International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 06 | June 2022 www.irjet.net p-ISSN: 2395-0072

A SURVEY ON STATIC AND DYNAMIC LOAD BALANCING ALGORITHMS FOR DISTRIBUTED MULTICORE SYSTEM

Dr.G.Muneeswari[1]

[1]*Professor, School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India*

Abstract - *In today's world with large number of super computers, it's essential to distribute the load efficiently among different cores among the multicore processing with proper load balancing algorithms. The core idea is that none of the multicore elements should be kept in the idle state. In-order to implement the efficient algorithm, the better understanding of the distributed system must be dynamically known which actually a challenging task is. Modern framework for load balancing methodologies are integrated with the application itself which can balance the load based on the application level knowledge. But the real scenario lies in the work of processing elements capabilities. So, these algorithms may lead to incorrect simulations. This paper focusses on the comprehensive study of existing load balancing algorithms implemented on the various distributed environments.*

Key Words: Load balancing, static, dynamic, resource utilization, Multicore, distributed system

1. INTRODUCTION

The distributed multicore system can be homogeneous or heterogeneous with different architectures. A process allocated to the multicore processing elements can be pre-emptive or non-pre-emptive. A proper and efficient load balancing algorithm must be incorporated mostly in the case of homogeneous system for better resource utilization. In the traditional system mostly sender initiated and the receiver initiated algorithms are implemented which may not be useful for the current massive parallel technology. A new novel hybrid algorithms which can combine the feature of static and dynamic allocation is required for keeping all the processing elements in the busy state.

The load can be divided based on the implementation of the algorithm. Ideally, underload, overload thresholds are fixed at the centralized master node depends on the expected number of process arrival and capability of computational elements speed of execution. In this paper, a two variant of static and dynamic algorithms are reviewed for various cloud based multicore environments.

The organization of the paper is given as follows: Section 2 describes the literature review and section 3 explains the performance metrics associated with the load balancing algorithms. Finally section 4 concludes the paper.

2. LITERATURE REVIEW

In this method, the performance (Motwani & Raghavan [1], Zhong Xu & Rong Huang [2] of all the processing elements are calculated before starting any task. This distributed multicore system follows the strategy of master slave processing. The master node allocates the task to various nodes based on the estimated load and arrival time of the tasks to the ready queue. Sometimes the slave nodes will be uded to calculate the work load of the incoming requests. In the McEntire et al [3] algorithm, a designated computational element selects a site for the execution of the new task. Whenever the task is created, a lightly loaded processing element is chosen depending on the complete picture of workload. Depending on the tasteful information, the load balancing algorithm quickly decides upon the load allocation and migration.

Unlike the static kind of algorithms, the dynamic algorithms proposed by Malik[4], Wang & Morris[5] allocates the tasks dynamically when one of the processors becomes under-loaded. In [6], the routing for hybrid wireless network becomes complicated with respect high transmission rate. When the intermediate hops are preoccupied with data then the load balancing algorithm find out a different route based on the estimated data load during data transfer. The load balancing [7] for cloud will be extremely complex as there are multiple nodes requested at the same time for client data access. A novel methodology related to the workload distribution is implemented on the RAID systems for the better throughput and performance.

IOT systems[8] have less memory and less power active devices wherein the real challenge is to find out the load of data transfer in web based IOT application. Data transfer to the web servers are implemented with the help of traditional sender oriented load balancing algorithms. Mohammed et al.,[9] proposed a server consolidation methodology for

incorporating load balancing algorithms in cloud based environments. Cloud based datacenters are taken into consideration with master slave activation and the master node or host is responsible for data transfer for balancing the load in the distributed environment. The overall survey [10] of load balancing algorithm along with the different performance metric comparison has been depicted in the work of Kashyap et al.,Mainly the distributed file system architecture is evaluated for the advanced cloud based systems. Central queue algorithm proposed by William Leinberger [11] based on the working of simple FIFO mechanism. Whenever there is resource request requirement comes to the centralized node which has the queue, it will be inserted at the beginning of the queue and whenever the processor becomes available then the request will be removed and given for execution.

Similarly, in-order to implement the load balancing concept, the task migration can be done in FIFO queue order. In homogeneous system [11], the real challenge predominantly lies in the allocation of requests to the respective servers where the data is present. Resources are placed in a resource pool and the resource manager is responsible for handling the incoming requests. Parallel computing system [12] handles load balancing with the help of master nodes and the slave nodes will be involved only with the computational work. Tremendous amount of massive parallelism is achieved with the help of this novel idea.

3. EVALUATION PAPRAMETERS

The performance of various load balancing algorithms is measured by the following parameters.

