Cryptography
Steganography

CIA → Confidentiality Integrity Availability.

Keys → Symmetric
→ Asymmetric

AES algo for encryption
↳ Advanced Encryption Standard.
IDEA

DMPG

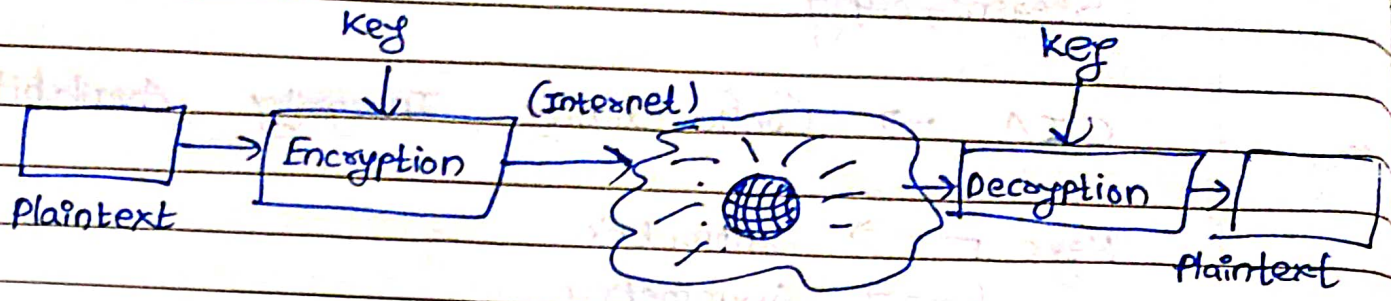
Stages of DLM
1) Data acquisition.
2) Data storage
3) Backup and Recovery.
4) Data Management/sharing
5) Data usage
6) Data retention & destruction

HPC

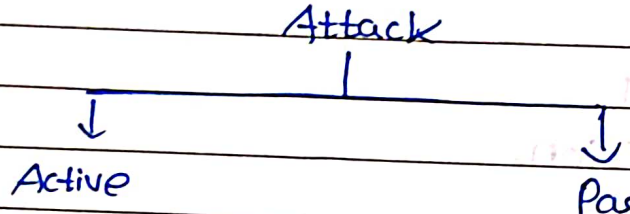
Intro to Parallel Computing
Ananth Grama, Anshul Gupta, George Karypis & Vipin Kumar.

Compute followed by communication is 11 processing.

Model of Network Security



Key(encryption) = key(decryption) ⇒ Symmetric Key.
Key(encryption) != key(decryption) ⇒ Asymmetric (Public) Key.



HPC

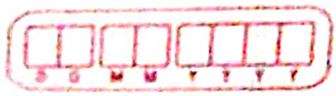
SPMD ⇒ Single Program Multiple Data.

Dichotomy of ||el Computing Platforms
① Logical organization → Programmer's view.
② Physical organization → Actual hardware organiz^n.

T-com ⇒ Time required for process communicat^n.

Granularity = $\frac{\text{Computation}}{\text{Communication}}$

PE → Processing Element.
IC → Interconnection Network.
PRAM → Parallel Random Access Machines



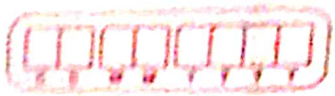
Coarse Grain → Small number of large sized tasks.
→ Lower Communicat^{n}.
Fine Grain → Large number of small sized tasks.
→ Higher Communicat^{n}.
→ Load balancing.

Memory Architectures
- UMA → Uniform Memory Access
- NUMA → Non-uniform -11-

PRAM
- EREW → Exclusive Read Exclusive Write → More towards sequential
- CREW → Concurrent Read Exclusive Write
- ERCW → Exclusive Read Concurrent Write.
- CRCW → Concurrent Read Concurrent Write.

- Locality of Reference.

PE → Processing Element.
IC → Interconnection Network.
PRAM → Parallel Random Access Machines



Coarse Grain → Small number of large sized tasks
→ Lower Communicat^n.

Fine Grain → Large number of small sized task.
→ Higher Communicat^n.
→ Load balancing.

