MPI
and
Multi-Node Network

A. Sahu
Dept of CSE, IIT Guwahati

Outline

- Basic pf MPI
- MPI Constructs and Example
- Running programming in IITG HPC system
- Reference and Other Resources

How to compile and run on a Linux Machine

$mpicc hello_mpi.c –o hello_mpi

$mpirun –np 4 ./hello_mpi

4 copies of hello_mpi process will run

MPI Examples

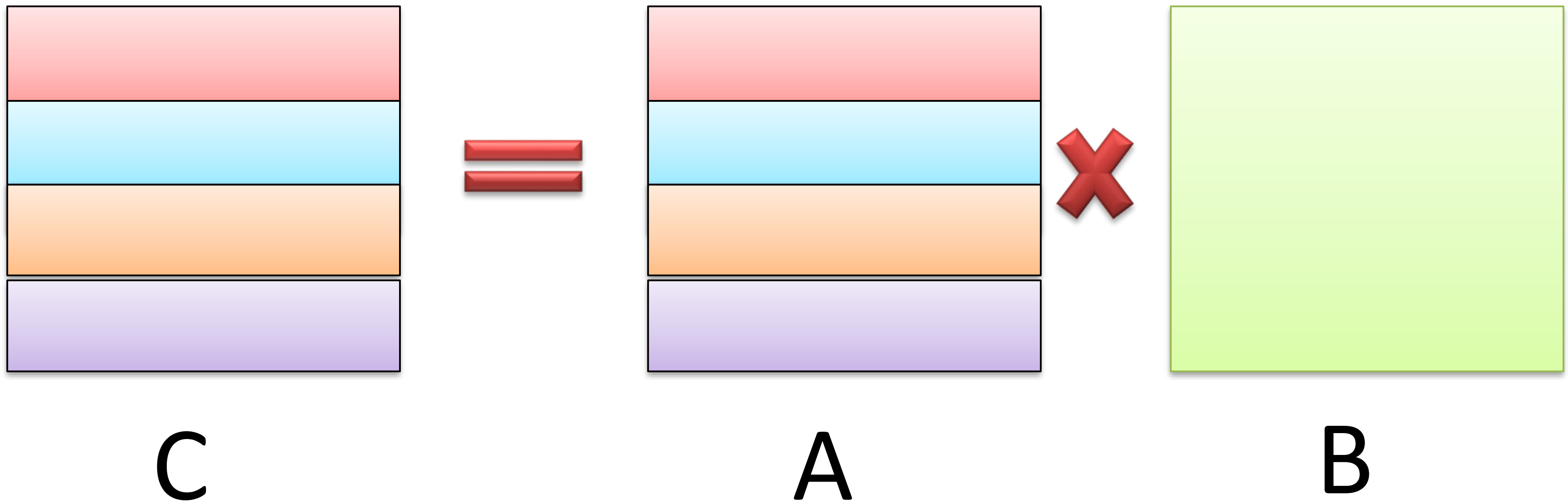
Example: Sum of N data

- Master Process
  - Data to be read by process 0 or MASTER
  - Divide the data in to N/M chunk size (N %M==0)
  - **SEND** respective chunk of data to other process
  - **Do local sum on each process (in master also)**
  - **RECV** sum of other process and calculate final sum
- Other Process
  - **RECV** data from Mater
  - **Do local sum on each process**
  - **SEND** local sum to MASTER

See the Code

Example: Matrix MUL

- c=axb: a[NRA][NCA], b[NCA][NCB], c[NRA][NCB]
- Work get divided: Based on Rows



Example: Matrix MUL

- c=axb: a[NRA][NCA], b[NCA][NCB], c[NRA][NCB]
- One Master Processor
- Many Workers, Assume NRA % NumWorker==0
  - Master divide the work between worker
  - Send respective rows of A and whole B to workers
  - RECV array C from all worker
- Every Worker
  - get some Row of A, Whole of B
  - calculate part of C
  - Send calculated C to Master

