Problems in Task Scheduling in Multiprocessor System

Mukul Varshney
CSE , Sharda
University, India

Jyotsna
CSE , Sharda
University, India

Abha Kiran Rajpoot
CSE , Sharda
University, India

Shivani Garg
CSE , Sharda
University , India

ABSTRACT

This contemporary computer systems are multiprocessor or multicomputer machines. Their efficiency depends on good methods of administering the executed works. Fast processing of a parallel application is possible only when its parts are appropriately ordered in time and space. This calls for efficient scheduling policies in parallel computer systems. In this work deterministic problems of scheduling are considered. The classical scheduling theory assumed that the application in any moment of time is executed by only one processor. This assumption has been weakened recently, especially in the context of parallel and distributed computer systems. This monograph is devoted to problems of deterministic scheduling applications (or tasks according to the scheduling terminology) requiring more than one processor simultaneously. We name such applications multiprocessor tasks. In this work the complexity of open multiprocessor task scheduling problems has been established. Algorithms for scheduling multiprocessor tasks on parallel and dedicated processors are proposed. For a special case of applications with regular structure which allow for dividing it into parts of arbitrary size processed independently in parallel, a method of finding optimal scattering of work in a distributed computer system is proposed. The applications with such regular characteristics are called divisible tasks. The concept of a divisible task enables creation of tractable computation models in a wide class of computer architectures such as chains, stars, meshes, hypercubes, multistage networks. Divisible task method gives rise to the evaluation of computer system performance. Examples of such performance evaluation are presented. This work summarizes earlier works of the author as well as contains new original results.

KEYWORDS: Load Scheduling, SISD, MIMD, MISD, SIMD, Communication Overhead

1. INTRODUCTION

NELARABINE is a purine nucleoside analog converted to its corresponding arbinosylguanine nucleoside tri phosphate results inhibition of DNA synthesis and cytotoxicity. The drug is ultimately metabolized into the active 51-tri phosphate ara –GTP which disrupts the DNA synthesis and induces apoptosis. Nelarabine is used to treat T-cell acute Lymphoblastic leukemia and T-cell lymphoblastic lymphoma. The Chemical name of Nelaribine (2R,3S,4S,5R)-2-(2-amino-6-methoxy-9H-purin-9-yl)-5 -(hydroxymethyl) oxolane-3,4-diol and chemical formula is $C_{10}H_{11}ClFN_5O_3$ and molecular weight is 297.27 g/mol.

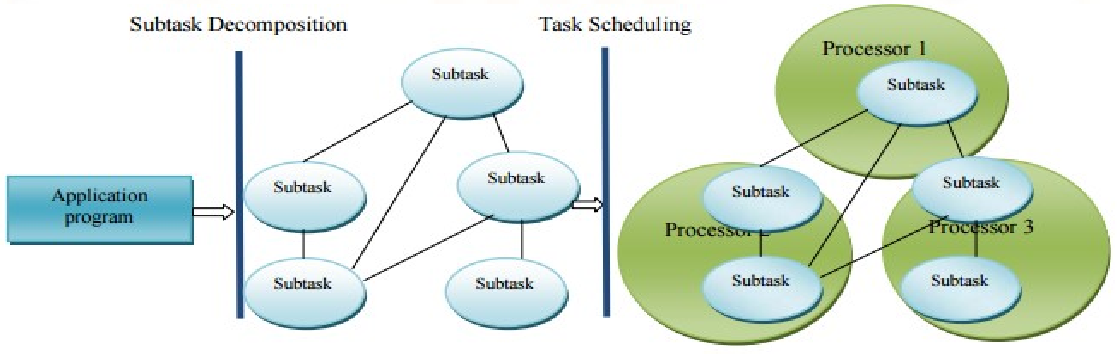

Fig.1. Transforming from an application program to task scheduling

II. PARALLEL COMPUTER SYSTEMS

A. Hardware

It is common to start a description of parallel systems with an attempt of classifying types of parallelism and types of parallel machines. A useful view on parallelism types is distinguishing between data parallelism and code parallelism. . Another view of parallel processing classification considers granularity of parallelism. Classical computers execute instructions in the order dictated by the sequence in the program code. This approach is called control-driven or von Neumann architecture.