Memory Architectures { UMA → Uniform Memory Access
NUMA → non-uniform -" -

PRAM

- EREW → Exclusive Read Exclusive Write → More towards sequential
- CREW → Concurrent Read Exclusive Write
- ERCW → Exclusive Read Concurrent Write.
- CRCW → Concurrent Read Concurrent Write.

- Locality of Reference.

29/07/24 Monday

DMPG (MOD 2) → Data storage & data availability.
HDD, SSD.
SATA Device.

- What is Data Center?
  - DAS → Direct Attached Storage
  - NAS → Network -11- -11-
  - SAN → Storage Attached Network.

DAS Interface
SCSI
SAS
SATA
PCIe

CNS

Attacks

Passive
- Release of msg content
- Traffic analysis

Active
- Masquerade
- Replay
- Modification of msg.
- Denial of service

HCI

WIMP → Windows Icons Menus Pointers.
Interface or
Windows Icons Mice & pull-down menus.

29/07/24
Tuesday

CENS

Classical Encryption Techniques

Plaintext, Ciphertext, Enciphering (Encryption),

Secret key → Symmetric.
Public key → Asymmetric.
Cryptographic algorithms → Ciphers.
Cryptanalysis → Science of studying attacks against cryptographic systems

⊙ Ciphers

Symmetric Cipher — Asymmetric Ciphers.

Block cipher
↓
encrypt data block of plaintext at a time (64 or 128 bit)
↑ DES ↑ AES.

Stream Cipher
↓
encrypts data one bit or byte at a time.

① Symmetric Encryption.

$Y = E_K(X)$ or $Y = E(K, X)$

algo.

Key

Plaintext.

$X = D_K(Y)$ or $X = D(K, Y)$

Classical Cryptography

Transposition ciphers

Substitution ciphers

Combination is product ciphers.

- All modern ciphers are "product ciphers".

DMPG

Storage Network Industry Association (SNIA)

RAID types

→ H/W RAID

→ S/W RAID

HCI Activity

Word processor for blind people.

Design a UI for word processor for blind people.

HPC Lab.

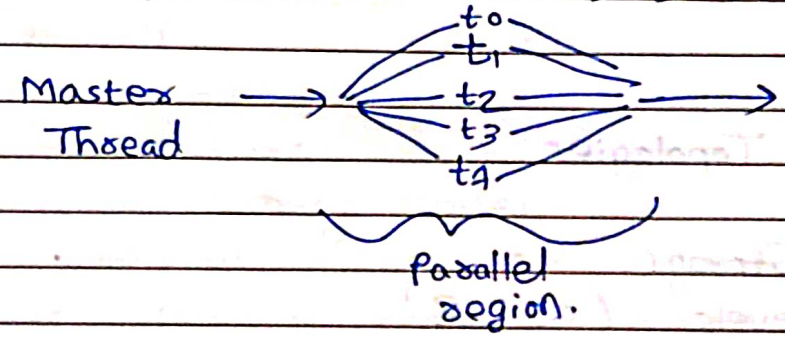
OpenMP

MPI

Cuda

⊙ OpenMP.

- portable shared memory programming
- pragmas for C/C++
- parallel section has to be defined by developer.



omp_get_num_threads() → Total no. of threads.

omp_get_thread_num() → thread no. for specific thread.

```
# pragma omp parallel
{
    // parallel section.
}
```

To set no. of threads:

1. export OMP_NUM_THREADS=2 // env. variable.
2. In program.

Q.1)

#include <stdio.h>
#include <conio.h>

int su

21/07/24
Wednesday

DMPG

- Storage attachment strategies
  - DAS
  - SAN
  - NAS
- Network Topologies
- VM's
- Cloud Storage
- RAID levels (0, 1, 5).
  - RAID 0 → Stripping.
  - RAID 1 → Mirroring.
  - RAID 5 → Stripping with parity.
  - RAID 6 → Stripping with distributed parity.

(Explain any 2 RAID levels given)

- Storage pooling
  - Primary Storage Pool.
  - Copy Storage Pool.
- Storage Provisioning.