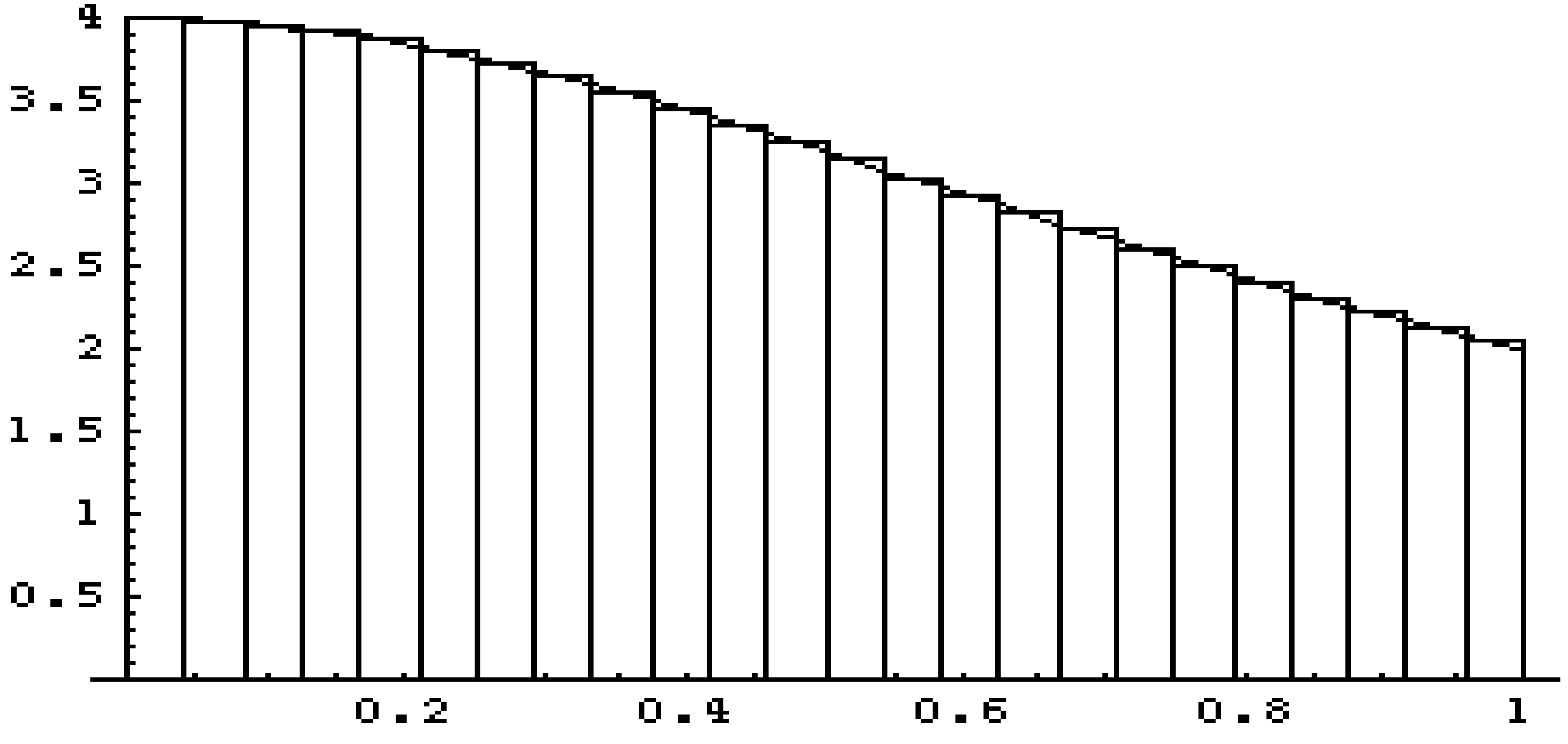
See the Code

Example: Compute PI

$$\pi = \int_0^1 \frac{4}{1+x^2}\, dx$$

Example: Compute PI

$$\pi = \int_0^1 \frac{4}{1+x^2}\, dx$$



How to write Program?

