https://dx.doi.org/10.24001/ijaems.3.3.12 *ISSN: 2454-1311*

The Concept of Load Balancing Server in Secured and Intelligent Network

K. Pendke, A. Kasrekar, A. Sawarkar, T. Yeddeloo, R. Bhusari

[1]Nagpur University, Rajiv Gandhi College of Engineering and Research, Wanadongri Road, India

Abstract— *Hundreds and thousands of data packets are routed every second by computer networks which are complex systems. The data should be routed efficiently to handle large amounts of data in network. A core networking solution which is responsible for distribution of incoming traffic among servers hosting the same content is load balancing. For example, if there are ten servers within a network and two of them are doing 95% of the work, the network is not running very efficiently. If each server was handling about 10% of the traffic, the network would run much faster.Networks get more efficient with the help of Load balancing. The traffic is evenly distributed amongst the network making sure no single device is overwhelmed.When a request is balanced across multiple servers, it prevents any server from becoming a single point of failure. It improves overall availability and responsiveness. To evenly split the traffic load among several different servers web servers; often use load balancing.Load balancing requires hardware or software that divides incoming traffic amongst the available serverseither it is done on a local network or a large web server. High amount of traffic is received by a network that have one server dedicated to balance the load among other servers and devices in the network. This server is often known as load balancer. Load balancing is used by clusters or multiple computers that work together, to spread out processing jobs among the available systems.*

Keywords— Load balancing, Intelligent server, Parallel Hash Join (LBJ) algorithm.

I. INTRODUCTION

These days' people understand a lot more about high availability, best practices and pattern. Over the past years it has been observed that there are problems in handling the load due to declined cost of hardware, advancement in common technology, explosive growth of internet and need of solving large scale problems. The question arises that what basically load is? Load is the amount of request for data performed by the user over the server.There is anovergrowing need for data transfer on move. This drives an urgent need to resolve Heavy overhead consumption in routing issues. The resources in distributed computing systems should be allocated to computational tasks. So as to optimize some system performance parameters that guarantee a user specified level of system performance.The load balancing is concerned were resource allocation policies to assign tasks to computing nodes. The primary function of a load balancer is to keep your application on running with no down time. It is invaluable tool for system architecture and it has the benefit of being pretty simple to understand.A local balancer is simply a device that sends internet traffic to a group of application that serves with an even load distribution. Load balancing improves the distribution of workloads across multiple computing resources. Load balancing is dividing theamount of work that a computer has to do between two or more computes so that more work gets done in the same amount of time and in general all uses get served faster. Load balancing can be implemented with hardware or software or both.

Server-side load balancer is usually a software program that is listening on the port where external clients connect to access services. The load balancer forwards requests to one of the "backend" servers, which usually replies to the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting back-end servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports.Some load balancers provide a mechanism for doing something special in the event that all backend servers are unavailable. This might include forwarding to a backup load balancer, or displaying a message regarding the outage.It is also important that the load balancer itself does not become a single point of failure. Usually load balancers are implemented in high-availability pairs which may also replicate session persistence data if required by the specific application.[5]

https://dx.doi.org/10.24001/ijaems.3.3.12 *ISSN: 2454-1311*

Numerous scheduling algorithms are used by load balancers to determine which back-end server to send a request to. Simple algorithms include random choice or round robinmore sophisticated load balancers may take additional factors into account, such as a server's reported load, least response times, up/down status (determined by a monitoring poll of some kind), number of active connections, geographic location, capabilities, or how much traffic it has recently been assigned.An important issue when operating a load-balanced service is how to handle information that must be kept across the multiple requests in a user's session. If the backend server has the information that is stored locally on one backend server then subsequent requests going to different backend servers would not be able to find it. This stored information can be recomputed, in case load-balancing a request to a different backend server just introduces a performance issue.

II. EXISTING SYSTEM

The existing system worked on load balancing by using a Dynamic Load Balancing Parallel Hash Join (LBJ) algorithm. This algorithm is used for shared-nothing hypercubecomputers.

Basically in this system, fragments get partitioned from a relation who is initially horizontal. These fragments are disjoint and are distributed among allthe processors. When a join operation is performed, each processor firsthashes its local portion of the relations into subbuckets. Theselocal subbuckets are then stored back into the local disks.After the subuckets are formed, the distribution ofthe data in each bucket is computed, and the matchingsubbuckets are then collected to their designated processor to form the corresponding buckets by the use of a largest subbucketretaining strategy to reduce communication overhead. The gathering of the subbuckets is based on the size of buckets to ensure balanced work load for each of the processors during the join phase.Each processing node (PN) on the hypercube is directlyconnected to each of its n neighbors, where n = log, Nand N is the number of processors. In this structure, eachprocessor is given a unique n-bit binary address. Each bit givesthe coordinate of that processor in the n-dimensional space.Thus, the neighbors of the processors are those processorswhose coordinates differ by exactly one bit position. Two PN's are named paired nodes and the communication step is called a pairing communication step.Overall the existing system consists of the system environment which is a shared-nothing hypercube multiprocessor computer and therelations are horizontally partitioned into disjoint subsets of the tuples and distributed across all the PN's.