HPC

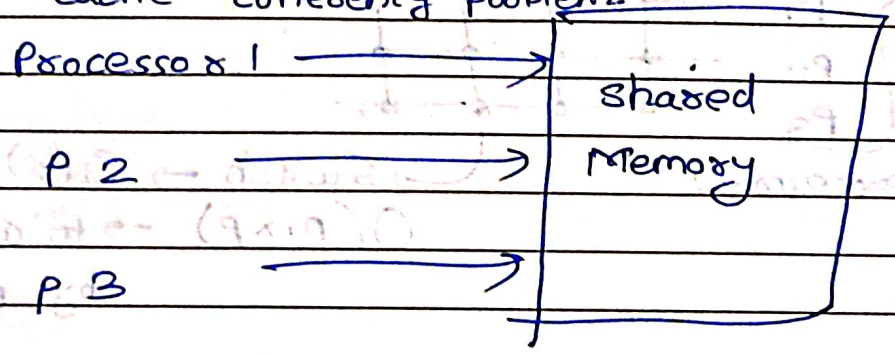
PRAM → Parallel Random Access Machine.

also stands for no of processors.

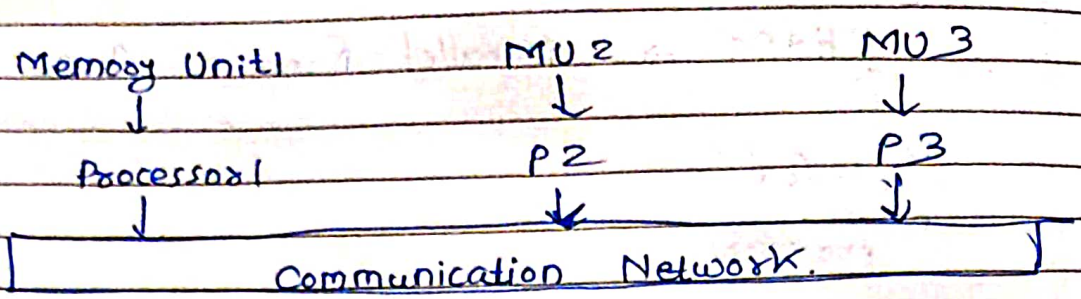
PRAM Model Implementation.

- Shared memory model (OpenMP)
- Message passing model (MPI)
- Distributed memory model

① Shared memory model
- emphasizes on control parallelism.
- share a common memory space.
- cache coherency problem.

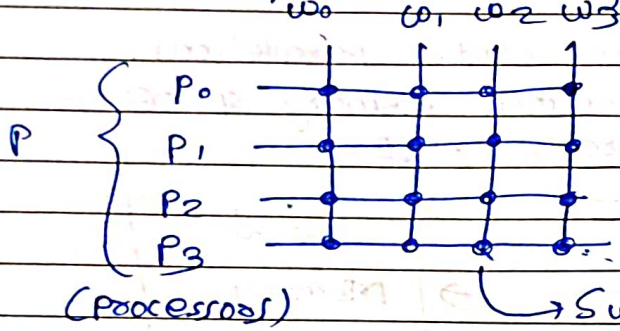


② Message Passing Platform
- MPI
- distributed memory approach.
- interaction through message passing.



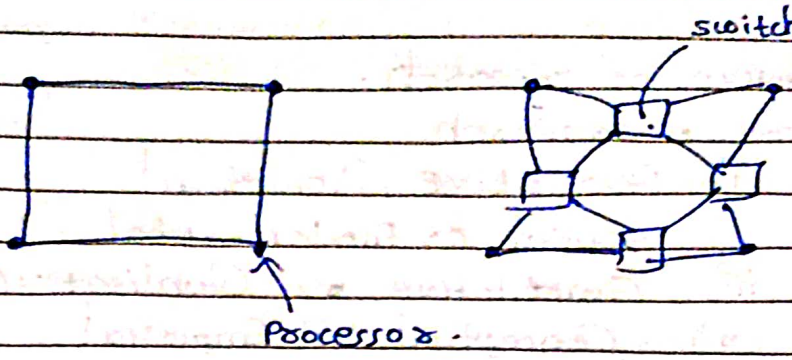
(5) Distributed Memory.
- processors have their local memory.
- no issue of cache coherency.
- explicitly need to define sub-tasks by programmer.

- Physical Complexity of 11^{el} computer.



→ Switch → O(1)
O(m × p) → to access memory by processor.