- Divide the range in to N interval/piece
  - Piece of size h = Range/N;
- Calculate area under each piece
  - Calculate the function value at piece X and multiply with piece size
  - h * F(X)
- Sum all the piece
  - $\sum_{i=1}^{n}$ h*F(X_i) with $X_i = R_{min}$+i*h

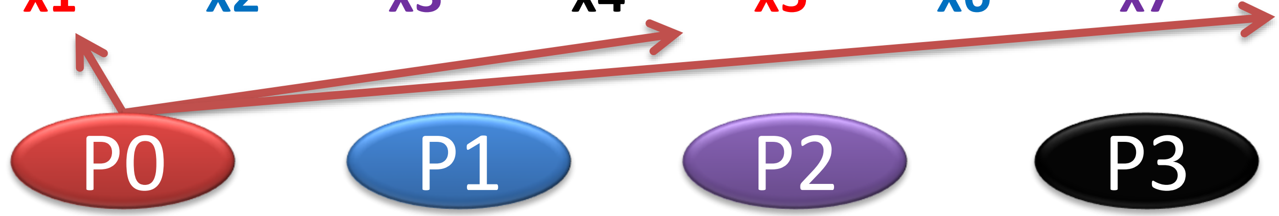
How to write Program?

```
printf("Enter Num intervals: ");
scanf("%d", &n);
h = 1.0 / (double)n;
sum = 0.0;
for (i=1; i<n; i++) {
  x = h*(i-0.5); Fx=4.0/(1.0+ x*x);
  sum = sum + Fx;
}
pi = h*sum;
printf("pi is approx %.16f", pi);
```

How to write Parallel Program?

- Divide the range in to N interval/piece
  - Piece of size h = Range/N;
  - **Suppose N =1000, NumProcessor =4**
- In Parallel: Calculate area under each piece

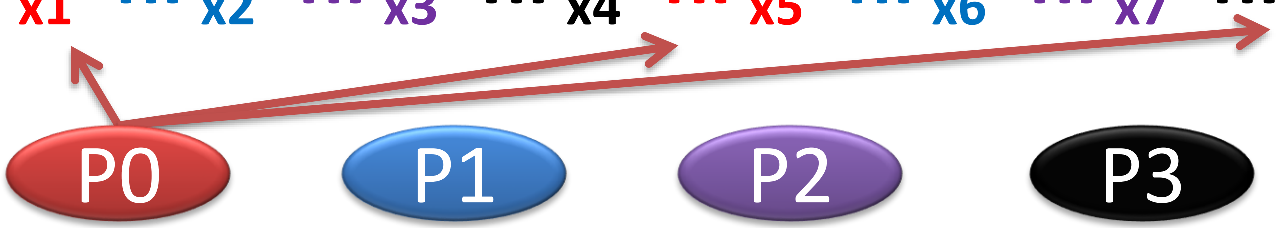
$$hF_{x1}+hF_{x2}+hF_{x3}+hF_{x4}+hF_{x5}+hF_{x6}+hF_{x7}+hF_{x8}+..+hF_{x999}$$



How to write Parallel Program?

- Divide the range in to N interval/piece
  - Piece of size h = Range/N;
  - **Suppose N =1000, NumProcessor =4**
- In Parallel: Calculate area under each piece

$$hF_{x1}+hF_{x2}+hF_{x3}+hF_{x4}+hF_{x5}+hF_{x6}+hF_{x7}+hF_{x8}+..+hF_{x999}$$



- $(hF_{x1}+hF_{x5+}+..+hF_{x997}) + (hF_{x2}+hF_{x6}+..+hF_{x998})+$
  $(hF_{x3}+hF_{x7}+..+ hF_{x999})+ (hF_{x4}+hF_{x8}+..+hF_{x996})$

Example: Compute PI

```
#include <mpi.h>
#include <math.h>
int main(int argc, char *argv[]){
  int n, myid, Nproc, i;
  double lsum, pi, h, sum, x, a;
  MPI_Init(&argc, &argv);
  MPI_Comm_size(MPI_COMM_WORLD, &Nproc);
  MPI_Comm_rank(MPI_COMM_WORLD, &myid);
  if (myid == 0) {
   printf("Enter Num intervals: \n");
   scanf("%d", &n);
  }
  MPI_Bcast(&n, 1, MPI_INT,0,
                  MPI_COMM_WORLD);
```