Eliminating overload

Whenever there is a overload situation which cannot be handled by the existing algorithms then the load balancing algorithms stops execution and eliminates further acceptance of workload from the clients.

Resource availability

This parameter will be very crucial for the user level satisfaction as the user always wants a seamless execution of tasks for their different requests. Availability always related to the fault tolerance of the distributed multicore systems. Fault tolerance makes the system work even in the case of failure of the system. So for better performance of workload migration fault tolerance is very important.

Accuracy level Identification

Static load balancing algorithms provide more accuracy with respect to the identification of workload as it is predefined. In the dynamic algorithms the speculation of work load sometimes depends on the delays inculcated on the network and leads to incorrect prediction of load.

Single Point of Access Vs Distributed

In the single point load balancing algorithms every load balancing decision as well as task migration will be happening at only one centralized node and the rest of the nodes or sites will be actively participating in the computational work. Ideally in the distributed load balancing algorithm all the nodes will be involved in the load balancing decision making which makes the entire system a fair kind og balancing the work load.

Dependent and Independent Processing

The decision of load migration depends on the type of execution carried on the processing elements on the shared system. If a task is divided into several subtasks and each individual subtasks are carried on different geographically distributed systems then the migration of workload should be carefully done by the load balancing algorithm for the final completion of the original task. For independent tasks, there will not be any complication as they can independently complete the execution.

Hardware and Software Utilization

Resource utilization request from the client can be hardware or software oriented. If more number of utilization comes from the same client and if other client requests are not handled then more adaptable dynamic strategies can be implemented.

4. CONCLUSION

Basically load balancing algorithms are categorized either as a static algorithm or dynamic algorithm. In this paper, we reviewed mostly static load balancing algorithms where it allocates the work to the processing elements probabilistically or deterministically and do not consider the application level runtime scenarios. It is concluded that it is not possible to find out the time of arrival of the process and also speculating the future execution times of different workloads. We also investigated very few dynamic algorithms in which the load can be predicted with respect to the network condition and processing elements capabilities. In modern technology implemented on top of the distributed multicore systems, a more reliable dynamic algorithms can be incorporated for better performance.

REFERENCES

[1] Motwani, R & Raghavan, P 1996, 'Randomized algorithms', ACM Computing Surveys (CSUR), vol.28(1), pp.33-37.

[2] Zhong Xu & Rong Huang 2006, 'Performance Study of Load Balancing Algorithms in Distributed Web Server System', CS213 Parallel and Distributed Processing Project Report.

McEntire Munro D,S & Brown, A,L 2000, 'Evaluating partition selection policies using the CMOS garbage collector', POSG ninth International workshop on Persistent Objects Systems, pp. 104-115.

[3] Malik, S 2000, 'Dynamic Load Balancing in a Network of Workstation', 95.515 Research Report.

[4] Wang, Z & O'Boyle, MF 2009, 'Mapping parallelism to multi-cores: a machine learning based approach' Proc. of the 14th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming.

[5] Yan Yan, Qiang Hu, Douglas M. Blough, "Load-Balanced Routing for Hybrid Fiber/Wireless Backhaul Networks", 2021 IEEE Global Communications Conference (GLOBECOM), pp.1-6, 2021.

[6] N. S. Raghava and Singh Deepti, "Comparative Study on Load Balancing Techniques in Cloud Computing", *Open journal of mobile computing and cloud computing*, vol. 1, no. 1, pp. 18-25, August 2014.

[7] Sagar Vats, Yachana Kumari, Tushar Singh, Sunil Chandra, Sushma Shirke, "Load Balancing in Specific IOT Systems", 2021 International Conference on Intelligent Technologies (CONIT), pp.1-6, 2021.

[8] Mohammed Ala'anzy, Mohamed Othman, "Load Balancing and Server Consolidation in Cloud Computing Environments: A Meta-Study", IEEE Access, vol.7, pp.141868-141887, 2019.

[9] Kashyap Dharmesh and Viradiya Jaydeep, "A Survey Of Various Load Balancing Algorithms in Cloud Computing", *International Journal Of Scientific & Technology Research*, vol. 3, no. 11, pp. 115-119, November 2014.

[10] William Leinberger, George Karypis &Vipin Kumar 2000, 'Load Balancing Across Near-Homogeneous Multi-Resource Servers', 0-7695-0556-2/00, IEEE.

[11] Archer C.J., Mullins T.J, Sidelnik A. and Smith B.E. "Parallel Computing System Using Coordinator and Master Nodes for Load Balancing and Distributing Work". United State Patent, 2010.