In control-driven computers have been divided into four classes: SISD (single instruction stream, single data stream), SIMD (single instruction stream, multiple data streams), MISD (multiple instruction streams, single data stream), MIMD (multiple instruction streams, multiple data streams). A multicomputer consists of a set of processors with local memories, interconnected by some kind of network. We will name by processing element (PE) a processor with local memory and a network interface. Tightly-coupled computers can be further dierentiated by the type of PE interconnection. In this work we limit considered interconnection types to: bus(es), point-to-point networks (called also single-stage networks).

In multistage networks PEs are connected by several layers of switches while the internal layer switches have no PEs attached. Multistage networks are divided here into: trees and multistage cube network

B. Software

In many common applications (programs) great potential parallelism can be found. Thus, programs can be executed via many concurrent threads (mutual relations between the notions of an application, a thread and a task). Computer systems should provide support for implementing parallelism of an application including the issues posed by scheduling. Parallel operating systems are evolving from previously existing systems and many ideas have been "naturally" inherited. Based on acceptable response time two load types have been distinguished in single-processor systems: terminal (or interactive) and batch load. Since batch tasks are submitted to the computer system far earlier than their actual execution begins, deterministic scheduling algorithms can be applied. For the terminal load which requires immediate response, access to processors is granted on the basis of FCFS, Round-Robin, multi-level priority queues etc

III. COMMUNICATION MODE

The next element of the architecture is the commutation mode. The commutation mode is a physical protocol for message routing. We describe commutation modes here because routing functions are increasingly executed by dedicated hardware. The methods we refer to in this section are also called switching or routing techniques. Among various commutation (or routing) modes we distinguish store-and-forward, circuit-switched and packet-switched.

In the store-and-forward mode when a PE sends a message to another PE located at distance d, the message (either as whole or in packet pieces) is sent to the closest PE on the path and it is stored there.

In the circuit-switched mode from the transmitter to the receiver a header of the message is sent which reserves all the links of a communication path to form a circuit between both PEs.

In the packet-switched modes the message is split into packets which consist of its. Flits are also called ow of control digits. These are words passed over a link in one control cycle.

Among the packet-switched modes three sub-types can be identified: wormhole, virtual-cut-through and buered-Wormholemodes. These modes differ in the behavior of its and packets when the packet cannot move forward (e.g. there is no free link). In the wormhole mode, the progressing of the message in the pipe is stopped. All the its remain in the intermediate buffers thus blocking the links.

In the virtual-cut-through mode its continue progressing on their way until reaching the site where the rst it is stopped. There a whole packet is waiting for release of the link. This mode assumes infinite capacity of buffers

In the buffered wormhole mode, its of some packets move until they reach the stopped it, then the whole packet is stored there. Yet, the number of packets that can be stored is limited by buyer’s capacity.

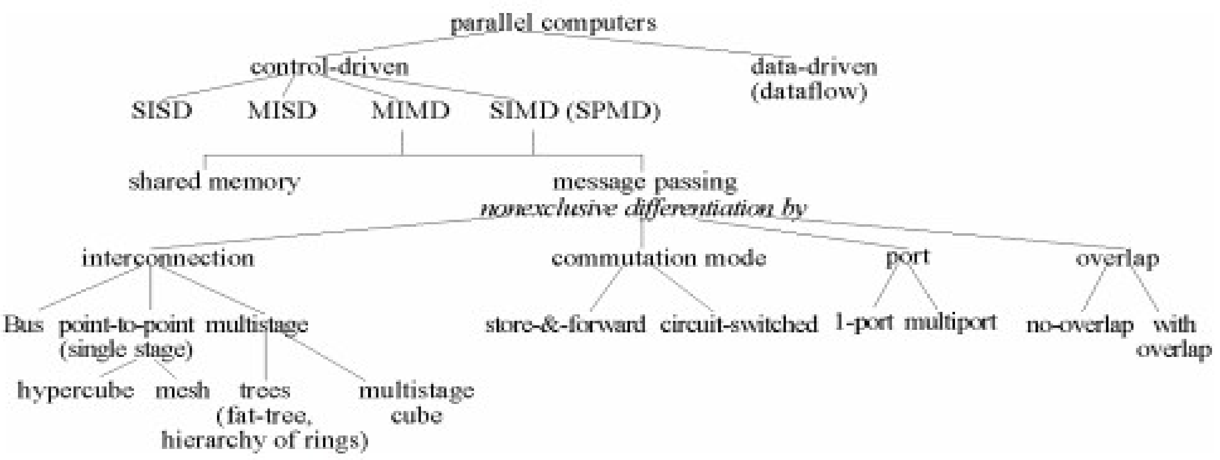

Figure 3 : The classification of parallel computers.

