Measures used for Performance Analysis of Scheduling and Routing in Distributed Systems

V. S. Tondre
Brijlal Biyani Science College, Amravati

V. M. Thakare
Professor & Head,
P.G. Department of Computer Science
SGB Amravati University, Amravati

R.V. Dharaskar
Director, M.P.G.I., Nanded.
rvdharaskar@rediffmail.com

S.S. Sherekar
Associate Professor
P.G. Department of Computer Science
SGB Amravati University, Amravati

ABSTRACT

A most challenging problems faced by the researchers and developers of the distributed real time system is what types of measures and requirements are considered to measure the performance of the new devised system for scheduling and routing.

This paper presents, the various measures required to count the performance of the system. There are large number of parameters needed to count the performance of the scheduling and routing. This study provides the complete analysis of the performance of the system and how to balance the various aspects to have the better results.

1. Introduction

The needs of the networked applications, protocols and distributed resource sharing; bring a strong inspiration for the research in large-scale network across the multiple domains and multiple technology networks and transport network. The growing number of newly emerging applications: such as teleconferencing, distance learning, IPTV, etc. [1] on the Internet has created an increasing need for efficient multicast traffic support.

Now days, many new applications required large bandwidth over relatively short period of time, small threshold, minimum delay and packet loss, etc,. Therefore, on the Internet, traffic problems arise in the present scenario of ICT [2]. So, the researcher receives increasing research attention to develop new dynamic circuits, protocols, and service level agreements, Scheduling and Routing techniques, and so on. These problems can be solved through many different ways.

The paper is focused on the different dynamic scheduling and routing measures to test the performance of the optimal throughput, turn around time, response time in communication technology.

2. Requirement of the Scheduling and Routing

One of the basic requirements of the Scheduling and Routing is to maintain and increase the quality of service of the system/application when employing the concept of adaptive and dynamic. Along with, there are other requirements that are necessary for specific applications.

In this paper, it considered the most demanding requirements so as to obtain the better results in terms of better CPU utilization, minimizing the CPU idle time, minimize the waiting time, minimize the interference time and estimate the memory demands and required time. In recent time, most of the dynamic resource management techniques including scheduling, routing and monitoring requires more advanced method. Accordingly, the performance is also measured. In the following section discussed the requirement factors of the new techniques and measures for the performance analysis.

2.1 Resource Utilization

The most important requirement of the task scheduling is the resource utilization. In the proposed system, it considered the resource is a processor. Here, find out the compilation time of the processor by using the current load of the processor and new assigned load to processor. As after knowing the compilation time of the processor, the next step is to find out the estimated utilization, actual processor utilization and average processor utilization [3].

Total completion times of each processor

Total completion times of each processor can be calculated using the following formula:

P_i (completion time) = current load of P_i + new load assigned to P_i

where, P_i is a processor and i=0,1,3,..n.

Processor Utilization

The net estimated processor utilization during the k^{th} sampling period is [4]

$$E(k) = \sum_{i=1}^{n} c_i r_i (k)$$

Where, c_i -is the estimated execution time

$r_i(k)$ -is the invocation rate of end-to-end task T_i

The actual processor utilization at runtime U(k) is

$$U(k) = G_p(k)E(k)$$

where, $G_p(k)$ is the processor utilization ratio.

Average Utilization of Processor

The next factor of the task scheduling is the average processor utilization. It is the important factor. The high average processor utilization implies that the load is well balanced across all the processors. The total execution time is reduced by keeping the processors highly utilized [3].

The expected utilization of each processor is based on the given task assignment. It is must be calculated. It is

Proceedings published by International Journal of Computer Applications® (IJCA)ISSN: 0975 - 8887

found by dividing the task completion times of each processor by the maxspan value. The utilization of each processor is:

$$P_i(utilization) = \frac{P_i(compilation\ time)}{maxspan}$$

Then, dividing the sum of the processors utilization by the total number of processors will give the average processor utilization (APU). The average processor utilization is reduces the probability of processors being idle for a long time.

Combined Fitness Function

The fitness function included the three main objectives that are maxspan, average utilization and number of acceptable processor queues[3]. It is given by the following equation:

$$fitness = \frac{1}{maxspan} \times average\ utilization \times \frac{acceptable\ queues}{processors}$$

It is the important factor. The high average processor utilization implies that the load is well balanced across all the processors. The total execution time is reduced by keeping the processors highly utilized.

2.2 QoS for Processor Utilization

The second requirement is the QoS. It is very complex problem for the processor utilization. In the proposed system, it provides the guaranteed service for the processor utilization. In the proposed system availability, bandwidths, delay, jitter and reliability are considered for the guaranteed service.