III. PROPOSED WORK

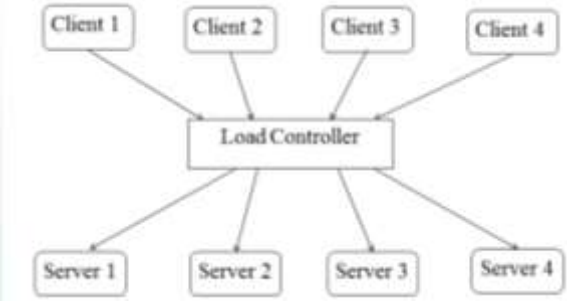

Fig.3.1: Proposed architecture

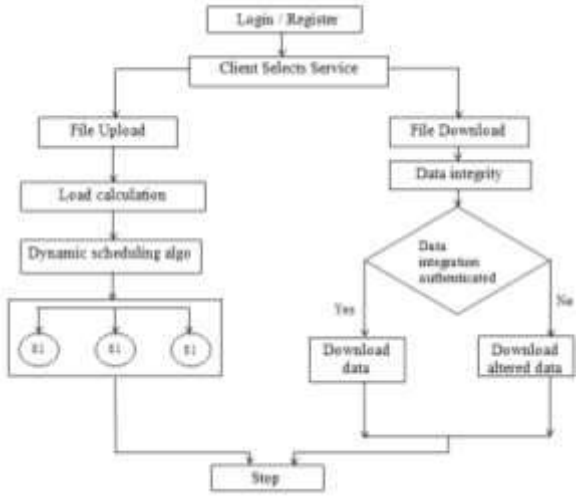

Fig.3.2: System architecture

3.3 Dynamic scheduling algorithm.

Dynamic scheduling, as its name implies, is a method in which the hardware determines which instructions to execute, as opposed to a statically scheduled machine, in which the compiler determines the order of execution. In essence, the processor is executing instructions out of order. Dynamic scheduling is akin to a data flow machine, in which instructions don't execute based on the order in which they appear, but rather on the availability of the source operands. Of course, a real processor also has to take into account the limited amount of resources available. Thus instructions execute based on the availability of the source operands as well as the availability of the requested functional units. Dynamically scheduled machines can take advantage of parallelism which would not be visible at compile time. Dynamic priority scheduling is a type of scheduling algorithm in which the priorities are calculated during the execution of the system. The goal of dynamic priority scheduling is to adapt to dynamically changing progress and form an optimal configuration in self-sustained manner. It can be very hard to produce well-

https://dx.doi.org/10.24001/ijaems.3.3.12 *ISSN: 2454-1311*

defined policies to achieve the goal depending on the difficulty of a given problem.

In parallel computation, the scheduling and mapping tasks is considered the most critical problem which needs High Performance Computing (HPC) to solve it by breaking the problem into subtasks and working on those subtasks at the same time. The application sub tasks are assigned to underline machines and ordered for execution according to its proceeding to grantee efficient use of available resources such as minimize execution time and satisfy load balance between processors of the underline machine. The dynamic scheduling algorithms allocate/reallocate resources at run time.

3.4 Work done

Before using the load balancing software, the user will have to first register on it. For this they will need to provide all basic information and submit it. Once registration is done, user get registered on website and now they can login using user name and password entered while registration. This is GUI which is visible to the user. While registration, various validation are applied such as if male entered by user is incorrect then registration form will not be accepted and a message will be displayed that is entered mail is invalid. Connectivity is established between front end and back end using connection string so that the data entered by user will get stored in SQL table. Whenever the file uploaded on any of the sub server, then the count of that sub server will get incremented by one. Likewise we will be able to find the load that has been generated on each of the sub server. Once the load is known for all sub servers, the task is assigned to the server having lesser load. For assigning the task we are using dynamic scheduling algorithm. The dynamic scheduling algorithm work as follows:

1. Input task
2. Generate load list
3. CPU load in load list
4. Find minimum load in load list.
5. Assign task to respective server
6. Update the server to server load

While downloading a file, a file download window will be displayed. Here all the sub server containing the uploaded file will be displayed. The user can download the required file from the respective server where the file was uploaded previously.Data integrity is also maintained. If there is any modification of data, changes in data or any update in data the user will be notified with the size of file, date and time. The user can download the modified version of the file if data integration is done or else the original file can be downloaded.

