Computer Architecture is set of ... that describes functionality, organization & implementation of computer system.

History of computer?

1960 Mainframe

- Classes of Computer System

1970 - mini computer → Time sharing

History 1980 → Desktop computer

1990 Internet

2000 → Cell phone

Hard real time embedded → nuclear powerplant

	Desktop	Server	Embedded system
1) Price system	\$500 - \$5000	5000 - 500,000	\$10 - \$100
2) Price of processor	\$50 - \$500	\$200 - \$2000	\$0.1 - \$10
3) design issues	Price Graphics	Reliability Availability throughput scalability TSA	Power consumption portability Price

Measuring Report Summarizing the Performance of Computer

→ Response time → is the time b/w start & completion of particular event

→ Throughput → Amount of work done per unit time.

→ Execution time → Latency to complete the event of particular time

Example: X is faster than Y

$n \Rightarrow \frac{\text{Execution time of } Y}{\text{Execution time of } X}$

Scanned by CamScanner

Since Performance = $\frac{1}{\text{execution time}}$

$\therefore$ $n = \frac{\text{performance of } X}{\text{performance of } Y}$

RKTS

Benchmark

1) Real Programs : eg. MS word
2) Kernels → Key piece from real applications
3) Toy Benchmark → eg Quicksort, assignment program (towers of hanoi)
4) Synthetic Benchmark → fake program that behave like the load from real programs, to match profile of real application

1) Desktop Benchmark → to measure performance of desktop
2) Server Benchmark → measure server performance

- Spec Rate → compute
- Spec CFS → for file servers
- Spec web → for webservers

- Spec 92 → organization to measure performance desktop systems
- Spec 95
- Spec 99
- Spec 92
- Spec 2000
- Spec 2006

standard for performance evaluation corporation (Spec)

3) TP Benchmark for TPC
   → Transaction Processing Computing system
   - TPC
   → TPH
   → TPC IOT → Big data

The computer A is 10 times faster than B for program 1.
B is 10 times faster than A for program 2.
A is 20 times faster than C for pgm 1
C is 50 times faster than A for pgm 2
B is 2 times faster than C for pgm 1
Identify the execution time for each of the system for pgm 1 & 2 & also identify the total execution time for system A B & C

p1 p2

A = 10b B = b

A = 20c C = c

B = 2c C = c

	A	B	C
p1	10	1	1/20
p2	1	10	50

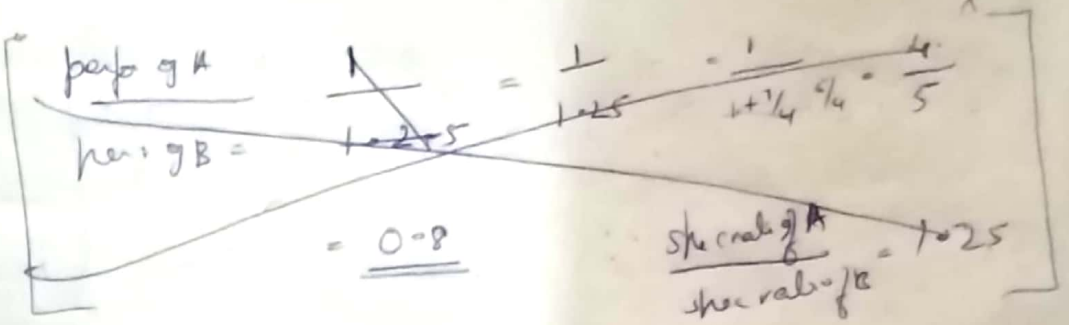
	A	B	C
p1	1	10	20
	100	1	1/50
	20(50) = 1	20(10) = 10	20
p2	1000	100	20
Total time	1001	110	4

Summarizing Performance

eg: If spec ratio is 50 that implies computer under test is 50 times better than reference computer

$$\text{Spec Ratio} = \frac{\text{Execution time of reference computer}}{\text{Execution time on computer under test}}$$