- Interconnection Network.
  - made up of switches & links.
  - carry data betⁿ processor & memory.
  - static / dynamic.
    - ↓ Direct network
    - ↓ Indirect network



HCI

Usability goals & measures:

1. Time to learn.
2. Speed of performance.
3. Rate of errors by users.
4. Retention over time.
5. Subjective satisfaction

ERP Evaluation

1 30 m.
2 60 %
3 very high.
4 90 %.
5 40 %.

RM.

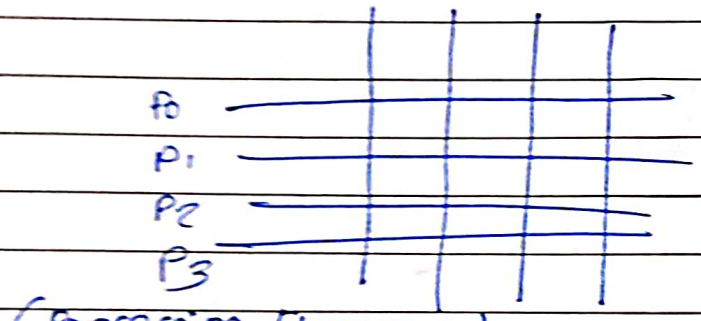
- Objectives of research.
- Types of research.
  i) Descriptive / Analytical
  ii) Applied vs Fundamental
  iii) Quantitative vs Qualitative.
  iv) Conceptual vs Empirical.

9/10/24 Thursday

HPC

- Network Topologies (Static / Dynamic).
- Topologies tradeoff performance for cost.

- Crossbars network topologies



(p x m)

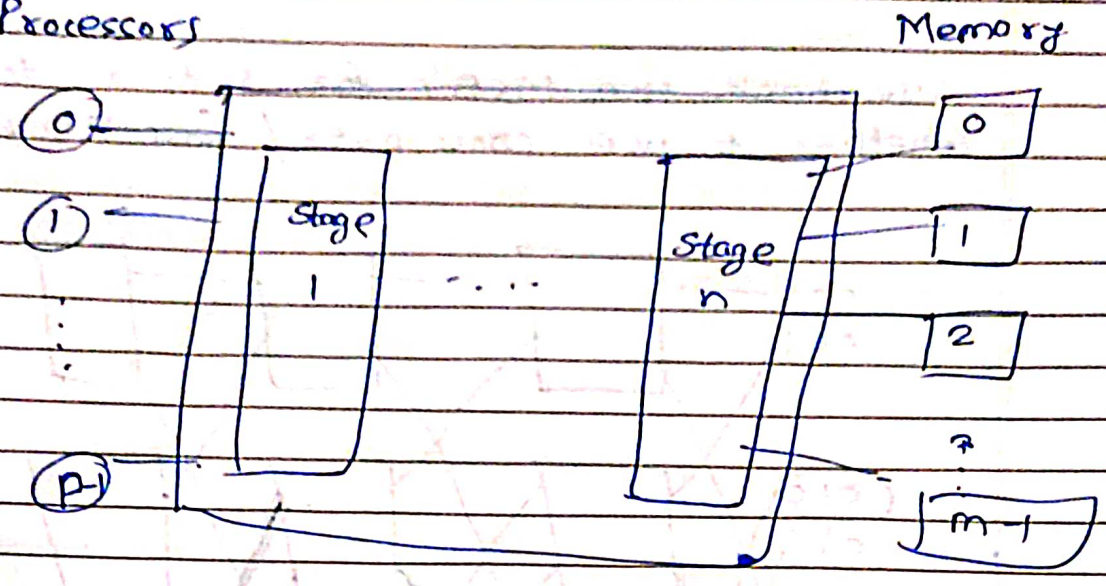
cost complexity = $O(p^2)$ (crossbar)

Networks

- Blocking
- Non-blocking.

Cost of switch = (degree of switch)^2

• Multistage Interconnection Network (MIN)



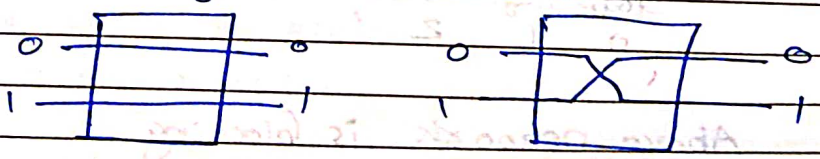
- Omega network (most commonly used MIN)
- $\log_2 P$ stages
- i/p i is connected to o/p j

$$j = \begin{cases} 2i & 0 \le i \le P/2 - 1 \\ 2i+1-P & P/2 \le i \le P-1 \end{cases}$$

$P = 8,\ i = 4 \Rightarrow \text{stages} = \log P = \log_2 8 = 3.$

Perfect Shuffle patterns connected using 2x2 switches.

