International Journal of Applied Engineering and Technology ISSN: 2277-212X (Online)
An Online International Journal Available at http://www.cibtech.org/jet.htm
2012 Vol. 2 (1) January-March, pp114-119/Saxena et al.
Research Article

AN ALGORITHMIC APPROACH AND COMPARATIVE ANALYSIS OF TASK ASSIGNMENT TO PROCESSOR FOR ACHIEVING TIME EFFICIENCY IN PROCESS COMPLETION

***Pankaj Saxena[1], Kapil Govil[2], Gaurav Saxena[3], Saurabh Kumar[4] and Neha Agrawal[5]**
[1]Teerthanker Mahaveer University, Moradabad (U.P.)
[2]Teerthanker Mahaveer University, Moradabad (U.P.)
[3]Maharaja Agrasen Mahavidyalaya, Bareilly (U.P.)
[4,5]Future Institute of Engineering and Technology, Bareilly (U.P.)
**Author for Correspondence*

ABSTRACT
Task assignment in a distributed manner enhances the distributive nature of the system and thus improves system performance. Matching the task with machines and scheduling task execution on the assigned machines is an aspect of task allocation. The task allocation in a distributed computing system finds extensive applications in the area where large amount of data is to be processed in relatively short period of time. This paper emphasis the problem of task allocation in a distributed computing environment by developing an algorithm which calculates the optimal time results in a distributed manner and thus improves the system reliability. Task allocation is an essential phase in the distributed computing systems. Task allocation to balance the load on multi computers is a challenge due to the autonomy of the processor and the interprocessor communication overhead incurred in the collection of state information, communication delays, redistribution of load...Etc. Task allocation algorithms can be of two basic categories-static and dynamic. static task allocation take decisions for assignment of task to processor based on the average estimated values of process execution times and communication delays at compile time where dynamic task allocation algorithms are adaptive to the changing scenario and take the decision at run time. The problem of task assignment in heterogeneous computing system has been studied for many years with many variations. In distributed computing the schedule by which task are assigned to processor is critical to minimize the execution time of the application. However the problem of discovering the schedule that gives the minimum execution time is NP-complete. The main advantage for considering DCS are higher throughput, improved availability and better access to a widely communicated area of information.

Keywords: *Distributed Computing System (DCS), Task Allocation, Throughput.*

INTRODUCTION
In DCS processors are connected together through a communication network. The term distributed computing system is used to describe whenever there are several computers interconnected in some fashion so that a program or procedure running on system with multiple processors. However the term has different meanings with regard to different systems because processors can be interconnected in so many ways for various reasons. In the most general form, the word distributed implies that the processors are in geographically separate locations. DCS gives higher throughput and improved availability. A Distributed Computing System (DCS) is a network of workstations, personal computers and/or other computing systems. A DCS accepts tasks from users and executes different modules of these tasks on various nodes of the system. The objective of Distributed Computing is to obtain higher execution speed in comparison to the one obtainable with uniprocessor system by exploiting the collaboration of multiple computing nodes interconnected. Distributed computing system (DCS) concurrently process an application program by employing multiple processors. It not only provide the facility for utilizing remote computer resources or data not existing in local computer systems but also minimize the system cost by providing the facility for parallel processing. To fulfill the requirement of faster computation, one approach is to use distributed

computing system. An interesting problem in DCS is the task allocation. The problem deals with finding an optimal allocation of tasks to the processor so that the execution cost and communication cost can be minimized. Task allocation is the process of partitioning a set of programming modules into a number of processing groups. Task allocation is a common problem in many different applications. However, there is no algorithm that can solve this problem exactly, so finding optimal solution in a reasonable time, for any large configuration of jobs is needed. An efficient approach is to partition a uniprocessor computing load into multiple units of execution and assigning them to the various processing nodes. The best possible speed up will obviously be obtained if the various partitions of the given computational task can run independently but in parallel. Various partitions of the task collaborate to achieve a common objective. The processing nodes of the system must be sharing the computational load and the processing nodes must be made busy as much as possible by receiving and executing multiple tasks. There are many ways by which the Allocation of tasks in a DCS may be done. Static task scheduling takes place during compile time before running the parallel application. In static scheduling algorithms, all information needed for scheduling must be known in advance. There are several techniques to estimate such information. Static task assigning is related to compile time. Round Robin, Random, Central Manager and Threshold are some load balancing algorithms which are static in nature. In DCS the allocation policy depends upon the time at which the allocation decisions are made. Static allocation techniques can be applied to the large set of real word applications. This research paper is an effort to identify a better way of distributing the computational load across the processing nodes to achieve higher system throughput in terms of number of tasks executed per unit time. Dynamic scheduling of tasks is related to run time.

