidth. No server remains ideal and no delay is encountered. The bandwidth is fully utilised. Thus, the project improves the quality of services in distributed system using load balancing approach.

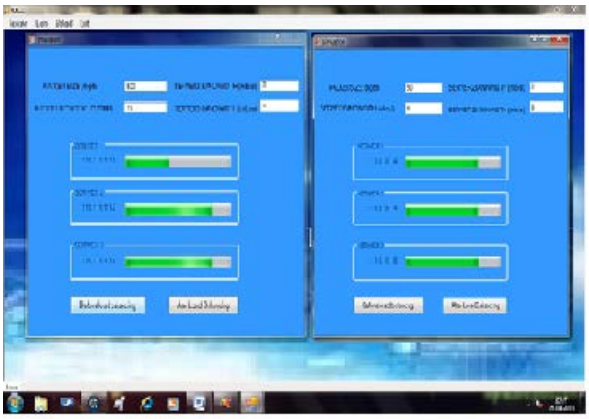

Fig 5: Simulation Form

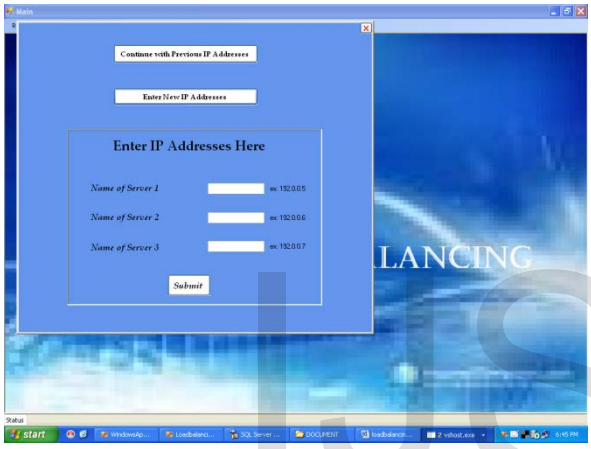

Fig 3: New IP Address Form

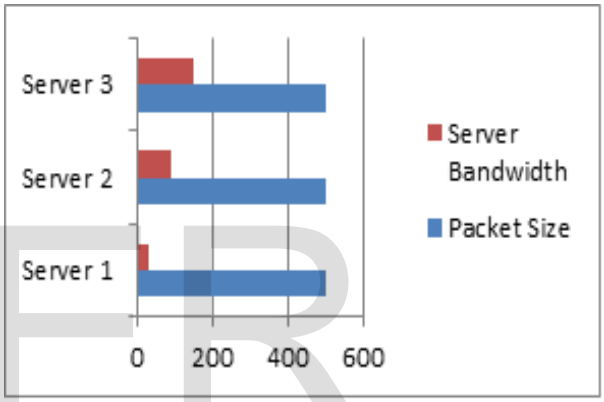

Fig 6: Before Load Balancing

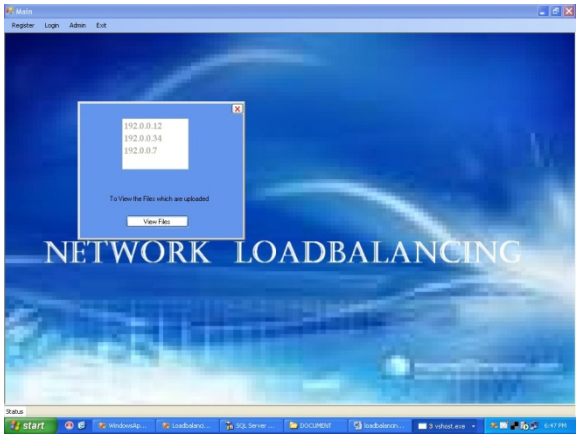

Fig 4: View Files Form

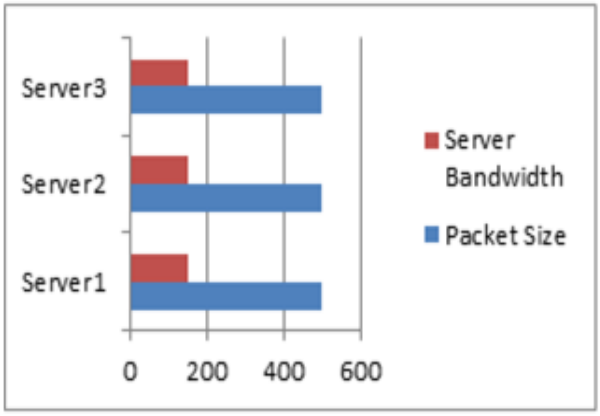

Fig 7: After Load Balancing

5. FUTURE WORK

A regeneration-theory draw near is undertaken to analytically exemplify the average overall completion time in a distributed system. The development has covered almost all the requirements. Further requirements and improvements can simply be done since the coding is mostly structured or modular in nature. Improvements can be appended by changing the existing module or adding new module. One important development that can be added to the task in future is file level backup, which is presently done for folder level.

6. REFERENCES

[1] Md. Firoj Ali and RafiqulZaman Khan. "The Study On Load Balancing Strategies In Distributed Computing System". International SJournalOf Computer Science & Engineering Survey (IJCSES) Vol.3, No.2, April 2012

[2] Ahmad I., Ghafoor A. and Mehrotra K. "Performance Prediction of Distributed Load Balancing On Multicomputer Systems". ACM, 830-839, 1991.

[3] Antonis K., Garofalakis J., Mourtos I. and spirakis P. "A Hierarchical Adaptive Distributed Algorithm for Load Balancing". Journal of Parallel and Distributed Computing, Elsevier Inc. 2003.

[4] Kokilavani .K . "Enhance Load Rebalance Algorithm for Distributed File Systems in Clouds". International Journal of Engineering and Innovative Technology (IJEIT). Volume 3, Issue 6, December 2013.

[5] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, April 2010.

[6] S.Penmatsa and T.Chronopoulos, Game-theoretic static load balancing for distributed systems, Journal of Parallel and Distributed Computing, vol.71, no.4, pp.537-555, Apr. 2011.