One of the requirements for QoS is a bandwidth. According to the task requirement the bandwidth is continuously varying. Therefore, it is necessary to estimate the available bandwidth, set it and then transfer the task for the execution towards the processor. In the proposed system it estimates the available bandwidth by using the following formula.

2.2.1 Available Bandwidth Estimation

In our algorithm, CPU continuously processing the jobs, so, periodically it sends completed jobs towards the destination node. For each arrived packets, the receiver checks FPF to determine the first packet of the packet-pairs. Then receiver can estimate the available bandwidth as:

$$R_{pp} = \frac{8\,M}{t_{gap}}$$

where, -M is the size of the packet (in bytes).

- t_{gap} is the inter-arrival time (in seconds) of packets.

2.2.2 Bandwidth allocation

During each sampling time period of the processor controller, the bandwidth allocator computes a desirable bandwidth allocation for each task T_i. The wireless network bandwidth allocation to each task T_i is recomputed by the bandwidth allocator if the presence of an object of interest was detected by any of the target tracking subtasks or a variation in the available bandwidth was detected during the previous sampling period [4]. For such types of tasks, bandwidth is allocated such that the net bandwidth utilization is below the set-point B^s i.e.

$$\sum_{i=0}^{n} b_i^s(k) \leq B^s C(k)$$

$b_i^s(k)$ is the bandwidth allocation for task T_i during the particular time period of the processor controller. The bandwidth is allocated to each end-to-end task as a function of p(k) and $p_i(k)$ as follows [5]:

$$b_i^s(k) = \begin{cases} \dfrac{B^s C(k)}{n} & if\ p(k) = 0, \\ b_{min} + [(B]^s C(k) - nb_{min}) \dfrac{p_i(k)}{} p(k) & if\ p(k) > 0, \end{cases} \quad \forall Ti \,|\, 1 \leq i \leq n.$$

Where, - p(k) and $p_i(k)$ are the total no. of variations of interest tracked by the system and number of variation being tracked by T_i during the particular time period, respectively.

- b_{min} is the minimum bandwidth allocation to each task so that the lower quality can be transmitted to the receiver.

If the number of objects of interest tracked by the system is 0, bandwidth is equally allocated to each task. If the total number of objects of interest tracked by the system is greater than 0, assume that all objects of interest are of equal importance and bandwidth allocation to tasks based on the number of objects currently being tracked by that task.

Another important factor is the availability. In the following equation it verifies for the availability.

2.2.3 Number of Acceptable Processor Queues

Assigning the tasks to the processor is another important objective. It may overload some of the processors. So, it is necessary to optimize the number of acceptable processor queues. Each processor queue is needed to check individually to see if assigning all the tasks on the processor queues will overload or underload the processors[3].

By using the light and heavy thresholds, it determines that the processor queue is acceptable or not. If the task completion time of a processor is within the light and heavy thresholds then the processor queue is acceptable. If it is below the light threshold and above the heavy threshold, then it is unacceptable.

The percentage of acceptable processor queue is calculated to optimize this objective. It is achieved by

$$Percentage\ of\ acceptable\ processor\ queues = \frac{No.\ of\ processor\ queues}{Total\ number\ of\ processor\ in\ the\ system}$$

The higher the percentage, the better this schedule is in terms of its load-balancing potential.

2.2.4 Average Delay

This is the average of overall delay for a packet to travel from a source node to a destination node. It include the route discovery time, the queuing delay at a node, the retransmission delay at the MAC layer and the propagation and transfer time in the channel.

2.2.5 Link Delay Function

A point to point communication network can be represented by an undirected graph G(V,E) where V is the set of the nodes and E is the set of edges. E represents the communication links connecting to the routers [5]. Link delay is defined as the sum of the perceived queuing delay, transmission delay, and propagation delay.

Therefore,

$$Delay(P) = \sum_{i=1}^{p} LD(e_i)$$

Where, -P is a path, in G which is composed of links e_i , i =1, 2 ...p.

2.2.6 *Packet Loss Rate Estimation*

The PLR is calculated from the number of packet lost at the receiver divided by the number of packets send by the sender during a certain observation period. In this work, the number of lost packets is detected by checking the gap in the PSN filed of the packet header. The number of packets sent can be estimated as the difference between the highest and lowest PSN during the observation periods.

2.2.7 *Average Throughput*

The average throughput is calculated by using the Average number of data packets received by the destination node per second.

