Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 6, June 2017, pg.175 – 179

Topological Properties of Multiprocessor Inter Connection Network

Mukul Varshney*
CSE, Sharda University
Mukul.varshney@sharda.ac.in

Anand Sharma
ACET, Aligarh
anandsharmaacet@gmail.com

Jyotsna
CSE, Sharda University
jyotsna.seth@sharda.a.cin

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. - The main focus of researchers nowadays is to massively increase the computational power of the system. Exploiting parallelism a necessity in designing high performance computer systems. In terms of hardware it refers to providing multiple simultaneously active processors (nodes). In terms of software it means structuring program as a set of largely independent subtasks (load). Research is active in the direction of developing new multiprocessor architecture and scheduling the partitioned program on to it in order to achieve high performance. In this proposed study.

Keywords— Load Scheduling, SISD, MIMD, MISD, SIMD, Communication overhead, speed up

I. INTRODUCTION

The increase of the computer speed and their ability to solve bigger and bigger problems is an everlasting challenge for the designers. As computer systems grow more complex and their speed increases the problems that must be overcome to further increase the speed and the "capacity" seem to grow even faster. The difficulties follow physical phenomena at the foundations of computer devices technology. Parallel computing is an important research area in Computer Science. However, it consist number of problems that are not solved in sequential machine such as designing of parallel algorithms for an application program, dividing of an application program into subtasks, synchronization and coordinating communication, and scheduling of the tasks onto the processors. Thus, to increase the speed the yield of the current sources must be higher or the size of the devices must be smaller. Furthermore, in order to minimize the number of defected circuits in one piece of silicon, the chips are reduced in size. This, and growing complexity of the processors results in increasing density of power dissipation. Yet, it cannot grow to infinity. Moreover, since the photolithography methods are limited by the light wave length further miniaturization becomes slower and more costly than in the recent years. Hence, it seems that unless new ways [1,9,7] are found to overcome the existing technological problems the development of processors will be slower and prohibitively expensive [1,2,3]. A solution to this problem can be in exploiting potential simultaneity in execution of some independent program fragments. In other worlds, exploiting parallelism of computations can be the answer.

Parallel computing is the next generation of computers and has made impact on various areas from scientific and engineering applications to commercial applications. Scheduling [1] a set of dependent or independent tasks for parallel execution on a set of multiple processors is an important computational problem where a large problem or an application is divided into a set of subtasks and these subtasks are distributed among the processors. The allocation of subtasks on processors and defining their order of execution is called as Task scheduling. Task scheduling is an NP-complete problem of its simple case [2] and some restricted cases [3,4,5,6,7]. It has been used in number of application from scientific to engineering problems. Figure.1 [8] shows the layout of transformation from an application program to task scheduling. Here, an application program is divided into a number of subtasks that are represented by Directed Acyclic Graph. They are allocated to the multiple processors and should be maintained precedence constraints [9] among the subtasks.

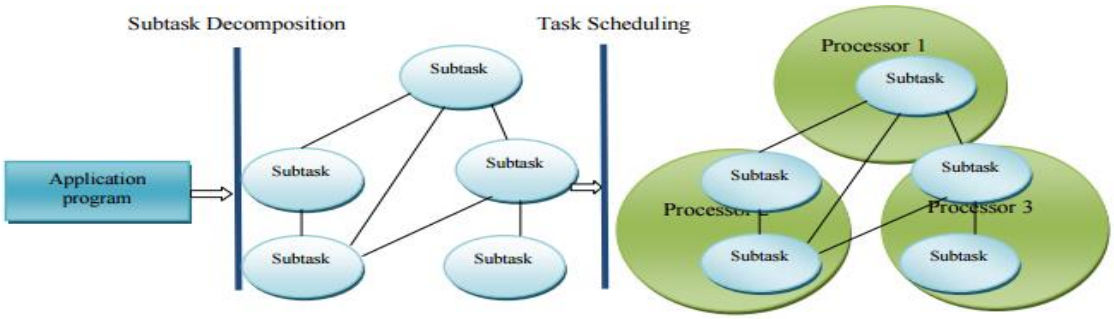


Fig.1. Transforming from an application program to task scheduling

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 differentiated 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.MULTIPROCESSOR INTERCONNECTION NETWORK

NETWORK CHARACTERISTICS

Operation Modes: There are two basic modes of network operation; synchronous and asynchronous. In the synchronous mode, the network is centrally supervised, connection paths established simultaneously. In the asynchronous mode, connection paths are setup or disconnected on an individual basis. The asynchronous mode is of operation is more appropriate for multiprocessor system**.**

Switching techniques: There are three basic switching techniques: circuit switching, packet switching and wormhole switching.