Spec Ratio of Computer A on Benchmark was 1.25 times higher than Computer B. So represent the defn in terms of performance



$$\frac{\text{spec ratio of A}}{\text{spec ratio of B}} = 1.25 = \frac{\text{exec of Ref}}{\text{exec of A}} \Big/ \frac{\text{exec of Ref}}{\text{exec of B}} = \frac{\text{exec of B}}{\text{exec of A}} = \frac{\text{perf of A}}{\text{perf of B}}$$

$$= 1.25$$

$$\boxed{\frac{\text{spec of A}}{\text{spec of B}} = \frac{\text{exec perf A}}{\text{perf B}} = \frac{\text{exec B}}{\text{exec of A}}}$$

$$\text{Geometric Mean} = \sqrt[n]{\prod_{i=1}^{n} \text{sample}_i}$$

$\sqrt[2]{2(4)}$ $\quad 2+4$

$\sqrt{8} = 2\sqrt{2}$

Q2) Show that the Relig G.M is equal to G.M of performance Ratio w.r.t to computer A & B

$\frac{A \cdot B}{\sqrt{AB}} = \frac{B}{A}$

$\Rightarrow$

$$\frac{\text{Geometric mean of A}}{\text{Geometric mean of B}} = \frac{\sqrt[n]{\prod_{i=1}^{n} \text{Spec Ratio } A_i}}{\sqrt[n]{\prod_{i=1}^{n} \text{Spec Ratio } B_i}}$$

$$= \frac{\sqrt[n]{\prod_{i=1}^{n} \frac{\text{execution time of ref}}{\text{exe time of } A_i}}}{\sqrt[n]{\prod_{i=1}^{n} \frac{\text{execution time of ref}}{\text{exe time of } B_i}}}$$

$\frac{\text{Spec Ratio A}}{\text{Spec Ratio B}} = \frac{\text{perf. A}}{\text{perf. B}}$

$$= \sqrt[n]{\prod_{i=1}^{n} \frac{\text{execution time of } B_i}{\text{execution time of } A_i}}$$

$$= \sqrt[n]{\prod_{i=1}^{n} \frac{\text{performance of } A_i}{\text{performance of } B_i}}$$

↑ geometric mean of performance

=

X Quantitative principles of computer design

3 principles

→ 1) Take the advantage of parallelism — { system level scaling; Processor level → pipelining }

2) Principle of locality → [spatial; temporal]

✓ 3) Focus on common case { cache

(optimize the case that are frequently used)

Rule of thumb: 90% code only 10% code

Amdahl's law

→ performance improvement to be gained from using some faster mode execution is limited by fraction of the time the faster mode can be used

$$\text{speedup} = \frac{\text{Performance of entire task using enhancement}}{\text{Performance of entire task without using enhancement}}$$

$$= \frac{\frac{1}{\text{Execution time using enhancement}}}{\frac{1}{\text{Execution time of entire task without using enhancement}}}$$

$$= \frac{\text{Execution time of entire task without using enhancement}}{\text{Execution time of entire task using enhancement}}$$

$$\text{Speedup overall} = \frac{\text{Execution time old}}{\text{Execution time new}}$$

$$= \frac{1}{\left(1 - \text{Fraction enhanced}\right) + \frac{\text{Fraction enhanced}}{\text{speedup enhanced}}}$$

~~Quantitative principles of Design~~

2 factors Amdahl's law gives us 2 ways for some enhancements

→ the fraction of the computation time in the original computer that can be converted.

fraction enhanced is less than 1

→ speed up enhanced greater than 1.

2nd factor

$$\text{Execution Time}_{New} (ET) = ET_{old}\left(\left(1 - Fraction_{enhanced}\right) + \frac{F_{enhanced}}{Speedup_{enhanced}}\right)$$

Overall Speedup Ratio

$$Speedup_{overall} = \frac{ET_{old}}{ET_{new}} = \frac{1}{\left(1 - F_{enhanced}\right) + \frac{F_{enhanced}}{Speedup_{enhanced}}}$$

Q
Amdahl
def
→ 2 factors
→ when cons

→

Q) Suppose that we want to enhance the processor used for web serving. the new processor is 10 time faster. Assuming original processor is busy with computation 40% of the time & is waiting for I/O 60% of time. what is the overall speedup gained by incorporating the enhancement.

