Metrics for Dynamic Load Balancing in a Parallel System

[1]Mohd Haroon,[2] Manish Madhav Tripathi,[3] Riyazuddin

[123] Integral University, Lucknow,India

Abstract: **The research depict in this paper spotlight on evaluating metrics for use with the dynamic load balancing of parallel system. In this load balancing approach are based on token and is used in union with Clustered Time Warp (CTW). CTW is an amalgam bringing together protocol, which makes use of a sequential algorithm inside clusters of Load and Time Warp among the clusters. A three different metrics are defined here, and measure their effectiveness in different simulation environments. One metric measures the processor utilization and next one metrics dealings the difference in virtual times between the clusters, while a third is a combination of these two metrics. In this paper have assessment of the execution time, memory using up and the throughput obtained in three simulation environments by each of these metrics and to the results obtained without load balancing, Our grouping of simulation are VLSI simulations, characterized by a huge number of Load and a short computational granularity; distributed network simulations, in which the workload fluctuate spatially over the execution of the simulation; and a pipeline simulation. Characterized by a single trend of message flow.**

Key word: **distributed system, parallel system, and metrics.**

I. INTRODUCTION

the purpose of load balancing in a grid system or parallel system to minimized the response time and also minimized the execution time, maximized the through computing various computing network, this approach further classified in two different category, one is the static load balancing, in this category in coming jobs or data are distributed among all computing nodes before the execution or before the staring the execution process, but in another technique that is dynamic load balancing in this category data or load are assign or periodically assigned to the computing node , once the processing is started then this load balancing technique can manage the load of entire network, during run time load migration is possible in dynamic load balancing, but during run time data migration is not possible in static load balancing, over head of static load balancing with respect to dynamic load balancing is minimum, and efficiency of load balancing of dynamic load balancing is better than the static load balancing, From the optimization point of view , by the load balancing the objective of load balancing can achieved, mean minimized the response time to all the incoming jobs and maximized the through put of the computing nodes, in this case individual jobs can be optimized by this approach, or group of jobs can also be minimized by this approach. There are studies on static load balancing that provides the system optimal solution, in such case some jobs experience higher response time, and some job experience lower response time, this thing depends on the length of the job. Few studies exist on static load balancing that provides individual optimal solution based on game-theoretic solutions. Competitive equilibrium approach for achieving both system optimal efficiency and individual optimality is proposed in However, it does not take into account run-time behavior. Several dynamic load balancing algorithms are proposed by many researchers, but the competitive equilibrium approach are used in system optimization, or jobs optimization, in both purpose competitive equilibrium approach are used, comparative equilibrium approach re used full in dynamic load balancing approach.

II. SYSTEM ARCHITECTURE

We consider a grid system having n computing nodes , and all computing nodes are used by n number of user concurrent or parallel, in both modes all the user can share the recourse, we suppose all the job arrived at computing nodes by the rate βi ad total jobs arrive at grid system are $\sum_{i=0}^{n} \beta i$. All the incoming jobs in a system are same.

If suppose k jobs arrived at any computing node A by the communication channel, some job from the k can be calculated at the computing node A, and rest of the job may be transferred to another available computing node in a grid system by the communication channel.

Modeling each node as an M/M/1 queuing system, the expected node delay at node A is as follows.

Delay= 1/ μi –βi where μi is the service rate of computing nodes A and βi is the load of computing node A.

Let us suppose that the expected communication delay among two computing nodes are independent the computing node architecture, but the communication delay can be depend on the communication channel in between computing nodes A to computing nodes B.

Examples of such network are the local area network, in which the communication delay is depending only the communication channel not for computing machine.

Therefore, over all response time of user j job is the sum of expected node delay at each node i and expected communication delay.

A. Required no application changed:

During the execution of the job ,computing nodes cannot modify the executed job, if the job are parallel, sequential or batched types , any types of jobs cannot be modify at run time of the computing machine.

B. Transparent

The jobs can be executed at local machine or remote machine is must be transparent, result of the job cannot be affected at the execution of the jobs, some delay must be included in execution time, but the result cannot be affected.

III. MATHEMATICAL MODEL

In this model the mathematical model of the computing system has given, in mathematical model the total load of the system can be calculated on the basis of individual load of the machine, like in any network suppose there is n computing machine, then total load can be calculated by the sum of load of first machine and load of second machine and so on

Total load= (load of M1+load of M2+….)

Now the system load can be calculated by the load of central processing unit and memory capacity, movement of data inside the data bus, several parameter can be depend on system load

Movement, the memory use and the fact whether the computer is on or off

Now the load of any computing machine is calculated by the given formula

"Li (t) = Alive (t) · (k · Processor (t) + (1 − k) ·Memory (t))",

IV. METRICS AND ALGORITHM

In this section we present the details of our load balancing algorithm and its associated metrics beside with a discussion of significant devise issues.