Circuit Switching: Sets up the switches and ports and establishes a dedicated path between an input-output pair, efficient for large transmissions

Packet Switching refers to a technique in which messages between any two terminals are broken into several shorter, fixed-length packets which are routed independently to their destination using store-and-forward procedures. Compared with circuit switching, packet switching is efficient for shorter and more frequent transmissions.

Routing Techniques: it is the method of establishing communication paths and resolving conflicts. Three basic routing techniques have been considered: centralized, distributive and logical decisions are made locally based on the current conditions. In the adaptive scheme, information about the network is collected globally, but routing decisions are made locally.

Interconnection Network Topology characteristics:

Number of Nodes: the number of nodes affects the complexity of the system and hence the cost of the system.

Degree of Node: it is the number of connections required at each node. it determines the complexity of the network, so it should be as low as possible.

Diameter: it is the measure of the maximum internodes distance in the network. It determines the distance involved in communication and hence the performance of multiprocessor system.

Extensibility: it is a property, which facilitates constructing large sized systems out of small ones with minimum changes in the configurations of the nodes.

Fault Tolerance: in multiprocessor network if one or more components fail then it should work adaptive. In the centralized routing, a central control makes all the logic decisions needed to set up communication paths. This scheme is more flexible for small to medium scale systems in the distributed scheme.

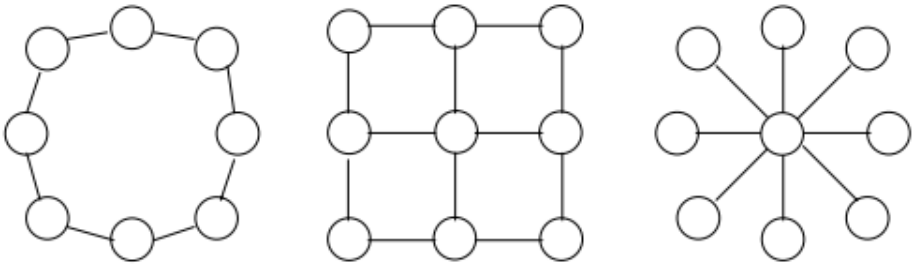
IV. INTERCONNECTION NETWORK TOPOLOGY

Topologies in the static networks can be classified according to the dimensions required for layout. For illustration, 1-dimentional, 2- dimensional, 3-dimentional are shown below.

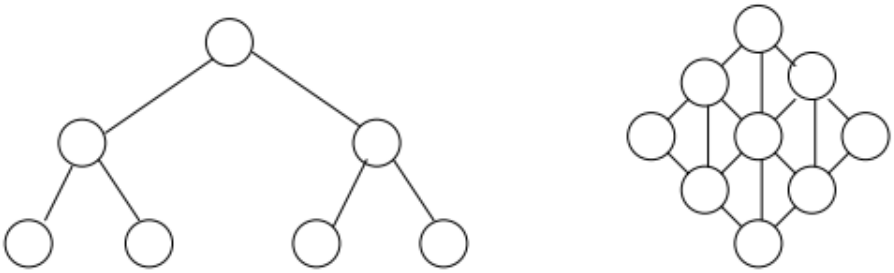
1-dimentional interconnection network

(a) Linear

2-dimentional interconnection networks



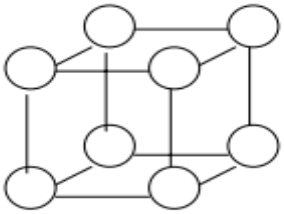
(b) Ring (c) Mesh (d) Star



(e) Tree (f) Systolic

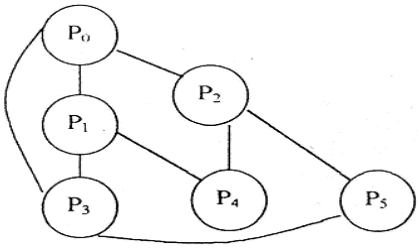
1-dimentional interconnection network includes the linear array used for pipeline architectures. 2-dimentional topologies include the ring, star, tree, mesh, and systolic array. 3-dimentional topologies include 3–cube, cube connected cycles, etc.