2.2.8 *Average Execution Time (AET)*

Execution time is the amount of time an application has to wait to get a reply from the connection admission control module after submitting a connection admission request [6]. Average Execution Time of *N* connections is defined as

$$AET = \frac{Sum\ of\ execution\ time\ of\ N\ connections}{N}$$

2.2.9 *Capacity Calculator for CPU*

Using system information obtained with the resource monitoring tool, a relative capacity metric is computed for each processor using a linear model. Let there are k processors in the system among which the partitioner distributes the work load. For node k, let P_k be the percentage of CPU available, M_k is the available memory and B_k is the link bandwidth [7]. The available resource at k is the first converted to a fraction of total available resources, i.e.

$$P_k = \frac{P_k}{\sum_{i=1}^{k} P_i}$$

$$M_k = \frac{M_k}{\sum_{i=1}^{k} M_i}$$

$$B_k = \frac{B_k}{\sum_{i=1}^{k} B_i}$$

The relative capacity C_k of a processor is then as the weighted sum of these normalized quantities by using the following:

$$C_k = w_p P_k + w_m M_k + w_b B_k$$

Where, w_p, w_m, and w_b are the weights associated with the relative capacity CPU, memory, and link bandwidth availabilities, respectively, Such that $w_p+w_m+w_b=1$. The weights are application dependent and reflect computational, memory and communication requirements.

$$\sum_{i=1}^{k} C_k = 1$$

If the total work to be assigned to all the processors is denoted by L, then the work L_k that can be assigned to the k^{th} processor can be calculated as $L_k = C_k L$.

Load imbalance is defined as

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

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 an account.

2.2.10 *Requirement of CPU*

The required of CPU entitlement (ureq). It is called as utilization controller[7]. It is calculates required CPU entitlement by using the following control law:

Ureq(k)=Ureq(k-1)-K(k)e(k-1)

2.2.11 *TCP-FTP friendly Rate Estimation*

There have been several analytical empirical studies to estimate the throughput of TCP and FTP in steady state.

2.2.12 *Receiver Coordination*

The coordination of receiver under the same bottleneck link is necessary to obtain the fairness. In particular this coordination is significant for layered protocol.

By using all above eqs. in this proposed system increase the QoS. It minimizes the jitter, error rate and checks the availability of the bandwidth so it is able to minimize the throughput increase the threshold value of the task.

It ensures high-quality performance for real time system. Network administrator is using the existing resources efficiently and makes sure the required level of service without reactively expanding or over provisioning their network by using QoS mechanisms. The main concept QoS of DRM is that to increase the throughput, services availability, timeliness and efficient resource utilization is achieved in the proposed system.

2.3 Adaptivity

In the system adaptive threshold policy and scheduling algorithm may used to change the system load. In this technique, it adjusted the threshold as the system load changes. It calculates the average load as follows

$$L_{avg} = \frac{\sum_{i=1}^{N}(CTT_i + QT_i)\sum_{k=1}^{M} TT_k}{N}$$

where,

- L_{avg} is the average load.

- CTT_i is the remaining execution time of the task currently being processed by processor i.

- QT_i is total execution time of all the tasks waiting at processor queue i.

- TT_k is execution time of individual tasks within the sliding-window.

- N is number of processors in the system and M is the number of tasks in the queue.

Since, the next generation information environment has the advanced capabilities. So that, the system must has an adaptive behavior to meet the user expectations and smoothness of the imbalances between demands and changing environment.

2.4 Performance Management

Since, many components are existed in distributed system, as a result, performance management becomes complex. All the components need to be monitored and controlled. Monitoring means track the performance activities of the resources, network and their applications. Controlling function enable the performance management, to make adjustments to improve performance. It supports the systems, to maintains the monitoring and controlling function to keep the track of the resources, network and their applications.

3. Conclusion

In this paper, several measures are defined that are used to measure the performance of the new designed scheduling or routing techniques. From this requirements and measures one can easily develop new technique and easily get idea that what types of parameters are required. In this paper, it is also mentioned that which parameters are minimized and maximize to increase the performance of the system.

4. References

[1] L. Mhamdi, "On the Integration of Unicast and Multicast Cell Scheduling in Buffered Crossbar Switches", IEEE Transaction On Parallel and Distributed Systems, Vol 20, No. 6, pp. 818-830, June 2009.

[2] Dynamic Resource Management Systems, Hreha, William, US Patent Issued on June 4, 2002.

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

[4] Nishanth Shankaran, Xenofon D. Koutsoukos, Douglas C. Schmidt, Yuan Xue and Chenyang Lu, "Hierarchical control of multiple resources in distributed real-time and embedded systems", Springer Science + Business Media, LLC 2007, Real-Time Systems, Vol. 39, pp. 237–282, 2008.

[5] Yuanyuan Yang, Jianchao Wang And Min Yang, "A Service-Centric Multicast Architecture And Routing Protocol", IEEE Transactions On Parallel And Distributed Systems, Vol. 19, No. 1, Pp. 35-51, January 2008.

[6] B. Devalla, A. Sahoo, Y. Guan, C. Li, R. Bettati and W. Zhao, "Adaptive Connection Admission Control for Mission Critical Real-Time Communication Networks", MilCom '98.

[7] 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)