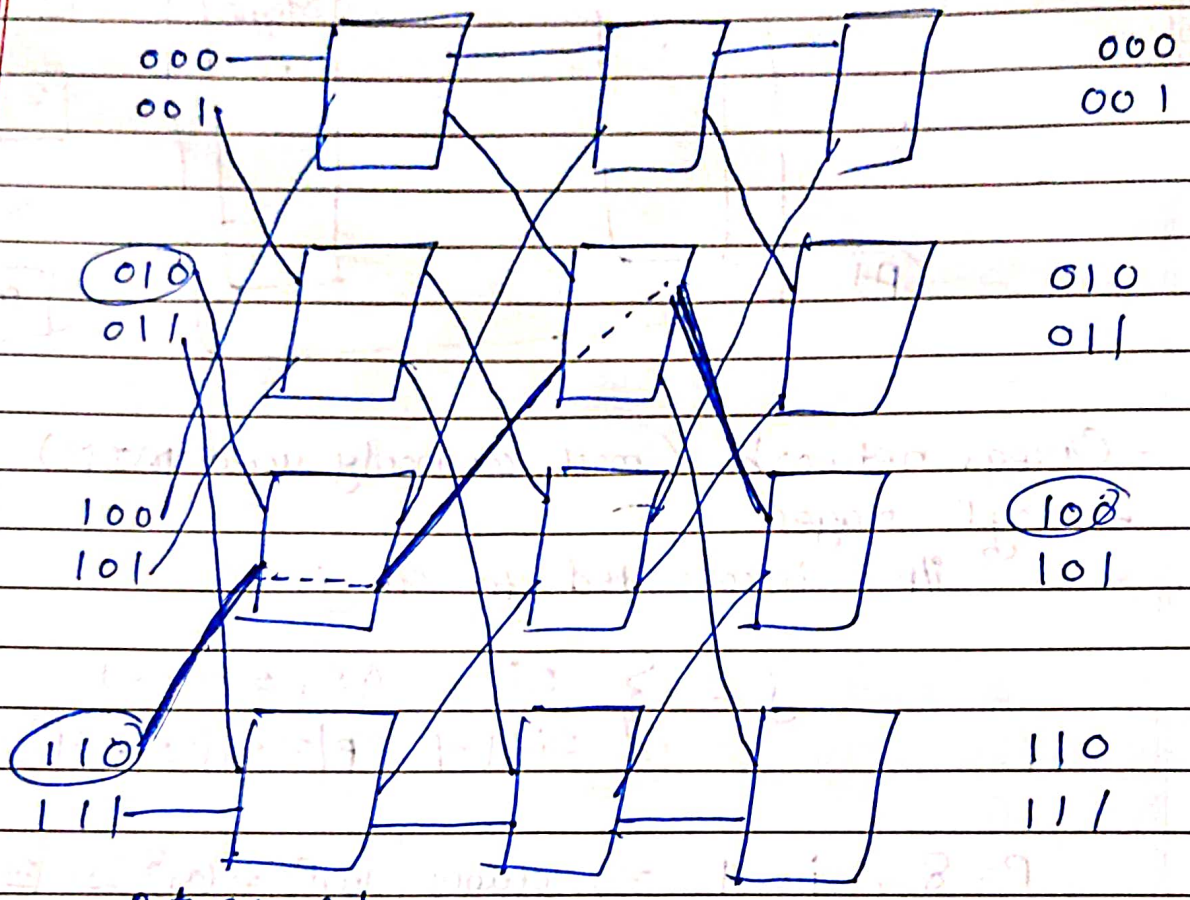
2 modes → Pass through, Cross over.



n(switching nodes) = $P/2 \times \log P$

s → binary repres^ of source.
d → —" — dest^n.