⇒ fraction enhanced → amount of time computer is doing computation

$$= \frac{1}{(1 - 10) + \frac{10}{}}$$

$$1 - 40 + 0.4$$

$$1 - 0.4 + \frac{0.4}{10}$$

$$= \frac{1}{\left(1 - \frac{40}{100}\right) + \frac{\frac{40}{100}}{10}}$$

$$= \frac{1}{(1 - 0.4) + \frac{0.4}{10}}$$

$$= \frac{1}{0.6 + 0.04}$$

$$Speed\ up_{overall} = \frac{1}{0.64} = \frac{100}{64} \frac{25}{16} = \frac{25}{16} = 1.56$$

Q) Example 2

A common transformation required in graphics processors is square root. Implementations of floating point (FP) square root vary significantly in performance, especially among processors designed for graphics. Suppose FP square root (FPSQR) is responsible for 20% of the execution time of a critical graphics benchmark. One proposal is to enhance the FPSQR hardware and speed up this operation by a factor of 10.

$$= \frac{1}{(1-0.2) + \frac{0.2}{10}}$$

$$= \frac{1}{0.8 + 0.02} = \frac{1}{0.82} = 1.22$$

20% / 100

20 / 100

Speed. $\frac{100}{20} = 5$

2nd Q factor = ½ of execution time = 0.5 speed up = 1.6

→

$$= \frac{1}{(1-0.5) + \frac{0.5}{1.6}}$$

$$= \frac{1}{0.5 + 0.3125} = \frac{1}{0.8125} = 1.23$$

better

Hence improving the performance of the FP operation overall is slightly better because of higher freq.

Q) Derive the CPU performance equation

→ we do it interms of clock cycle

CPU time = CPU clock cycles for program * clock cycle time ——— (1)

CPU time = CPU clock cycles for a program × $\frac{1}{\text{clock rate}}$

Number of instructions executed → the instruction path length or instruction count.

clock cycles per instruction (CPI) = computed as

$$CPI = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}} \quad -(2)$$

from (1) & (2)

$$CPI * \text{Instruction count} = \frac{\text{CPU time}}{\text{CPU clock cycle time}}$$

$$\text{CPU time} = CPI * IC * \text{CPU clock cycle time}$$

Alternatively

$$\text{CPU time} = \frac{\text{clock cycle}}{\text{Instruction}} \times \frac{\text{Instruction}}{\text{program}} \times \frac{\text{second}}{\text{clock cycle}}$$

$$= \frac{\text{second}}{\text{program}} = \text{CPU time}$$

$$\text{CPU clock cycles} = \sum_{i=1}^{n} IC_i \times CPI_i \quad -(3)$$

where IC_i represent number of time instruction is executed in a program

CPI_i represents the average number of clock cycles for Instruction for Instruction

(3) in (1)

$$\text{CPU time} = \left(\sum_{i=1}^{n} IC_i \times CPI_i \right) \times \text{clock cycle time}$$

the overall CPI can be computed using eqn.

(3) + (2)

overall CPI

$$CPI = \frac{\sum_{i=1}^{n} IC_i \times CPI_i}{\text{Instruction count}}$$

← no of times the instruction is fetched

$$CPI = \sum_{i=1}^{n} \frac{IC_i}{\text{Instruction count}} \times CPI_i$$

Example 3

→ suppose we have made the following measurement

freq of FP operation is 25%

Avg CPI of FP operation is 4.