[7] E. Altman, T. Basar, T. Jimenez, and N. Shimkin.Routing in two parallel links: Game- theoretic distributed algorithms. J. Parallel and distributed computing, 61(9):1367-1381, September 2001.

[8] P. BeaulahSoundarabai, Sandhya Rani A., Ritesh Kumar Sahai, Thriveni J., K.R. Venugopaland L.M. Patnaik, "Comparative Study on Load Balancing Techniques in Distributed Systems",International Journal of Information Technology and Knowledge Management, pages 63(7), 2009.

[9] Heiner Ackermann, Simon Fischer, and Martin Hoefer. Distributed algorithms for QoS load balancing. In Proc. 21st Symp. Parallelism in Algorithms and Architectures (SPAA), pages 197{203, 2009.

[10] Heiner Ackermann, Simon Fischer, Martin Hoefer, Marcel Schongens, Distributed Algorithms for QoS Load Balancing.In Proc. 19th Symp. Discrete Algorithms (SODA), pages 314{322, 2008.

[11] Petra Berenbrink, Tom Friedetzky, Leslie Ann Goldberg, Paul Goldberg, Zengjian Hu, and RusselMartin. Distributed sel_sh load balancing. SIAM J. Comput., 37(4):1163{1181, 2007.

[12] Petra Berenbrink, Tom Friedetzky, ImanHajirasouliha, and Zengjian Hu. Convergence to equilibria indistributed, sel_sh reallocation processes with weighted tasks. In Proc. 15th European Symposium onAlgorithms (ESA), pages 41{52, 2007.

[13] Petra Berenbrink, Martin Hoefer, and Thomas Sauerwald. Distributed sel_sh load balancing on

networks.In Proc. 22nd Symp.Discrete Algorithms (SODA), 2011.

[14] Benjamin Doerr and Leslie Goldberg. Drift analysis with tail bounds. In Proc. 11th Intl. Conf. ParallelProblem Solving From Nature (PPSN), volume 1, pages 174{183, 2010.

[15] Paul Goldberg. Bounds for the convergence rate of randomized local search in a multiplayer load balancing game. In Proc. 23rd Symp.Principles of Distributed Computing (PODC), pages 131-140,2004.

[16] Piyush Gupta and P. R. Kumar.The capacity of wireless networks. IEEE Trans. Inf. Theory, 46 (2):388-404, 2000.

[17] Bruce Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. Advances Appl. Prob., 13:502-525, 1982.