3-dimentional interconnection

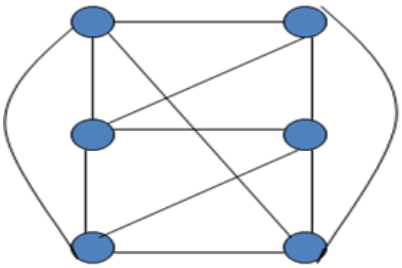


(g) 3-cube

Linearly Extensible Multiprocessor Architecture



Linerly Extensible Tree



Linerly Extensible Cube

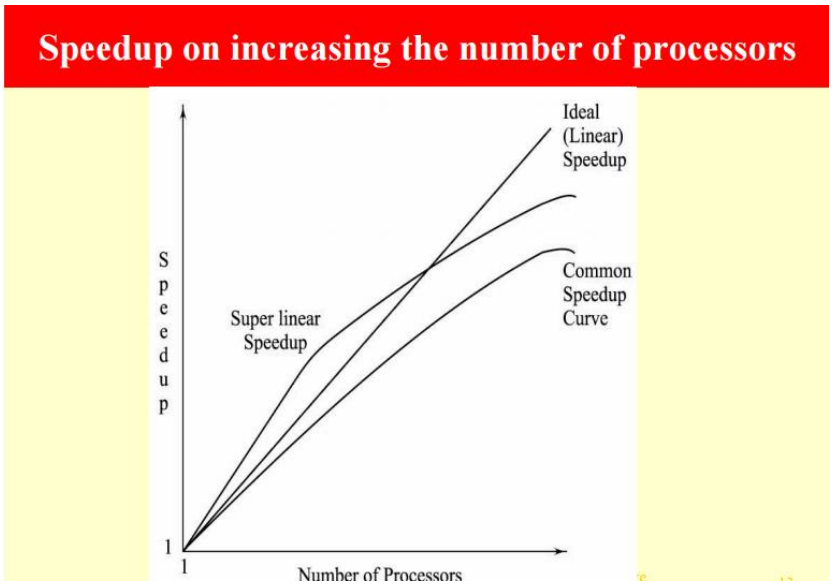
V. TOPOLOGICAL PROPERTIES

Network	No. of Link	Node degree.	Diameter	Bisection Width
Star	N-1	N-1	2	N-1
Completely Connected	N(N-1)/2	N-1	1	$(N/2)^2$ √N
Binary Tree	N-1	3	$2(Log_2N -1)$	1
Illiac Mesh	2N	4	√N -1	2 √N
Hypercube	$NLog_2N/2$	N	Log_2N	N/2
De-bruijn	2N-1	4	Log_2N	N/Log_2N
LET	N+2	4	√N	2
LEC	$(N/2)^2 +3$	4	N	N

VI. MULTIPROCESSOR SPEED UP

Designers of uniprocessor systems, measure performance in terms of speedup • Similarly, multiprocessor architects measure performance in terms of speedup.

Speedup generally refers to how much faster a program runs on a system with n processors than it does on a system with one processor of the same type



Linear speedup – the execution time of program on an n-processor system would be l/ nth of the execution time on a one-processor system

When the number of processors is small, the system achieves near-linear speedup • As the number of processors increases, the speedup curve diverges from the ideal, eventually flattening out or even decreasing

There is some limitation on multiprocessor speed up in terms of Interprocessor communication • Synchronization • Load Balancing

Whenever one processor generates (computes) a value that is needed by the fraction of the program running on another processor, that value must be communicated to the processors that need it, which takes time • On a uniprocessor system, the entire program runs on one processor, so there is no time lost to interprocessor communication

It is often necessary to synchronize the processors to ensure that they have all completed some phase of the program before any processor begins working on the next phase of the program.

VII. CONCLUSIONS

In this work we considered selected topology of multiprocessor architecture and their characteristics. 3-dimensional architecture gives better performance in compare to other architecture. 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. Their is some limitation in terms of communication over head and processing overhead.

REFERENCES

[1] Manaullah, "Performance Evaluation of Multiprocessor Architectures", Ph.D. thesis, Jamia Millia Islamia, 2002.
[2] Janhavi B., Sunil Surve, Sapna Prabhu, "Comparison of Load Balancing Algorithms in a Grid", 2010 International Conference on Data Storage and Data Engineering, pp 20-23, 2010.
[3] Abdus Samad, M. Q. Rafiq and Omar Farooq, "A Novel Algorithm for Fast Retrieval of Information from a Multi processor Server", 7 th WSEAS Int'l Conf. on SOFTWARE ENGINEERING, PARALLEL AND DISTRIBUTED SYSTEMS (SEPADS '08), University of Cambridge, UK, pp 68-73, 2008.
[4] Chandra, Pushpendra Kumar, Sahoo, Bibhudatta, "Dynamic Load Distribution Algorithm Performance in Heterogeneous Distributed System for I/O Intensive Task", TENCON 2008 - 2008, TENCON 2008, IEEE Region 10 Conference, pp 1-5, 2008.
[5] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma, "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology 38, pp 269-271, 2008.
[6] Abdallah Boukerram, Samira Ait Kaci Azzou, "Implementation of Load Balancing Algorithm in a Grid Computing", American Journal of Applied Sciences, 2006.
[7]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.
[8] M. Q. Rafiq, "Studies on the Performance Evaluation of a Linearly Extensible Multiprocessor Network", Ph.D. thesis, Univ. of Roorkee, 1995.
[9] 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.
[10] 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.

