Load–Balancing - The Essential Factor in Communication Network of Distributed System

V.S.Tondre
B.B.Sc.C.,
Amravati, India

Dr.R.V.Dharaskar
M.P.G.I.,
Nanded, India.

Dr.S.S.Sherekar
S.G.B. A.U.,
Amravati, India.

ABSTRACT

The load balancing is vital factor in communication networks of distributed system. The main purpose of this factor is to assign arriving or internally generated tasks or jobs among processing nodes of a distributed system in such a way that provides the highest utilization of the system resources and also maintain the fairness to the jobs.

If the load balancing is not implemented properly, it decreases the QoS of real-time applications. It also hampers the effectiveness, accuracy of the data transmission technique, scheduling and processing time of the technique. So, it is essential to consider the factor that affects the load in the network.

The main requirement of the load balancing in the system is, the two nodes can have direct communication only if their distance is under certain threshold, low bandwidth and nodes are energy constraint. Therefore, it is crucial to balance the number of packets passed by each node. In this paper, the focus is on the techniques that are designed specially to balance the load that exhibit adaptive approach for providing efficient and effective performance in the distributed system.

Keywords

Load balancing, distributed system, routing and scheduling, bandwidth, resource management, stretch factor.

1. INTRODUCTION

In the Internet, usually large volume of data is transferred from source towards the destination. Many times in an overlay multicast system or in wireless communication network, streaming contents are delivered towards destination. Whatever, paths are available that used for the other transmission. While transferring the data over the internet, it uses variety of routing techniques in the variety of the environment with the path diversity for subsequent packets. It uses several protocols that were designed for multigroup networks.

The load-balancing is a crucial factor in routing. This process can be defined by three rules: the location rule, the distribution rule, and the selection rule. The location rule determines which processors will be involved in the balancing operation. Load-balancing domains can be either global or local. A global domain allows the balancing operation to transfer load from one processor to any of the processors in the system, while a local domain only allows balancing operations to be performed within the set of nearest-neighbor processors [1]. The distribution rule determines how to redistribute the workload among processors in the balancing domain. This rule depends on the balancing domain that is determined by the location rule. While, the selection rule decides on whether the load-balancing operation can be performed preemptively or nonpreemptively. The former may be transferred to other processors in the middle of execution while, in the latter, tasks can only be transferred if they are newly created.

This property is used to equally spread the load on the processors and maximize the utilization of the resource in the distributed systems. It also minimizes the task execution time. To achieve the goal, load balancing scheme should be fair while distributing the load between the processor across the network processors. It should be more effective for that constantly update the load information on the each processor [2].

2. THE CRAM OF METHODS FOR LOAD BALANCING

In this section, methods and architecture for load balancing are discussed.

2.1 GA-based Load balancing

A fixed number of tasks, each having a task number and a size, is randomly generated and placed in a task pool from which tasks are assigned to processors. As load-balancing is performed by the centralized GA-based method, the first thing to do is to initialize a population of possible solutions. This can be achieved by using the sliding-window technique.

In static load balancing, all information regarding load balancing decision is previously known in advance. Therefore, all the tasks are allocated during compile time. It will not affected by the state of the system at the time. In dynamic load balancing, all tasks are allocated to the processors dynamically as they arrive. Dynamic load balancing gives better performance compare to the static [2].

2.2 Stretch Factor and Load-Balancing Ratio in Routing

The communication network, among all the routing algorithms, mainly depends on the shortest path routing algorithms and the load balanced routing algorithms. These two types of algorithms are regarded to two different quality measures: stretch factor and load balancing ratio. Stretch factor means the worst case ratio between the length of the path used by the algorithms and the length of the shortest path. Load balancing ratio means the worst case ratio between the maximum load acquired by the algorithm and that of the optimal load balancing routing algorithm [3].

The tread-offs between the two measures is studied in [3], i.e. stretch factor and load balancing ratio for the relations of growth restricted graph. It shows the combinatorial hop on the load balancing ratio for the optimal c-short routing algorithm. Yet, it is NP-hard to compute the set of shortest paths which minimizes the maximum load. The algorithm, presented that find the shortest paths with maximum load within an O(log n) factor of the optimum. For this, the online virtual circuit routing algorithm is applied. It does not deal with the scheduling, only discovers the paths. After selecting the path, one may use any other methods for scheduling.

2.3 Middleware Architecture Integrated in an Application Server

The designed architecture supports dynamic clustering of QoS-aware Application Servers (QaAS's) and load balancing. To meet application SLAs, it requires a careful assessment of the correct amount (and characteristics) of the resources needed.