IV. LITERATURE SURVEY

In this paper, Hisao Kameda studied that job processing can be shared by distributed computer system in the event of overloads. This paper has considered two optimal policies which dynamic and static. In these policies, job scheduling is defined in order to minimize the system mean response time. The main aim of this paper is to determine till what extent the optimal dynamic load balancing policies performs better than the static one by an exhaustive numerical investigation on a model for which both the policies are analytically studied. The overall mean response time is minimized by both the policies. A comparison of the performance of dynamic and static policies has been studied in more sophisticated models. In these models there are overheads which are considered, but the truly optimal dynamic policy is not accurately obtained. Load balancing policies have a purpose to improve the performance by distributing the work load among more than one node. Dynamic load balancing policies reacts to the current system state and static load balancing policy depends on the average behavior of the system[1].

In this paper, Katherine M. Bawngartner and Benjamin W. Wah have explained a load balancing strategy for a computer system which is connected by multi-access broadcast network. The strategy that has been used in this paper is to implement an efficient search technique for finding stations with maximum and minimum work load by using the existing broadcast capability of these networks. A global scheduling strategy is studied in this paper, whose objective is to find the maximum attainable performance of such kind of strategy. A bus connection is common in many local DCSs and workstations, so therefore a system with a broadcast bus is chosen. This paper describes the proposed strategy, the motivation of its development, its implementation and its performance. The second part of this paper describes a scheduling strategy. In this strategy the workload is transferred from the loaded processor to the minimally loaded processor. The detail studied of the frequency of operation the technique for isolating the processors, a characterization of load distribution and performance based on simulations is done. The overview of the scheduling problem on a broadcast bus is also under seed in this paper. A broadcast bus is connected by a multiple identical processor. Each processor can have arrivals to the system or from the bus. Independent task can be model by the jobs. The results are returned to the originating processor if and only if jobs are migrated to a processor across the bus after the execution is completed. The queue, at each processor is finite: only a limited number of jobs may be waiting for execution [2].

https://dx.doi.org/10.24001/ijaems.3.3.12 *ISSN: 2454-1311*

In this paper, Ming-Chang Huang & hossein Hosseini and K. Vairavan have recommended a load balancing protocol called fuzzy logic receiver, which uses fuzzy logic control for load transfer. The nodes are located for load transfer dynamically by a logical hierarchically structure on which the protocol is based. The performance of the protocol is better than the BID algorithm for the data is studied from the simulation results shown in this paper and it also showed that it is better than the Non-fuzzy logic receiver-initiated protocol when utilization is less than 0.7 the protocol has the same system performance Non-fuzzy logic receiver-initiated protocol when system utilization is higher than 0.7. The outcomes support the system architecture and the protocol that are used in this paper results in better system performance as compared to other conventional load balancing methods [3].

In this paper, Kicn A. Hua (member of IEEE) and Jeffrey X. W. Suthey have put forward a Dynamic Load Balancing Parallel Hash Join (LBJ) algorithm for shared-nothing hypercube computers. In this firstly a relation is horizontally partitioned into disjoint fragments and then it is distributed among all the processors. When a join operation is been performed, each processor first hashes its local portion of the relations into the sub-buckets. These sub-buckets are then stored into the local disks. After the formation of sub-buckets, the data that has been distributed in each bucket can be computed, and the matching sub-buckets are then received to their designated processor to form the related buckets using the largest sub-bucket retaining strategy in order to reduce communication overheads. The collection of sub-buckets depends on the size of the buckets that ensures balanced workload for each of the processors during the join phase. In this paper, author stated a load balancing scheme for the hypercube interconnection topology. The technique that has been proposed is based on a partition tuning strategy that takes the advantage of the already balanced portion of the data space. To keep the cost of communication low the data transmission can be confined to the overflow areas [4].

In this paper, Anuradha Sharma and Seema Verma examined that Grid computing is an extension of distributed computing that includes coordinating and sharing of computational power, storage of data and network resources across dynamic and geographically dispersed organization. To unite the power of widely distributed resources, and provide non-trivial services to users is one of the motivations of Grid computing that has been studied in this paper. An efficient load balancing system is one of the essential part of the Grid to achieve this goal. The rapid growth in use of computer has lead in the increased number of applications is also surveyed in this paper. These applications make use of the shared hardware and software resources (e.g. memory processor, files etc.) which increased the amount of submitted job across internet. They found a solution to the problem that can be solved if the application is distributed across different computers in such a way that the job response time and the overhead on a single computer are reduced. Proper distribution of applications across different available resources is termed as load balancing. The paper describes that some algorithms do not specify memory requirement of the jobs while giving the jobs to particular resources and some algorithms do not consider the cost for communication that cannot be neglected. The requirement of memory is very important in order to complete the execution of jobs at the selected resources within the time limit in realizing a real Grid system [5].

