Parallel & Distributed computing

Monday 6

★ Divide & conquer

Parallel
↓
serial vs parallel communication

→ Parallel computing
↓ execution of single task by multiple processors at the same time.
→ Multi core (Processing chips)
→ Multi processor
ex: google

Distributed
h/w or s/w components located at networked comp. communicate & coordinate only by passing msgs
Ex: online class

computing
↓
process to complete a goal-oriented task using computer.
↳ Include design & development S/w & H/w
For what →
processing & managing Information

Tuesday 7

Why use parallel computing?

Wednesday 8

→ Reduce time for
→ Save time & Money
→ More complex problems
→ provide concurrency
→ advantage of Non-local resources

Why uses distributed computing
→ Functional
→ Economical
→ Reliable
→

August 2015

T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
			30	31																										

Thursday 30

Challenges in Distributed Systems

1. Hetrogeneity:-
    - networks → hybrid cloud, Private, public → solution: same protocol for all networks
    - Computer hardware

Friday 31

- OS
- Programing Laguages
- Implementation by differet developers
- Mobile code
- Middle ware

2. Fault Tolerance:-
    →Dependable

Saturday 1

Attributes
- Avalability
- Reliability
- safety
- Maintainability

Sunday 2

Types of Failure
1. Omission Failure
    - Recieve omission
    - send omission
2. Timing Failure
3. Arbitary Failure

Distributed Memory Programming Models

Thursday 6

⇒ Memory Model:-

↳ how processes / threads view and access memory in parallel / distributed system

Friday 7

shared ↙ ↘ Distributed → data sharing through message passing

→ Distributed Memory:-

Each processor (node) → its own local memory
Data not shared directly → it is explicitly communicated through message passing.

★ Each process has two types of data

1. Local data   2. Remote data

Saturday 8

⇒ Architecture

↓ Components

- Multiple Independent nodes / computers
- Each has its own CPU, memory, storage
- Interconnected through high-speed LAN

Sunday 9

August 2015

T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
			30	31																										

Monday 10

⇒ How process communicate?

processes exchage data through

• Message Passing

⇓

Library for writing application program — implemented using APIs

• MPI

Tuesday 11

designed for hetrogenous Machines ← • PVM (Parallel Virtual Machine)

• Socket or Remote procedure calls

⇒ Message passing Mechanism

• send • recieve ⇒ both can be blocking / non-blocking

⇒ Strategies to divide data:-

Block distribution (divide block per process)

Wednesday 12

Cyclic distribution [Each process gets nth element

Block-cyclic [Hybrid collection of both]

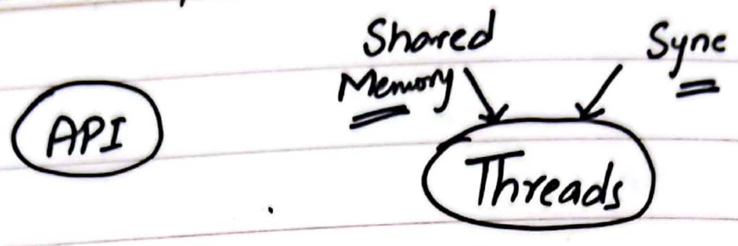
September 2015

T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S
			27	28	29	30																								
					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