A. Metrics

In the event of dynamic load handling a several metrics is necessitate to uncover the system load s well as is also control the reassignment on the jobs and migration on the jobs, Incomparably, the metric ought not only be basic and fast to be able to multiply, but in addition effective. In this paper three different metrics usually are discuss first an example may be processor utilization, brand advanced simulation velocity, and combination involving both.

If various (simulation) processes usually are interconnected, a disagreement of their relevant virtual periods can effect in an increase in the volume of messages arriving during the past, and cause rollbacks. Each time a process is folded back from moment ti to moment tj, all work performed throughout on this occasion period is removed. System resources used through the corresponding real time interval could have been industriously employed by simply other processes.
Controlling the rate where processes advance their own corresponding virtual periods will curtail the difference between your virtual clocks, and as significance, reduce the volume of rollbacks which occur from the simulation. The virtual time of an processor is understood to be the minimum virtual time of all the processes residing in that processor. A processor with no dealings to course of action sets its virtual time to infinity.

For a method simulated in the genuine time interval (t start off, t end), the Processor Progress Simulation Rate (PASR) specifies the rate involving advance in virtual time relative to real time. Let t1 and t2 be two real-time values, with t2> t1. Define STt because the simulation time at real-time t. Let ΔST represent the change from the simulation time in the period interval (t1, t2);

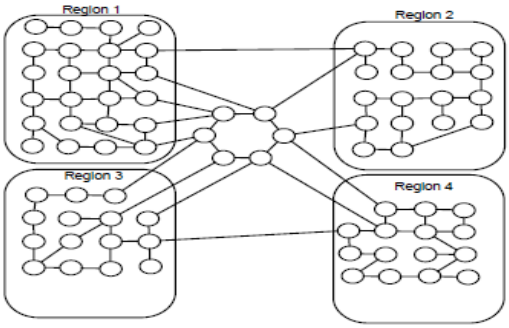
Δ (ST) t1 t2 =ST t2-ST t1

The PASS is understood to be: Δ (ST) t1 t2/t2-t1

A processor using a PASS higher than average is prone to being rolled rear, because it is prior to other processors throughout virtual time. If at all slowed down, the frequency with which it's rolled back by simply other processors could decrease. A message meters sent from P1 to be able to P2 will force P2 to roll to a virtual time before vt1, since the actual timestamp of meters is vt1. P2 then has to cancel all the previous work done from the (t1, t2) real-time interval. Hence, moving some load from processors along with high PASS to be able to others with reduced PASS should quicken the slow processors and decelerate the fast ones.
Number involving researchers feel that it is advisable to maximize the available parallelism from the system by keeping processor utilization often possible. For devices where no any priori estimates involving load distribution usually are possible, only actual plan execution can reveal just how much work has been recently assigned to specific processors.
Let us specify effective utilization because the proportion of work had done by way of a processor which seriously isn't rolled back. However, it is impossible for just a processor to determine the effective utilization with a given point from the simulation since it might rollback later and cancel each of the work that continues to be done. In an estimate on the effective utilization is used for load computation. Consequently we utilize the processor utilization (PU), looked as the ratio on the processor's computation moment (in seconds) between t1 and t2 to be able to t2-t1;

PU= computational time in (t1, t2)/t2-t1

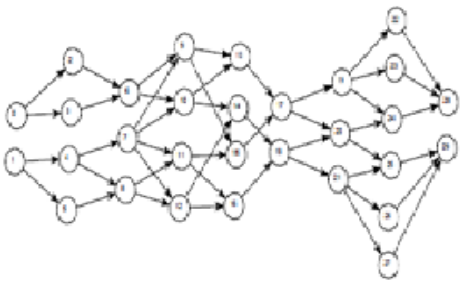
"Processor utilization allows for the point that messages in the device might be involving different size, and might require different service periods. It also accounts for the point that two processors may advance their virtual clocks through the same value, even if the computation moment is different".
A variety of the two metrics, PU/PASR, was also tested in your experiments. The combination was that will increase the utilization of the processors, even though maintaining.



Maximum advance simulation rate and minimizing the number rollback.

B. Pipeline Model

An additional model which was simulated is often a manufacturing pipeline. The model contains thirty processes, as well as two sinks and two sources, fixed in nine stages. Each process is often a cluster of 625 rational processes connected in the mesh topology. Clusters with the same stage were mapped to the same Processor. Messages within the system flow from sources to the sinks, following different paths. At each and every stage, the message will be served and forwarded to another stage, until it reaches the sink, where it leaves the machine. The service period distribution is deterministic plus the routing decision in each stage is governed with a uniform distribution. The pipeline model exhibits a lot of rollbacks which are due to messages starting with the same source, next different paths, and coming to the same processor in the (possibly) different order from the one in which they were generated.