Example: Compute PI

```
h = 1.0 / (double)n; sum = 0.0;
for (i=myid+1; i<=n; i+= Nproc) {
     x = h*((double)i - 0.5);
     sum += 4.0 / (1.0 + x * x);
}
lsum = h*sum;
MPI_Reduce(&lsum, &pi, 1, MPI_DOUBLE,
      MPI_SUM, 0, MPI_COMM_WORLD);
if (myid == 0)
   printf("pi is approx %.16f\n", pi);
MPI_Finalize();
return 0;
}
```

IITG HPC clusters: Spec

- 4 login nodes
- 126 compute node
- 16 GPU compute nodes
- 16 Phi compute nodes
- Total 126+16+16= 158 nodes
  - Each node 12 cores * 2 threaded
  - Effective 24*158 =3792 cores

Running MPI program on IITG HPC clusters

- Logic to one login nodes : non GPU/PHI
  - param.-ishan.iitg.ernet.in (172.17.0.7)
- Compile MPI-code

Running MPI program on IITG HPC clusters

- Logic to one login nodes : non GPU/PHI
  - param.-ishan.iitg.ernet.in (172.17.0.7)
- Compile MPI-code
- Run using srun or sbatch
  - In s batch specify number of node, task per node
  - Total process
- SLURM : Simple Linux Util for Resce Mngt
  - Scheduler the JOB efficiently, user need not to worry where it is scheduling

Resources

- https://computing.llnl.gov/tutorials/mpi/
- V. Kumar, A. Grama, A. Gupta, and G. Karypis. ***Introduction to Parallel Computing****: Design and Analysis of Algorithms*. Benjamin-Cummings Publ. Co, 1994 **[metis software]**
- Michael J. Quinn. ***Parallel Programming in C with MPI and OpenMP*****.** McGraw-Hill Education Group. 2003.
- Joseph JáJá. ***An Introduction to Parallel Algorithms***. Addison Wesley Longman Publishing Co., Inc.,, USA. 1992

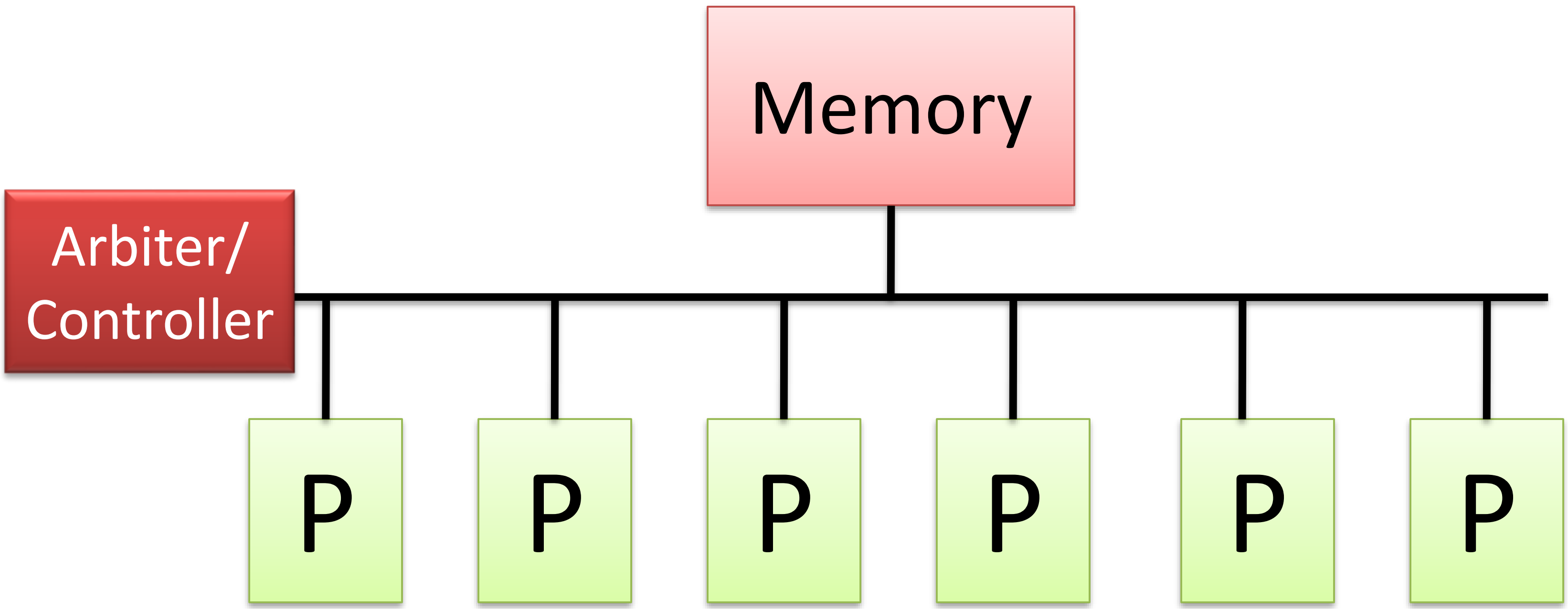
Multi-Node Architectures and Topology Embeddeding