Objective

The objective of present research paper is to give a new technique to assign the different task on processor and to balance the load for getting the work done in minimum possible time duration in a distributed network of many processors where each processor accomplishes the task to get the optimal results more quickly as well as more efficiently task allocation problem is an important problem where the number of tasks is more then the number of processors. The type of assigning task to the processor is static in nature.

MATERIALS AND METHODS

For adopting any technique firstly it is must to understand the problem clearly. The problem is how to allocate the tasks to the processor for getting the time efficiency. In this scenario we have fixed number of tasks and fixed number of processors. Number of tasks are more then the number of processors. Our objective is to achieve the results in minimum possible time. We use the technique which is based on permutation. We calculate the total number of ways by which we can complete all the tasks. Now we determine the maximum number of tasks which can be allocated to a single processor by making sure that no processor will be empty. Now the tasks with maximum number of files have been allocated to the processor which will take the minimum completion time. In a similar manner we will allocate all the tasks to the processor and will calculate the total time to finish all the tasks.

ALGORITHM

Start Algo

Step1: Read the number of tasks in n.
Step2: Read the number of processor in r.
Step3: Choose the tasks which have maximum number of files.
Step4: Find out the processor which takes minimum time to complete the task.
Step5: Allocate the tasks in step3 to the processor selected in step4.
Step6: Calculate the time taken by the selected processor to complete the task in step3.
Step7: Repeat the procedure from step1 to step 6 while (all the tasks! =allocated)
Step8: Calculate the total time to finish all the tasks.
End Algo

IMPLEMENTATION

Let we have four processors and six tasks.

Processor P_1 takes 5 seconds.
Processor P_2 takes 4 seconds.
Processor P_3 takes 6 seconds.
Processor P_4 takes 8 seconds.

Diagrammatically it can be shown by the following way-

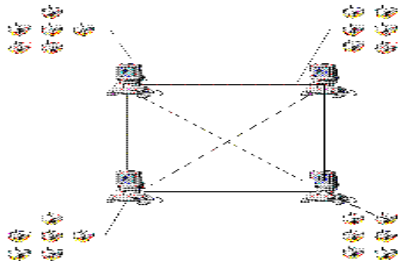

Figure 1: Task Allocation

It is considered that-

Task T_1 has 4 files.
Task T_2 has 3 files.
Task T_3 has 2 files.
Task T_4 has 1 file.
Task T_5 has 5 files.
Task T_6 has 6 files.

Every processor must be engage for balancing the load so if we allocate 3 tasks to any single processor then rest of the 3 tasks can be allocated to other 3 processor as we are using total 4 processor. There are total 6 tasks and utilization of all the processors is must so,

Maximum number of tasks can be done by a processor=3

All the tasks can be done by all the processors = ${}^{6}P_4$ Number of ways
=6! / (6-4)!
=360 Number of ways

Since we want that that processing of all the task must be done in minimum time period so we will choose the tasks which has maximum number of files and will allocate to that processor which takes minimum time for task completion.
So we will take the tasks T_1, T_5, and T_6 at processor P_2.