We check the MSB's for s & d to decide whether to use cross-over / pass-through.



pass through.
cross over

source → 1 1 0
dest → 1 0 0

stage 0 stage 1 stage 2

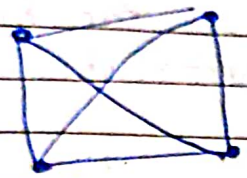
Cost complexity = $O(p \log p)$

Above network is blocking

Crossbar network
$O(p^2)$

multistage network
$O(p \log p)$

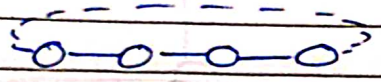
- Completely connected network:



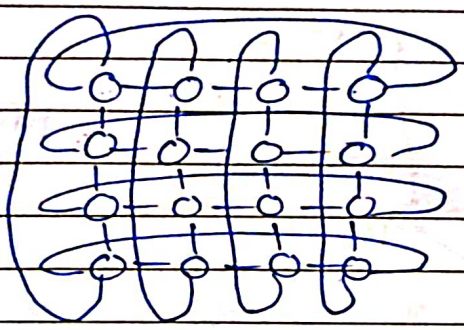
$\Rightarrow 6$

$\frac{8 \times 7}{2} = 28.$

1-D- torus



2-D -torus



(2-D mesh with wrap around link)

$p = 16.$

02/08/24 Friday

- Dense or Sparse architecture.

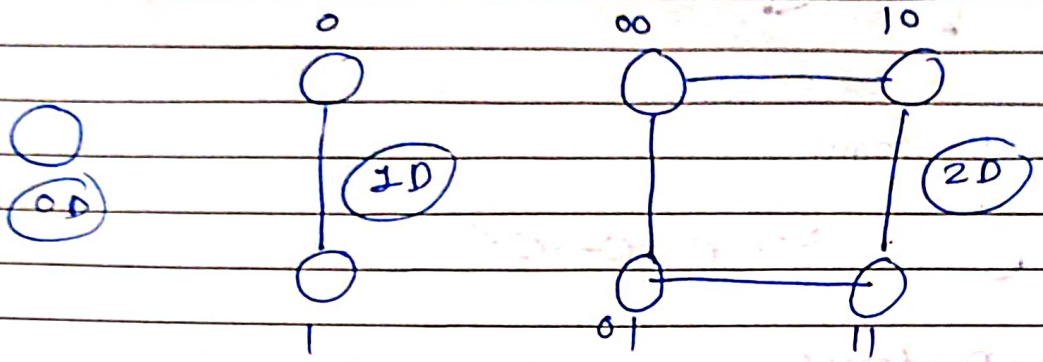
$T_s \rightarrow$ Time for sequential
$T_p \rightarrow$ Time for parallel.

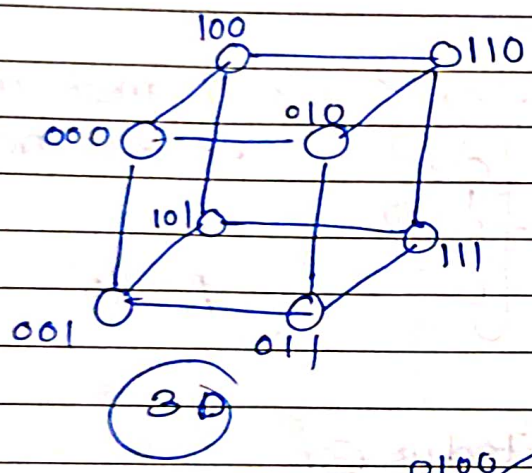
- Parallelism go hand in hand with Concurrency.