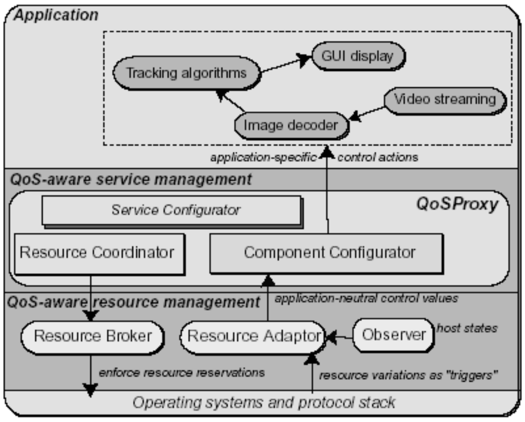

Fig. QoS-aware Middleware Architecture

After analysis of the observations, mainly middleware architecture is responsible for the dynamic configuration, runtime monitoring, and load balancing of a QoS-aware cluster. Therefore, no modifications to these applications are required. It operates transparently to the hosted applications. It consists of the following three main services: Configuration Service, Monitoring Service, and Load Balancing Service. The prototype of architecture has implemented using the open source J2EE application server Jboss. In this architecture, as shown in the fig., the size of the cluster can be changed at runtime, to meet the nonfunctional application requirements of SLA [4].

2.4 Autonomic Resource Management for multiple Spanning tree

Viking [5] is a multi-spanning tree Ethernet architecture, which aspires to address the deficiencies of the original Ethernet architecture. Therefore, commodity enterprise Viking2 is an autonomic network resource management system, which aims to take on a switched Ethernet network's configuration to dynamically measured input traffic loads, therefore to maintain high network utilization efficiency without requiring human intervention.

To maintain high network utilization efficiency without to have need of human involvement, Viking2 adopt a switched Ethernet network's configuration to dynamically measured input traffic loads [6]. Viking2 totally bases on its network reconfiguration decisions on long-term traffic load measurements across the entire network [6].

The Viking has two main features; to maximize the total network resource utilization efficiency, it includes a network-wide load-balancing routing algorithm, which computes paths for node pairs. It has a power of VLAN group mechanism in modern Ethernet switches, which is used to embed multiple spanning trees into a physical mesh network, where VLAN has its own spanning tree. When congestion occurs, the central Viking Manager (VM) retrieves load statistics from VNC's time-to-time and recomputed routes between VNC pairs and VLAN configurations. Congestion may occur when a link's carried load exceeds its capacity. VLAN maintain a real time global traffic matrix. When traffic load fluctuates, it leads to unstable networks. But Viking2 takes a reactive approach in that it reconfigures the network when congestion occurs and it also improve the stability by strives to minimize the disruption when reconfiguring a network to eliminate congestion. Effectiveness of dynamic resource management system enumerated by the number of congested bits, which is defined as the number bits that need to be dropped because link's load exceeds its capacity.

2.5 Mapping Array to the Distributed memory Codes

The data mapping is depends on the program structure, the characteristics of the underlying system, the number of processor, the optimization of compilation system and the problem size. The vital characteristics data movement, parallelism and load balancing are also considered to effectively solve the mapping problem. Particularly, automatic data distribution plays an important role in making parallel system usable. It maps arrays onto the distributed memory nodes of the system according to the array access pattern and parallel computation done in application.

In [7], present a novel framework to automatically determine the data mapping and parallelism. It combines data distribution and dynamic redistribution with parallelism information in a single graph representation: The Communication Parallelism Graph (CPG). It contains information about the data movement and parallelism within phase and possible data movement due to remapping between them. This information is measured in time units, which represent the data movement and costs of computation. It allows the alignment, Distribution and redistribution problem. In CPG, general purpose linear 0-1 integer programming technique, which guarantees the optimality of provided solution. It is also used to solve the minimal cost path problem. It is effective in solving a variety of compilation problem. The cost model is based on profiling information with some parameters of the target systems such as the number of processor, the parallel thread creation overhead and the communication latency and bandwidth. The general data mapping scheme is used to explain the original HPF data mapping and loop parallelization directives.

2.6 System Sensitive Run-Time Management

The solutions of partial differential equations with dynamic adaptive methods are used locally optimal approximations can give highly advantageous ratios for cost and accuracy when compared with static methods. Distributed implementation of these adaptive methods, offer the possibility for the accurate solution of realistic model. It may lead to the interesting challenges in dynamic resource allocation, data distribution and load balancing, communication and coordination and resource management.

In [8] presents the design and assessment of a system sensitive partitioning and load-balancing framework for distributed adaptive grid hierarchies that motivate parallel adaptive mesh-refinement (AMR) techniques for the solution of partial-differential equations. The framework uses system capabilities and current system state to select and tune appropriate distribution parameters (e.g. partitioning granularity, load per processor) to maximize overall application performance. As shown in Fig., in the system sensitive framework first monitors the state of resources that are associated with the different computing nodes. It uses this same information to compute there relative computational capacities. These relative capacities are then used by the system-sensitive petitioner that has been integrated into GrACE (is an approach to distribute AMR grid hierarchy) for dynamic distribution and load-balancing. Resource monitoring tool is used to determine the system characteristics and state at run-time. It gathers the information about the CPU availability, memory usage and link-capacity of each processor. This information is then passed towards the capacity calculator.
After that a relative capacity metric is computed for each processor using a linear model by using the following:

$$C_k=w_pP_k+w_mM_k+w_bB_k$$

Load imbalance is defined as

$$I_k = \frac{|W_k - L_k|}{L_k} \times 100 \quad \%$$

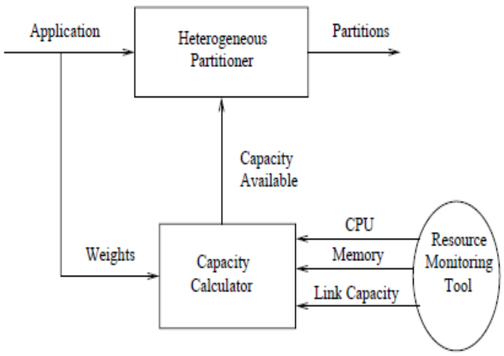

Fig. Block diagram of System Model

3. CRITICAL ANALYSIS OF THE TECHNIQUES

In this section load balancing factor of the techniques are critically analyzed and presented.

3.1 Effectiveness of the GA-based Algorithm

The load balancing using genetic algorithms has been very effective especially in the case of a large number of tasks. In the central scheduler was also effective as it can handle all load-balancing decisions with minimal inter-processor communication. The threshold policy used also provided better performance. It achieves the goals of minimum total completion time and maximum processor utilization [2].

3.2 Properties of Routing Network

In the technique, trade-off between two important qualities of measures of the routing network for the growth restricted network is used. Those properties are load-balancing and stretch factor [3]. It reduces interference in transmission and average numbers of nodes are covered by any unit disk. The average density is more appropriate for a set of wireless nodes with uneven distribution.

3.3 Services of QoS-aware Middleware Architecture

It operates transparently to the hosted applications. It consists of the following three main services: Configuration Service, Monitoring Service, and Load Balancing Service [4]. The prototype of architecture has implemented using the open source J2EE application server Jboss. In this architecture the size of the cluster can be changed at runtime, to meet the nonfunctional application requirements of SLA.

3.4 Performance Viking2 DRM Algorithm

Viking2 is used to remove congestion which is occurred due to long term traffic load changes through dynamic resource management [6]. The effectiveness of Viking2's DRM algorithm is measured by the amount of congested traffic or the total number of bytes which are dropped in each test run.

3.5 Accuracy and Efficiency of the Proactive Approach

The framework is based on a single data structure, it's called as Communication Parallelism Graph (CPG), it integrates all the data movements and parallelism related information into inherent in each of the phase of program plus additional information denoting remapping. [7].

3.6 Advantages of the System Sensitive Partitioning

This scheme reduces the total execution time of the application and the load imbalance as compared to a scheme that does not take the relative capacities of the computing nodes into account. System sensitive partitioning and load balancing framework has the following advantages is it maximizes the overall performance of the application [8].

4. CONLUSION

In this paper, the vital property of the communication network- load balancing is discussed and analyzed. It is seen that, these methods are easily improve the worst case delay, rate threshold, minimizes the processing time, delay jitter and queuing delay without loss of link utilization.

5. REFERENCES

[1] A.E. Kostin, I. Aybay, and Gurcu Oz, "A Randomized Contention-Based Load-Balancing Protocol for a Distributed Multiserver Queuing System", IEEE Transactions On Parallel And Distributed Systems, Vol. 11, No. 12, Pp.1252-1273, December 2000.

[2] Albert Y. Zomaya and Yee-Hwei The, "Observations on Using Genetic Algorithms for Dynamic Load-Balancing", IEEE Transactions On Parallel And Distributed Systems, Vol. 12, No. 9, pp. 899-911, September 2001.

[3] Jie Gao And Li Zhang, "Trade-Offs Between Stretch Factor And Load-Balancing Ratio In Routing On Growth-Restricted Graphs", IEEE Transactions On Parallel And Distributed Systems, Vol. 20, No. 2, Pp. 171-179, February 2009.

[4] G. Lodi, F. Panzieri, D. Rossi, And E. Turrini, "SLA-Driven Clustering Of Qos -Aware Application Servers", IEEE Transactions On Software Engineering, Vol. 33, No. 3, 186-197, March 2007.

[5] S. Sharma, K. Gopalan, S. Nanda, and T. Chiueh. "Viking: A Multi-Spanning-tree Ethernet Architecture for Metropolitan Area and Cluster Networks", In IEEE INFOCOM, 2004.

[6] S. Lin, S. Sharma, and T. Chiueh, "Autonomic Resource Management for Multiple-Spanning-Tree Metro-Ethernet Networks", Sixth International Symposium On Network Computing And Applications (NCA), 2007.

[7] Jordi Garcia, Eduard AyguadeÂ , and JesuÂs Labarta A Framework for Integrating Data Alignment, Distribution, and Redistribution in Distributed Memory Multiprocessors", IEEE Transactions On Parallel And Distributed Systems, Vol. 12, No. 4, Pp. 416-431, April 2001.

[8] Shweta Sinha and Manish Parashar, "System Sensitive Runtime Management of Adaptive Applications", Conference: International Parallel and Distributed Processing Symposium / International Parallel Processing Symposium – IPDPS (IPPS), (2001) (citation:7)