Avg CPI of our instn is 1.33

Freq of FPSQR = 2%

→ CPI of FPSQR is 20

→ Assume next 2 design alternatives are decrease CPI of FPSQR to 2 or decrease the avg CPI of all FP ops to 2.5. Compare the 2 design alternatives using the performance eqn.

$$CPI = \frac{25 \times 4}{100} + \frac{2 \times 20}{100}$$

$$= 1 + 0.4$$

$$= 1.4$$

$$= \frac{25(0.5)}{100} + \frac{2(40)}{100}$$

$$\Rightarrow CPI_{\text{overall}} = \sum CPI_i \times \frac{IC_i}{\text{Instruction count}}$$

$$= FP \text{ operation}$$

$$= 4 \times 25\% + 1.33 \times 75\%$$

$$= \frac{4 \times 25}{100} + \frac{1.33 \times 75}{100} \%$$

$$= 1 + 0.99$$

$$= 2.0$$

25/100

Compute the CPI for the enhanced FPSQR by subtracting the cycles saved from the original CPI

$$CPI_{\text{with new FPSQR}} = CPI_{\text{original}} - 2\% \left(CPI_{\text{old FPSQR}} - CPI_{\text{new}}\right)$$

$$= 2 - 0.02(20 - 2)$$

$$= 2 - 0.02(18)$$

$$= 2 - 0.36$$

$$= 1.64$$

$$CPI_{\text{new FP}} = CPI_{\text{original}} - 25\% \left(CPI_{\text{old FP}} - CPI_{\text{new FP}}\right)$$

$$= 2 - 25\%(4 - 2.5)$$

$$= 2 - \frac{25}{100}(1.5)$$

$$= 2 - \frac{1}{4}(1.5)$$

$$= 2 - 0.375$$

$$= 1.625$$

Note: If CPI is low the machine is better.

Performance magnitude better with FP enhancement since CPI is slightly lower

Speedup with new FP $= \frac{\text{CPU time original}}{\text{CPU time new FP}}$

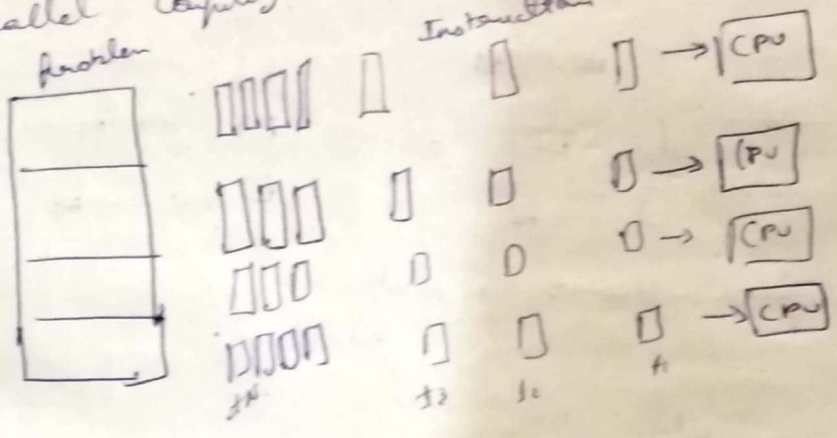
$$= 2/1.625 = 1.23$$

Introduction to parallel processing

→ simultaneous executing multiple instruction of a computational task

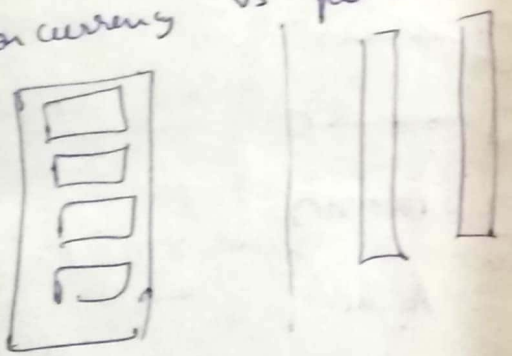
thread level → Multiprocessor → concurrent execution
Instruction level → Pipelined