IV. FORMULATION OF TASK SCHEDULING PROBLEM

A task scheduling problem consists of the application model, system computing model and performance evaluation metrics. This section will discuss an application model, a system computing model followed by performance evaluation metrics.

A. Application model

Task scheduling of a given application is represented by directed acyclic graph (DAG) G1= (T, E), where T is the finite set of m tasks {T1, T2, T3…Tm} and E is the set of edges {eij }between the tasks Ti and Tj. Here, each edge represents the precedence constraints between the tasks Ti and Tj such that Tj can not start until Ti completes its execution. Each task Ti is associated with an execution time ET (Ti) and each edge eij is associated with a communication time CT (Ti, Tj) for data transfer from Ti to Tj. If there is a direct path from Ti to Tj then Ti is the predecessor of Tj and Tj is the successor of Ti. A entry task does not any predecessor, similarly, an exit task does not any successors. Layout of a DAG with six tasks is shown in figure 4 [16]

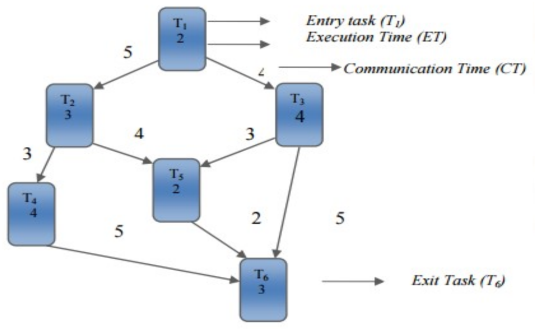

Fig. 2 DAG Model with six tasks

Where T1, T2, T3, T4, T5 and T6 are different tasks of the given DAG. The execution time ET (Ti) and communication time CT (Ti, Tj) [16] of tasks are: ET (T1) = 2 ET (T2) = 3 ET (T3) = 4 ET (T4) = 4 ET (T5) = 2 ET (T6) = 3 CT (T1, T2) = 5 CT (T1, T3) = 4 CT (T2, T4) = 3 CT (T2, T5) = 4 CT (T3, T5) = 3 CT (T3, T6) = 5 CT (T4, T6) = 5 CT (T5, T6) = 2

B. Algorithm

Rules Design objectives of the algorithm:

• Able to dynamically regulate task allocation to each processor according to changes in the system load, optimize the utilization of processor resources

• Minimize the effect on the working performance of the processor. The node processor in the algorithm is defined into the four states as follows:

R I: Suppose A1, A2, B1 and B2 represent the processor node set in heavy load, busy, light load and no load states respectively, load balancing is carried out in following steps:

• Equilibrium operation is conducted between the heavy loaded node and light loaded node

• Equilibrium operation is conducted between the busy node and the no load node when there is no heavy load Node. • Repeat steps 1 and 2 until no migratable task can be selected or the loads on the processor Nodes are relatively balanced.

R II: The execution time of the parallel program is always restricted by the task at the slowest execution speed L denotes the load. So when load L moves from A1 to B1.

C. Load Evaluation

It's common to evaluation node load by the array length of CPU. And this load evaluation mode is concise and rapid .In order to evaluation load state of each node; we let character C to represent process capability of each node in homogeneous multiprocessor system. Namely, it's maximum process number that can be processed by CPU in per unit time.