Outline

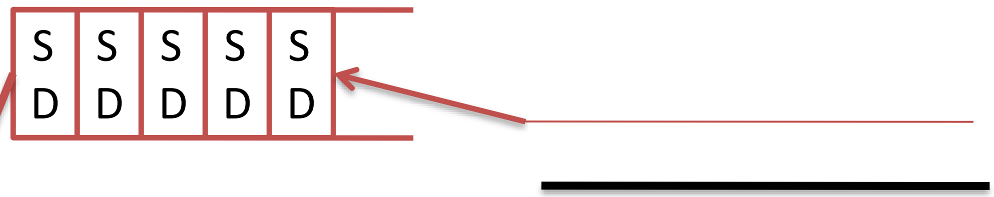
- Multi-node Architecture
- Interconnection and Topology Embedding
- Programming : MPI
  - To be taught after Mid-Sem : 2 classes
- Scheduling Concepts
- Independent Tasks, Dependent Tasks

Computer Interconnection Network

Bus interconnection/Shared Memory

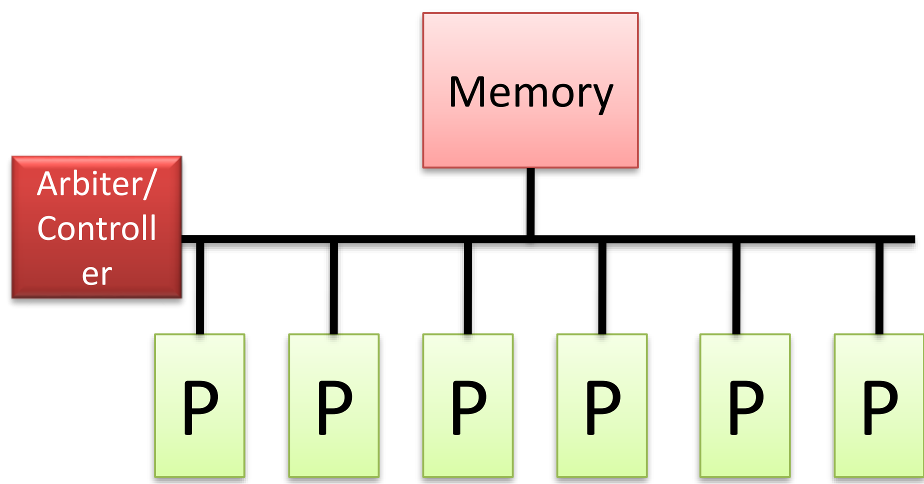


BUS Protocol: Queue Based



BUS Arbiter

Grand based on Policy/Priority to one S/D of Queue



S (Source) and D (Destination) may be Processor/memory

FCFS is Common Policy

Verdict : Share BUS

- Utilization saturates with number of requests
- Saturate more quickly as processor increases

- So it is not scalable with number of processor
- If number of processor > (8 or 10), Bus interconnection is Bad

Large Collection of Computer

- Connected Using Network
- Example Grid System
  - Geographically different location
- Data Center
  - Many Container //Static N/W
  - Many Racks in a Container // Static N/W
  - Many Chassis/Rack-Server in a Rack // Static N/W
  - Many Servers/Socket/Processor in a Chassis/Rack-Server //QPI or BUS
  - Many cores in a Socket/Processor : QPI/Fully Connected/ BUS
  - Many HW-threads in a Core

PARAM ISHAN

Switched Networks

BUS	Switched Network
• Shared media	• Switched paths
• Lower Cost	• Higher cost
• Lower throughput	• Higher throughput
• Scalability poor	• Scalability better