if ($T_p > T_s$)
    use parallel
else
    use sequential

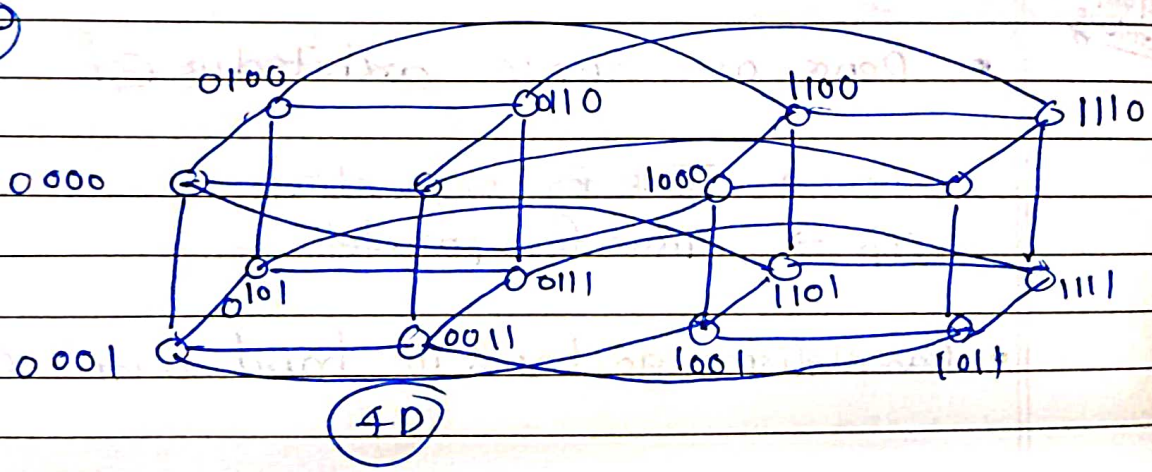
⊙ Hypercube.

- Every D dimentional hypercube has 2^D nodes.
- Every hypercube with $D > 0$, can be formed by connecting 2 D-1 hypercubes.





$$\text{links} = \frac{P \times \log P}{2}$$



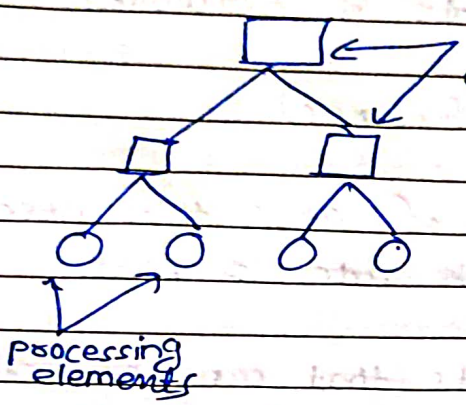
Dist^n bet^n P's is bit difference (Hamming distance)

Farthest dist^n bet^n 2 nodes = $\log_2 P$

e.g. (0000, 1111) = 4 links

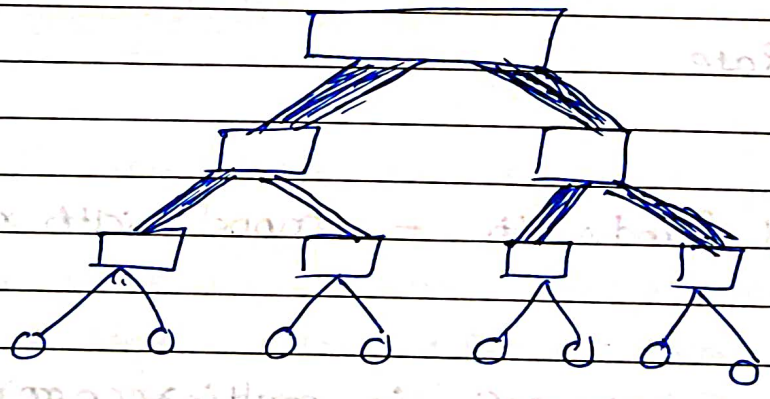
$\log_2^{16} = 4$

⊙ Tree-Based Network.



Drawback: Traffic through root only.

Fat Tree:



Q How to evaluate architecture

Diameter → dist^n bet^n 2 farthest points

Linear arr = $p-1$

mesh = $2(\sqrt{p}-1)$

tree = $\log p$

hypercube = $\log p$.

complete connected = $O(1)$.

Bisection Width → min wires to be cut to divide network into 2 networks.

Cost → no. of links (no. of wires)

Arc Connectivity → min. no. of arcs that need to be removed to divide network into 2 parts.

Channel width → no. of bits that can be communicated over a channel simultaneously, bet^n 2 nodes.

Peak Rate

Channel Bandwidth = channel width x peak rate.