So, total time taken by processor P_2 to complete the tasks T_1, T_5 and T_6= (4+5+6)*4 Seconds
=15*4 Seconds
=60 Seconds
=1 Minute

Similarly,

Total time taken by processor P_4 to complete the task T_4= (1*8) Seconds
=8 Seconds
Total time taken by processor P_3 to complete the task T_3= (2*6) Seconds
=12 Seconds
Total time taken by processor P_1 to complete the task T_2= (3*5) Seconds
= 15 Seconds
Total time= (8+12+15) Seconds
=35 Seconds

So, Total time taken to finish all the tasks by involving all processors =1 Min+35 Seconds
= 1 Min 35 Seconds or (95 Seconds)

COMPARATIVE STUDY

Following are given the two tables namely Table 1 and Table 2. Task can be assigned to the processor in two ways, first one is static and second one is dynamically or at run time. Table 1 show the comparison based on some issues like preemption, thrashing, predictability and many others. By this table some pros and cons of dynamic as well as static allocation of task to processor can also be calculated

Table2 shows a comparative and analytical study of different load balancing algorithms such as threshold, round robin, central queue and random. This table focuses on several issues like fault tolerant, process migration.Etc.by this table different issues to balance the load on processor can be observed by different algorithmic views.

Static Vs Dynamic Task Assignment: Comparative Analysis

Table 1: ***Task Assignment***

S. No.	Static Task Assignment	Dynamic Task Assignment
1.	Compile time task assignment	Run time task assignment
2.	No preemption	Preemptive as well as non preemptive
3.	Better predictability	Not much predictable
4.	Shows less reliability	Greater reliability
5.	It does not involves processor thrashing	Substantial processor thrashing

Table 2: Load Balancing

Sr. No.	Threshold	Round Robin	Central Queue	Random
1.	It does not include overload rejection	Not include overload rejection	It includes overload rejection	No overload rejection
2.	Includes no fault tolerant	No fault tolerant	Includes fault tolerant	No fault tolerant
3.	Includes no process migration	Not include process migration	Not include process migration	No process migration
4.	Static in nature	static	Dynamic in nature	Static
5.	Decentralized in nature	decentralized	centralized	Decentralized

Conclusion

As conclusion gives the result of the objective in any kind of research. In the present research paper this part shows us that how we can use the permutative approach in making of any task allocation algorithm and can increase the parallelism as well as the throughput of the entire system in a minimal time. The results are also compared with other time based allocation algorithms to check the comparative results. This paper gives an easy and efficient approach of scheduling tasks to processor.

REFERENCES

Benoita A, Casanovab H, Rehn-sonigoc V and Roberta V (2011). Resource allocation for multiple concurrent in-network stream-processing applications. *Journal of parallel computing* **37(**8) 331-348.

Mohammad I.daoud, Nawwaf kharma (2008). A high performance algorithm for static task scheduling in heterogeneous distributed computing systems. *Journal of parallel and distributed computing* **68(4)** 399-409.

Parag C pendharkar (2011). A multi agent memetic algorithm approach for distributed object allocation. *Journal of computational science* **21 (4)** 353-364.

Murat sensoya, Wamberto, W.vasconcelosa, Timothy, J.normana, Katia sycaraa (2012).Reasoning support for flexible task resourcing. *Journal of expert system with applications* **39 (2)**1998-2010.

Nirmeen A.bahnasawyb, Fatma omaraa, Magdy A.koutbb, Mervat mosab (2011) .Optimization procedure for algorithm of task scheduling in high performance heterogeneous distributed computing system. *Egyptian informatics journal***12 (3)** 219-229.

Reakook hwanga, Mitsuo genb, Hiroshi katakana (2008).A comparison of multiprocessor task scheduling algorithms with communication cost. *Journal of computers and operation research* **35 (03)** 976-993.

Kwangsik shin, Myongjin cha, Munsuck jang, Jinha jung, Wanoh yoon, Sang choi (2008). Task scheduling algorithm using minimized duplications in homogeneous systems. *Journal of parallel and distributed computing* **68(08)**1146-1156.

Fatma A.omaraa, Mona M.arafab (2010). Genetic algorithms for task scheduling problem. *Journal of parallel and distributed computing* **70(01)** 13-22.

P.chitra, R.rajaram, P.venkatesh (2011). Application and comparison of hybrid evolutionary multiobjective optimization algorithms for solving task scheduling problem on heterogeneous systems. *Research article on applied soft computing***11 (02)** 2725-2734.

Zhiao shi, Jack j.dongarra (2006). Scheduling workflow applications on processors with different capabilities. *Research article on future generation computer systems* **22(06)** 665-675.