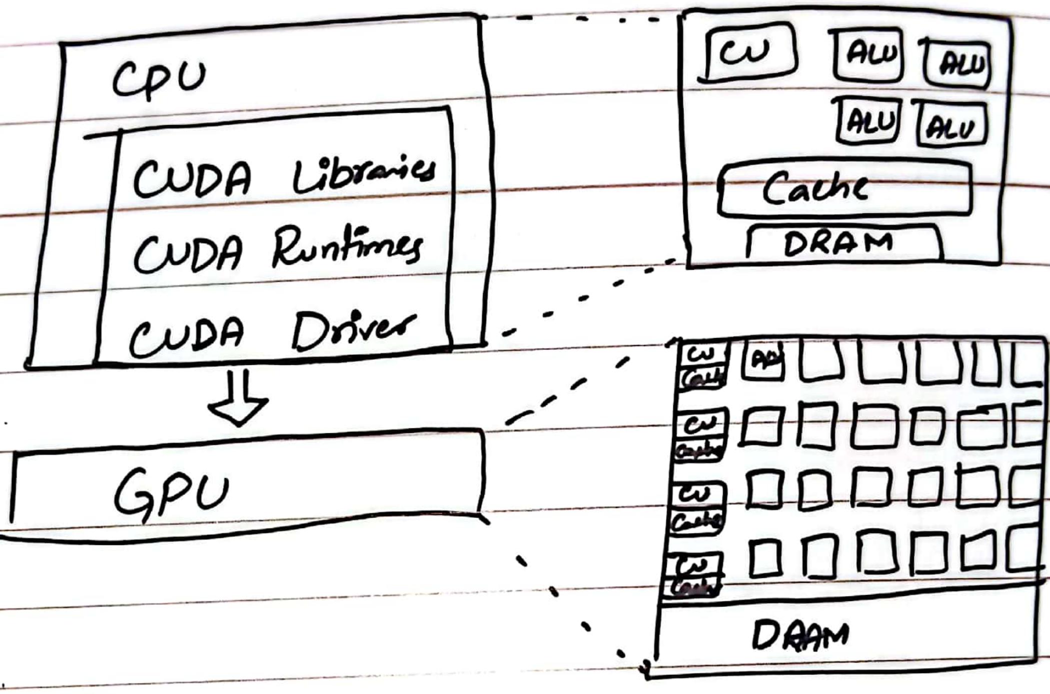
Thursday 13

CUDA (Compute Unified Device Architecture)

Parallel Computing Architecture



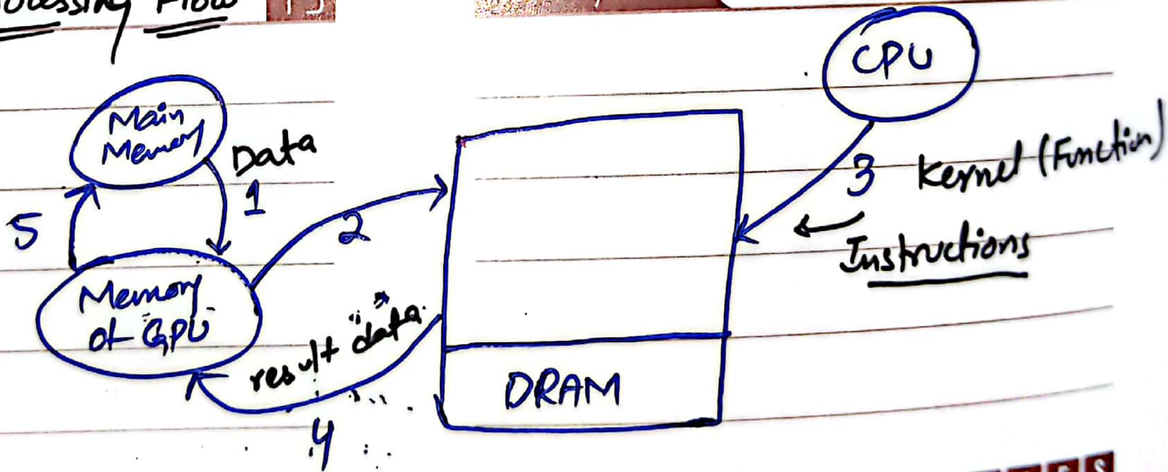
Friday 14



Saturday 15

Processing Flow

Sunday 16



Lec #2

→ Amdhal's law ←

↳ speed up → used in parallel computing

→ calculates theorotical speedup in latency of the exeution of a task

★ scalability :-

↳ handle more work by adding more resources

↳ Types :-

→ scale in / scale out (H) } Adding or removing new nodes

↓ scale up (V) / scale down } Adding or removing resources to a single Node communication cost depend on:

i - Programing Model for communication
ii - Network topology
iii - Data handling and routing
iv - Associated network protocols

=> Message passing cost in parallel computers:

i - Startup time (t_s)
ii - Per-hop time (t_h)
iii - Per-word transfer time (t_w)

=> Message Passing:

i - store and forward routing
ii - Packet routing
iii - cut-through routing

Decomposition Tehniques

1- Recursive:-

2- Data Decomposition:-

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \times \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

$\downarrow$ Data set 1 $\qquad$ $\downarrow$ Data set 2 $\qquad$ Task set

★ important point:-

Different data set

same operations

3) Exploratory Decomposition:-

Decomposition goes hands in hand with problem exeution.