V. CONCLUSIONS

In this work we considered selected methods of scheduling in multiprocessor computer systems. With the advent of modern computer systems it turned out that classical scheduling methods in many cases are not satisfactory Therefore, two new scheduling models were analyzed here: multiprocessor tasks and divisible tasks. Multiprocessor tasks require several processors simultaneously, thus allow for expressing task parallelism at high level of abstraction. This model allows for finding simple solutions of problems which in other setting are again intractable. Moreover, divisible task concept permits introducing computer architecture context which in the classical approach is often highly generalized to make problems manageable. In this way scheduling problems have been combined with communication optimization problems. Experimental simulation has shown that the algorithm can solve very well the load balancing problems in a multiprocessor system and significantly improve the system performance and task processing. The proposed algorithm is characterized by less load balancing frequency and less overhead. It can be seen from the results of the experiment that use of the algorithm proposed in the paper can get good performance

REFERENCES

[1] Wentao Wang, Xiaozhong Geng, Qing Wang 'Design of a Dynamic Load Balancing Model for Multiprocessor Systems' IEEE 2011, 641-643.

[2] Xin Liang Tang , Peng Liu, Zhen Zhou Wang, Bin Liu 'A load balancing algorithm for homogeneous multiprocessors system IEEE 2010. 1186-1190. .

[3] Ralf Hoffmann, Andreas Prell, and Thomas Rauber 'Dynamic Task Sheduling and Load Balancing on cell processors' IEEE 2010, 205-212

[4] Dynamic Load Balancin Shiyuan Jin, Guy Schiavona and DamlaTurgut, "A performance study of multiprocessor task scheduling algorithms", Journal of Supercomputing ,Vol.43,pp.77-97,2008.

[5] M.R.Garey and D.S.Johson,Computers and Intractability: A Guide to the Theory of NP - completeness,1979.

[6] G.N Srinivas and Bruce R. Musicus. "Generalized Multiprocessor Scheduling for Directed Acyclic Graphs" In Third Annual ACM Symposium on Parallel Algorithms and Architectures, pp.237-246, 1994.

[7] G.L.Park, BehroozShirazi, Jeff Marquis and H.Choo. "Decisive Path Scheduling: A new List Scheduling Method" Communication of the ACM, vol.17, no.12, pp. 472-480,Dec1974.

[8] Min You Wu "On Parallelization of static scheduling Algorithms", IEEE Transactions on Parallel and Distributed Systems, Vol.10, No.4, , pp 517-528,April 1999

[9] C.H.Papadimitrious and M.Yannakakis, "Scheduling Interval-Ordered Tasks, " SIAM Journal of Computing,Vol.8,pp.405-409,1979.

[10] R.Sethi, "Scheduling Graphs on Two Processors," SIAM Journal of Computing,Vol.5, No.1,pp.73-82, Mar.1976.

[11] Oliver Sinnen,"Task Scheduling for Parallel Systems" Wiley-Interscience Pulication, 2007.

[12] Edwin S.H and Nirwan Ansari, "A Genetic Algorithm for Multiprocessor Scheduling", IEEE Transaction on Parallel and Distributed Systems, Vol.5, No.2, Feb.1994.

[13] Hadis Heidari and Abdolah Chaechale , " Scheduling in Multiprocessor System Using Genetic Algorithm," International Journal of Advanced Science and Technology,Vol.43, pp.81-93,2012.

[14] RavneetKaur and RamneekKaur,"Multiprocessor Scheduling using Task Duplication Based Scheduling Algorithms: A Review Paper", International Journal

of Application or Innovation in Engineering and Management,Vol.2 Issue 4,pp.311-317,April,2013.

[15] XiaoyongTang,Kenli Li and Guiping Liao ," List Scheduling with duplication for heterogeneous computing systems", Journal of Parallel and Distribute Computing,Vol.70, pp.323-329,2010.

[16] MostafaR.Mohamed and MedhatH.A,"Hybrid Algorithms for Multiprocessor Task Scheduling", International Journal of Computer Science Issues, Vol.8, Issue.3 No.2 May,2011. g Scheduling Model Based on Multi-core Processor, IEEE DOI,2010,398-403. FLEXChip Signal Processor (MC68175/D), Motorola, 1996.

[17] A new method for scheduling load balancing in multi-processor systems based on PSO, IEEE Computer,2011, 71- 76 .

[18] T. Armitage and J.G. Peters. Circuit-switched broadcasting in 3-dimensional toroidal meshes. manuscript, 1995