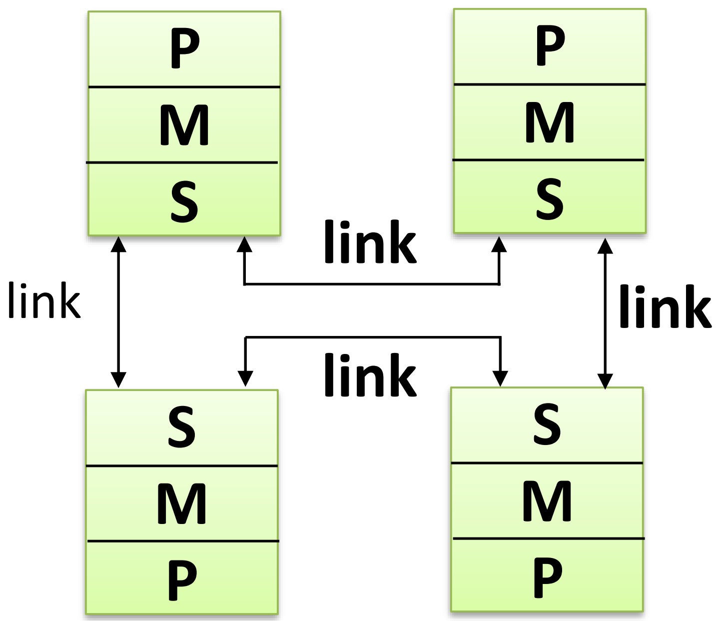
Interconnection Networks

- Topology : who is connected to whom ?
- Direct / Indirect : where is switching done ?
- Static / Dynamic : when is switching done ?
- Circuit switching / packet switching : how are connections established ?

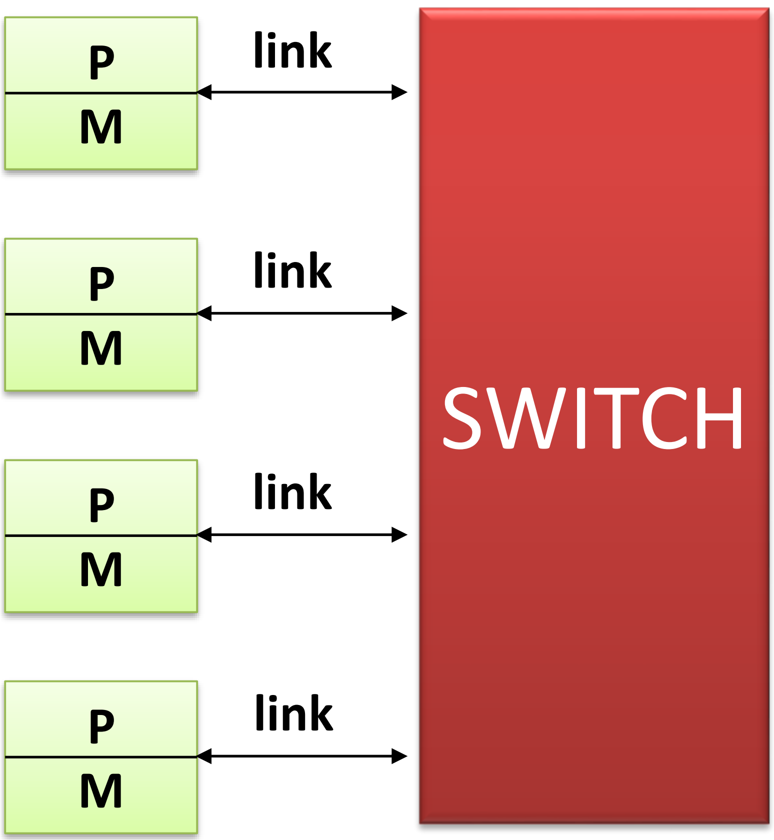
Interconnection Networks

- Store & forward / worm hole routing : how is the path determined ?
- Centralized / distributed : how is switching controlled ?
- Synchronous/asyn : mode of operation?