Cache Coherence in multiprocessor system

- Every processor may have different cache copy

HCL → MS Dynamics, SAP

06/08/24 Tuesday

DMBG

Cloud Storage
- low cost high bandwidth.
- CSP's (Cloud Storage Providers)
- Public & Private clouds.
- Safety
- Cloud Types:
  1) Public 2) Private 3) Hybrid 4) Community
  5) Mobile
- Functionality
  1) Syncing 2) Enhanced Security
  3) Collaboration Tools 4) Space efficiency
  5) Disaster Recovery 6) Pay-as-you-Go Cost)

07/08/24 Wednesday Cache Coherency

08/08/24 Thursday

HPC.

Message Passing Costs in Parallel Computers.

1) Startup time (ts) → only once.
2) Per hop time (th) → at each hop, latency.
3) Per-word transfer time (tw) → m(words) x tw.

⊙ Store & forward Routing

$$t_{comm} = t_s + (mt_w + t_h)l.$$

↑ multiplicative term.

where, t_{comm} ⇒ Communication time.
t_s ⇒ Startup time.
m ⇒ no. of words.
t_w ⇒ per word transfer time.
t_h ⇒ per hop time.
l ⇒ no. of links traversed.

If t_h is very small then,

$$t_{comm} = t_s + mlt_w.$$

21/08/09

Human Relations at work

Father of mgmt studies - Fredrick Taylor.
Henry Fayol - 14 principles.

1) Division of work
2) Authority
3) Discipline
4) Unity in command.
5) Unity in direction
6) Alignment of personal & general interests
7) Renumeration
8) Centralization.
9) Scalar Chains
10) Order
11) Equality

HPC

⊙ Packet Routing
- message is broken down into packets & pipeliner them through the network
- errors checking, sequencing & header information.

total communication cost $= t_{comm} = t_s + t_h l + t_w m$.

$t_h l$ ↑ additive term.

- Disadv: Each packet will carry more information

⊙ Cut-through Routing
- message divided into flits (basic unit)
- flits are too small, thus also header.
- traces msg decides path & all flits are routed in same path.

$t_{comm} = t_s + t_h l + t_w m$.

t_w is very small

if l is large, use cut through routing.
if l is smaller, use store & forward (overhead is less)

- Simplified cost model for communicating messages:

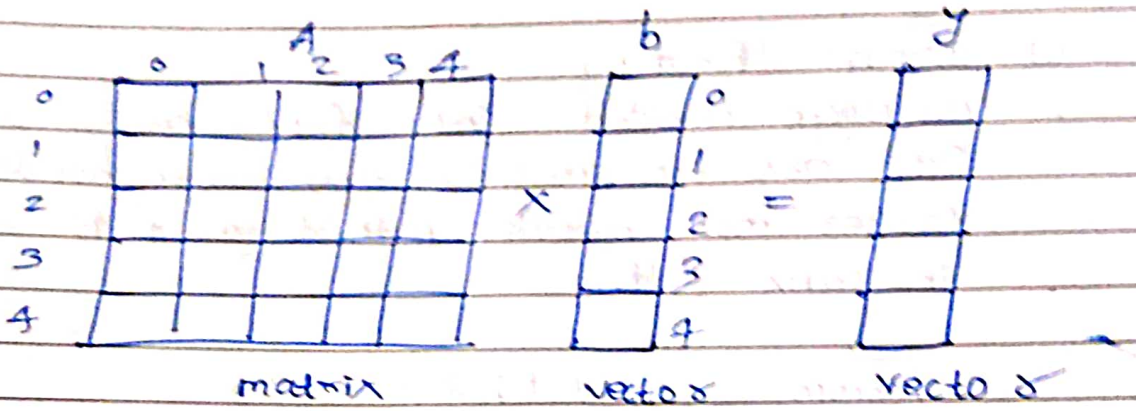
  $t_{comm} = t_s + lt_h + t_{wm}$ ($t_s + t_w m$)

(Mod 1 Complete)

Mod 2 Principles of Parallel Algorithm Design.

(Authors: Ananth Grama, Anshul Gupta, George Karypis & Vipin Kumar)

Step 1: Decompose the task into sub-tasks.



Here, decomposition can be done in 'n' no. of ways.
Tradeoff bet^n memory & communication overheads.

- Maximum degree of concurrency: no. of tasks (max) running parallely

- Algorithms differ for dense & sparse matrices.