Lecture 2

Some PE’s have shared memory, whilst others could have only local memory- if one is slow, it will not have to depend on the other and still run its processes independently.

RAM

Memory could be considered infinite and divided into basic units that can be accessed by location/ address. PU can access any memory location- random access. Can use the graphical representation to represent a random access machine PE connected to a memory.

We try to abstract the more important properties and ignore less important ones in theoretical modelling of serial computing.

Important:

Memory

PU

PRAM- Multiple PE’s connected to a global memory. Similar to RAMachine. All PE’s have uniform/ equal access to the global memory and we assume that this memory has infinite size. The issue is if multiple PE’s need to access the same memory location. Solution is to serialise such contending accesses- at one time only one PE is allowed to access a particular memory unit.

Simultaneous access to same memory location can lead to unpredictable data in PE’s, also in memory locations that are being accessed

Variants of PRAM:

EREW-PRAM. Serialized memory access. Minimum concurrency in memory accesses

CREW-PRAM.

CRCW-PRAM

LMM- Local Memory Machine. Each unit is a RAM (P1 and M1 etc)

Performance of real world parallel applications are more dependent on the communication time rather than the calculation time. High performance interconnection networks could lead to more efficiency in parallel applications. Not only in the form of shorter parallel executions time but also more processing elements that can be exploited in parallel applications.

Chart

Description automatically generated

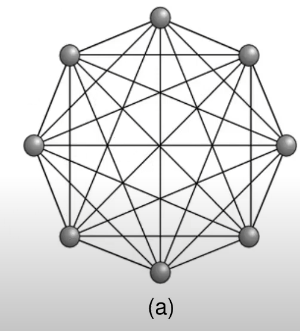
1. Fully connected network- each node (considered as a RAM, PE with a memory module) is connected to every other node. LMM model where each PE has its own local memory and each processing node is connected via its interconnection network.
2. Fully connected crossbar network- modular memory machine. Each PE is connected to every other memory module. Each intersection in grid represents a switch. Bold dots represent a state in the connection e.g. P1 is connected to M1 etc.

Both are non-blocking, in which connection between pair of input and output nodes does not block connections between any other pair of input and output nodes

Text, letter

Description automatically generated

Node degree- how many edges connect to that node, therefore **how many links connect to the node**



1. Node degree is 7

Regular as each node has same degree of 7

Hop count- number of links in a path. Minimal hop count is shortest num links after considering all paths.

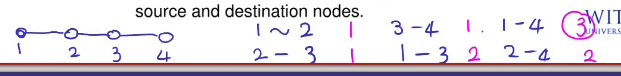
A picture containing accessory, giant, dome

Description automatically generated

Minimum hop count- 1

**Diameter is maximum hop count amongst all pairs of nodes**

E.g.



The purple is the hop count. 3 is the largest amongst all the pairs, therefore is the diameter.

**Quantitative measures for performance properties of interconnection network**

Text

Description automatically generated

Bisection width- min number of links taken out to split network into 2

Chart, line chart

Description automatically generated

This bisection width is 1

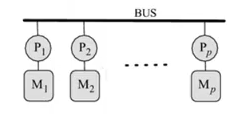
**Channel bandwidth-** the number of words a channel can communicate over a second

**Bisection bandwidth** is product between bisection width and channel bandwidth

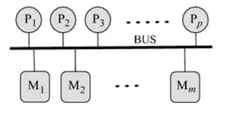
**BUS**

Bus- type of interconnection network

Bus topology



LMM



MMM- modular memory machine

Most nodes would wait for bus to be free, more time spent on communication rather than computation if we have too many PE’s, therefore it is not scalable.

Bus is not a regular interconnection network, as degree=1 for end nodes.

**RING**

A picture containing accessory

Description automatically generated

Ring is regular, degree=2

**2D MESH**

A picture containing sandglass

Description automatically generated

Each node has 4 neighbours except those along the borders

2d mesh- 48 nodes. Bisection width, split in half, remove 6 communication links-> bisection width

Chart

Description automatically generated with medium confidence

**2D TORUS**

Diagram

Description automatically generated

We turn a 2D mesh into a 2D torus by connecting its end nodes

Diagram

Description automatically generated

Every node has degree=4 since it has 4 neighbours therefore it is a regular interconnection network.

4 edges, 4 links. P number of nodes, therefore Cost = (p x 4)/2

3d torus- each node has 6 neighbours

**HYPERCUBE**

**A picture containing diagram

Description automatically generated**

**Number of nodes formula is as above for d- dimensional hypercube**

We can use binary digits to represent nodes

Diagram

Description automatically generated

2 binary bits for 2d hypercube

Can use 3 bits to label 3d hypercube

Diagram

Description automatically generated

Just draw 2 2d hypercubes and then connect them and add 0’s and 1’s in front of them accordingly

To make d dimension hypercube- use 2 d-1 dimensional hypercubes. Each label should differ by one bit with its neighbours.

e.g. 0010- 1010, 0110, 0000, 0011 (differs by 1 bit from original)

**MULTISTAGE NETWORK**

Multistage network- wherever the lines are joined, they are connected to those memory modules.

Cost is different from fully connected one in terms of number of switches needed to construct such networks. 8 x 8 example= 32 switches

8 x 8 fully connected- 64 switches

32- same connectivity as fully connected cross bar with less cost

**FAT TREE**

Getting num links- count how many links are at the root of the node.

In this example, **bisection width num links= 8 when you split the tree into 2.**

**Diagram

Description automatically generated**

Fat tree- called that because towards the root of the tree it has more links

Diameter= 2log(2) (num links)

Static fully connected- diameter is 1 as each node is connected to every other node, so that is the shortest distance between each pair of nodes and all possible pairs of nodes the max distance would still be 1 as they are all equal

Each node is connected to 2 link, each link is connecting 2 nodes therefore edges are counted twice therefore we are dividing by 2 therefore you get this cost.