In this paper, R. Vaishnavi, Jose Anand and R. Janarthanan have studied the drawbacks post by the previous Grid systems on the data grids storage services, so they have proposed the cryptographic protocol that is able to fulfil the requirements for storage security related with a generic desktop data grid scenario. This paper has proposed a protocol that uses three basic mechanisms to complete its goal: (A) symmetric cryptography and hashing, (B) an information dispersal algorithm and (C) fragmentation factor quantitative metric. The results that are received define a strong relationship between the assurance of the data at rest, the FF of the volunteer storage clients and the number of fragments required to rebuild the original. This paper defines the enhancement of the overall security of the desktop data grid, using a protocol that uses three basic techniques to cope with untrusted volunteer nodes participating in these environments. The use of cryptography and information dispersal algorithms is not new for securing data and metadata in distributed systems but the main contribution of this research paper consist of enhancing the mechanisms with the adoption of the fragmentation factor representing the guarantees offered by the volunteer storage client to the grid user's storage data and the sensitivity of the data [6].

In this paper, a solution is proposed for dynamic load balancing by Javad Mohammadzadeh, M- Hossein Moeinzadeh, Sarah Sharifian-R, Leila Mahdavi. As the nature of the problem is NP-hard the desired methods are heuristic. A Simple Scheduling Method, Round Robin and Genetic Algorithm are described as previous methods of this problem so as to improve the results, a new modification of Genetic Algorithm is explained. The problem proposed in this paper is to schedule a number of

https://dx.doi.org/10.24001/ijaems.3.3.12 *ISSN: 2454-1311*

non-uniform tasks to a number of non-uniform processors in order to minimize max-span and maximize processor utilization. The non-uniform property of tasks and processors means difference in size of tasks and computational power of processors [7].

V. CONCLUSION

Previously it was observed that there were many problems arising regarding the balancing of the load amongst the computational resource.In our project we worked on this project and came with a conclusion of developing a load balancer. The main goal of the Load Balancing algorithm is to distribute the jobs among processors to maximize throughput, minimize total turnaround of jobs, to match the application need with the available computing resources, maintain stability, and resource utilization. Hence the load balancer handles all the processes that will be handled same time. Proper decision making will be done by toolkit for load management and all the resources will be efficiently used.This paper presents the allocation of tasks to the computing nodes so that nodes evenly loaded.

VI. SCREENSHOTS

REFERENCES

[1] Hisao Kameda El-Zoghdy Said Fathyt Inhwan Ryut Jie Lis University of Tsukuba, A Performance Comparison of Dynamic vs. Static LoadBalancing Policies in a Mainframe - Personal ComputerNetwork Model, Sydney, Australia December, 2000

[2] Katherine M.Bawngartner and Benjamin W.Wah, A GLOBAL LOAD BALANCING STRATEGY FOR ADISTRIBUTEDCOMPUTER SYSTEM, Department ofElectrical and Computer Engineering and the Coordinated Sciences Laboratory University of Illinois at Urbana-Champaign 1101W.Springfield Avenue Urbana

[3] Ming-Chang Huang, S. Hossein Hosseini and K. Vairavan, A Receiver-Initiated Load Balancing Method In Computer Networks Using Fuzzy Logic Control, Department of Electrical Engineering and Computer Science University of Wisconsin – Milwaukee

[4] Kicn A. Hua. and Jeffrey X. W. Su, Dynamic Load Balancing in Very Large Shared-Nothing Hypercube Database Computers, 12 December 1993

[5] Anuradha Sharma and Seema Verma, A SURVEY REPORT ON LOAD BALANCING ALGORITHM IN GRID COMPUTING ENVIRONMENT

[6] R.Vaishnavi, Jose Anand and R.Janarthanan, Efficient Security for Desktop Data Grid Using Cryptographic Protocol, 4th-6th June 2009

[7] Javad Mohammadzadeh, M-Hossein Moeinzadeh, Sarah Sharifian-R, and Leila Mahdavi, Scheduling Dynamic Load-Balancing in Parallel and Distributed Computers using Modified Genetic Algorithm with Time Dependent Fitness Function.

[8] Raja Naeem Akram and Ryan K. L. Ko, Unified Model for Data Security – A Position Paper, 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications.

[9] Archana B. Saxena and Deepti Sharma, analysis of threshold based centralized load balancing policy for heterogeneous machines, International Journal of Advanced Information Technology (IJAIT) Vol. 1, No.5, October 2011.

[10] Dr. BhavaniThuraisingham, Data Mining for Security Applications.

[11] Md. Shahjahan Kabir ,Kh. Mohaimenul Kabir and Dr. Rabiul Islam, process of load balancing in cloud computing using genetic algorithm, 2, June 2015