Example:- puzzle problem

3	1
2	—

→

1	2
3	—

4) Speculative Decomposition:-

Action to be taken is based on ~~p~~ output of preceeding part

Example:-

switch case

Topic:

Memory Consistency Models

↓

defines how data updates are shared and viewed across multiple nodes.

It set rules for synchronization.

Approaches

strict Relax

↓

reliability, availability, performance

Types of consistency

- strong
  - ↓ all writes operations are updated on all node instantanously
- sequential
  - ↓ defines a contract between programmer and architecture to define the order and sequence of operations
- casual
  - → W/A GcH
  - → Feed (insta / Fb)
- Monotonic

Monotonic Read:

If you read a value from the system, next time you read it either get the exact same one or the recent updated one.

Monotonic Write.

Once a write happens, all future writes happen in the correct order.

e.g:

If you send a message, system ensure to not reverse the order of your updates.

Topic:-

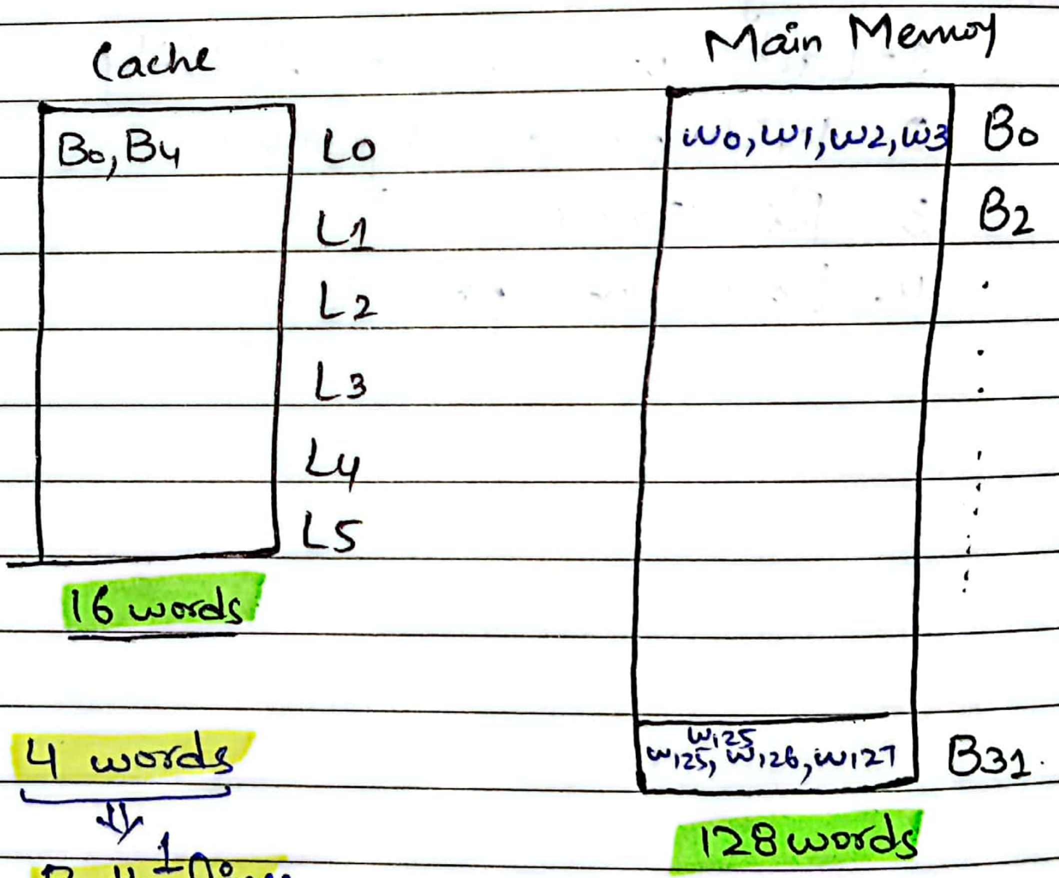
Cache Memory Mapping

* CPU directly connected to cache

MIPS 10^6 GIPS 10^9

RAM → words → min addressable unit of memory

Direct Mapping



4 words
↓
Both 1 lines and 1 block size

K mod n

K ↓ Block no.

n ↓ no. of lines

Physical Address

Block No	Block offset
5	2

0-127 => 7 bits

3 bit	2 bit	2 bit
Tag	line no	Block offset

Tag ↓ which block in line