C. Distributed Network model

The final model is a distributed communication model (figure 12). Two kinds of experiments were conducted on the model. In the first experiment, messages are homogeneously dispersed on the network. The second experiment representation a national communication network divided into four regions. In this model, we experimented with the rejoinder of dynamic load balancing to a continuous change of loads on the Processors. During the course of the simulation, messages were concentrated on different regions, one region at a time. For instance, at one point messages were concentrated in region 1, and regions 2, 3 and 4 were lightly loaded. After a period of time, region 2 became saturated with messages, and regions 1, 3 and 4 were lightly loaded. The simulation runs on 10 processors, with 7-8 nodes mapped to each processor. Inter processor communication was minimized by mapping the connected nodes to the same processor. On each node a message is served and with a probability of 30%, is forwarded to a uniformly selected neighbor. Nodes have service times governed by exponential distributions (with different means).

V. CONCLUSION

On this paper, we evaluated your performance of about three metrics for apply while using dynamic load controlling of parallel technique. The metrics were used by way of a token depended active load balancing algorithm which has been implemented in partnership of Clustered Moment Warp. Clustered Moment Warp, as your name entail, records into clusters, and relies on a sequential algorithm in clusters and Moment Warp linking groups.

In order to measure weight on the processors many of us defined three metrics model utilization (PU), model advance simulation rate (PASS), and a combination of these metrics. To weigh the performance of your algorithm with every one of the metrics, several products were simulated reasoning level VLSI products, an assembly pipeline model plus a distributed communication circle model. Each of these kinds of models was selected because of their diverse characteristics the VLSI simulations because of a large number of Load, low computational granularity in addition to paucity of active Load during the course of any simulation; the distributed network simulation with the spatial variation with the workload during the course of the simulation; and the pipeline simulation with the uni directional nature of message stream. Experiments were completed on the BBN Butterfly GP1000, some sort of 32 nodes sent out memory multiprocessor. The particular simulation time, recollection. Consumption and efficient throughput were calculated. The effective throughput is the number of no rolled back messages in the system per system time. Results obtained using every one of the metrics was when compared to those obtained without load balancing

REFERENCES

[1]. Said Fathy El-Zoghdy. “A Load balancing Policy for Heterogeneous Computational Grids”, Vol. 2, No. 5, 2011”

[2]. S. Xian-He, W. Ming, GHS: “A performance system of Grid computing”, in: Proceedings of the 19th IEEE International Symposium on Parallel and Distributed Processing, 4–8 April 2003.

[3]. X. Tang and S. T. Chanson. “Optimizing static job scheduling in a network of heterogeneous computers”. In Proc. of the Intl. Conf. on Parallel Processing, pages 373–382, August 2000.

[4]. Mohd Kalamuddin Ahmad, Mohd Husain, “Required Delay of Packet Transfer Model For Embedded Interconnection Network”, International Journal of Engineering Research, vol 2, issue 1,jan 2013.

[5]. Kalamuddin Ahmad , A.A. Zilli Mohd. Mohd. Husain, “A Statistical Analysis And Comparative Study of Embedded Hypercube”, International Journal of Computer Applications, Volume 103, Oct 2014.

[6]. Mohammad Haroon, Mohammad Husain, “Analysis of a Dynamic Load Balancing in Multiprocessor System”, International Journal of Computer Science engineering and Information Technology Research, Volume 3, March 2013.

[7]. Mohammad Haroon, Mohammad Husain, "Different Scheduling Policy For Dynamic Load Balancing in Distributed System", 3rd international conference TMU Moradabad.

[8]. Mohammad Haroon, Mohammad Husain, "Different Types of Systems Model For Dynamic Load Balancing", IJERT, Volume 2, Issue 3, 2013.

[9]. Mohammad Haroon, Mohammad Husain, "Different Policies For Dynamic Load Balancing", International Journal of Engineering Research And Technology, Volume 1, issue 10, 2012.

[10]. Mohd Haroon Ashwani Singh, Mohd Arif, "Routing Misbehabiour In Mobile Ad Hoc Network", IJEMR, Volume 4, Issue 5 ,October 2014.

[11]. Abdul Muttalib Khan, Mohd. Haroon Khan, Dr.Shish Ahmad, "Security In Cloud By Diffie Hellman Protocol",International Journal Of Engineering And Innovative Technology(IJEIT), Volume 4 , Issue 5 , November 2014.

[12]. AM Khan, Mohd Haroon, S Ahmad, "Security in Cloud by Diffie Hellman Protocol", international Journal of Engineering and Innovative Technology(IJEIT)Volume 4,Issue 5, November2014.

[13]. afsaruddin mohd haroon, riyazuddine, mohd shahid, "Adjacent Selection Method for Load Balancing in Distributed Network by Artificial Intelligence", International Journal Of Advanced Research In Electrical, Electronics And Instrumentation Engineering, 2015/8/20.