Direct and Indirect Networks



DIRECT



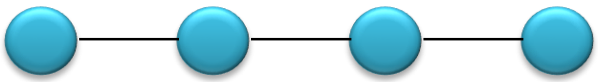
INDIRECT

Static and Dynamic Networks

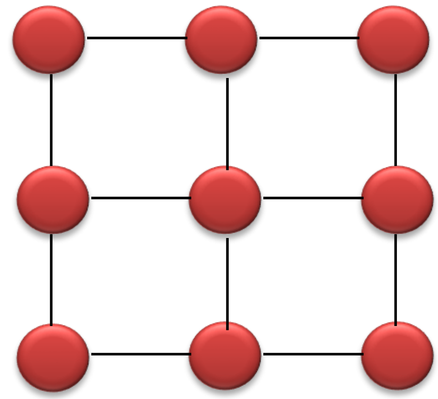
- Static Networks
  - fixed point to point connections
  - usually direct
  - each node pair may not have a direct connection
  - routing through nodes
- Dynamic Networks
  - connections established as per need
  - usually indirect
  - path can be established between any pair of nodes

Static Network Topologies

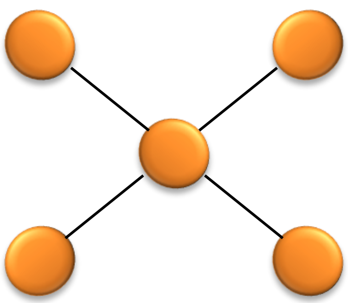
Non-uniform connectivity



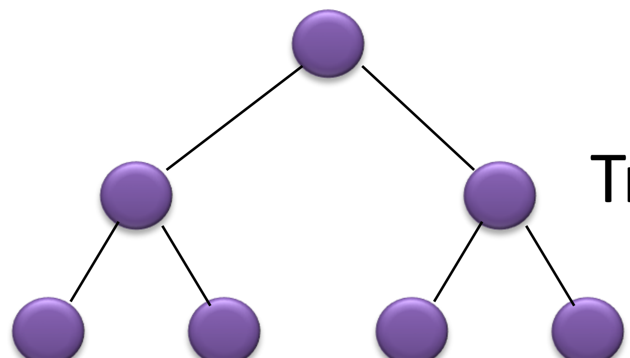
Linear



2D-Mesh



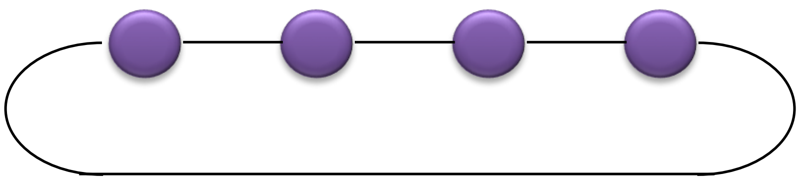
Star



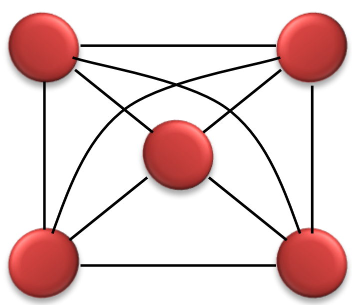
Tree

Static Networks Topologies- contd.

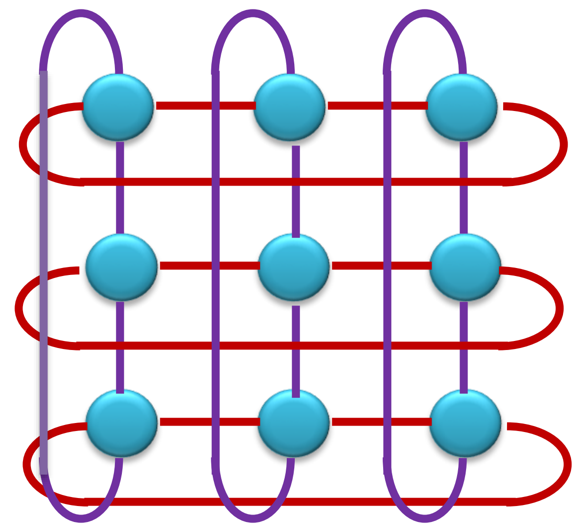
Uniform connectivity



Ring



Fully Connected



Torus