Parallel computing



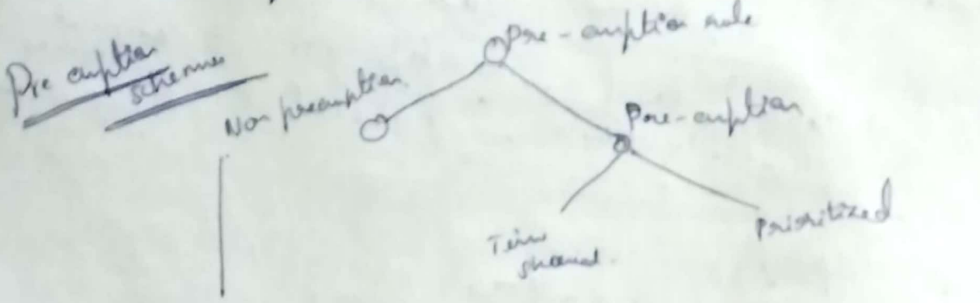
data level parallelism
procedure level parallelism

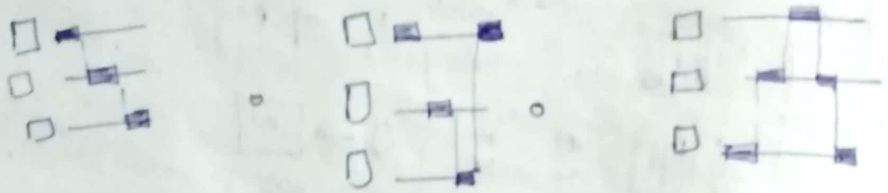
Concurrency vs parallelism



~~to~~ multiple client & 1 server
→ to satisfy all client we have to apply Pre-emption like

Pre emption scheme





In real time pre-emption is a must

Parallel execution

. N client N server model
synchronous or Asynchronous

Start of same time

Types and levels of parallelism

available → six types → whole working loop we need to do it
utilized → occurring during execution → handle at execution level

Types of available parallelism

→ Data parallelism → for (i=0 to 99)
→ arise from data structure a(i) = b(i) + c(i)

→ Functional parallelism
→ Independent tasks apply different operations to different data

a ← 2
b ← 3
m = (a+b)/2
s = (a²+b²)/2
v = m - s

Available level
user (pgm) level
procedure level
loop level
Instruction level

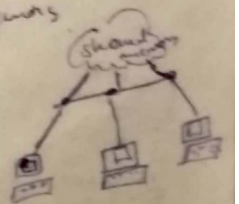
user level
multiprogramming
multitasking
multithreading
process level
thread level
Instruction level
OS level
architecture level

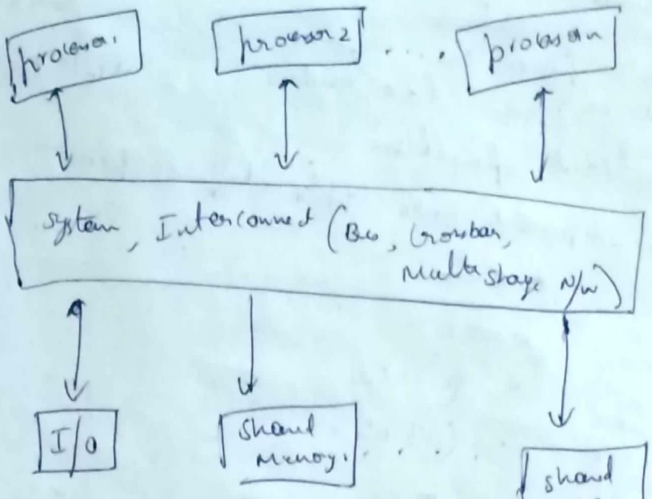
multithreading ←
multitasking ← process level
multiprogramming ← both at pgm & process level (program running parallel)

Shared Memory Multiprocessors

→ uniform memory access (UMA) model
→ Non uniform memory access (NUMA)
→ Cache-only memory architecture (COMA)

UMA → In a N/w multiple computer having a shared memory
→ time sharing system, tightly coupled system





tightly coupled → mean more dependency

share data → dependency b/w processors.
→ to access same variable

→ tightly coupled system

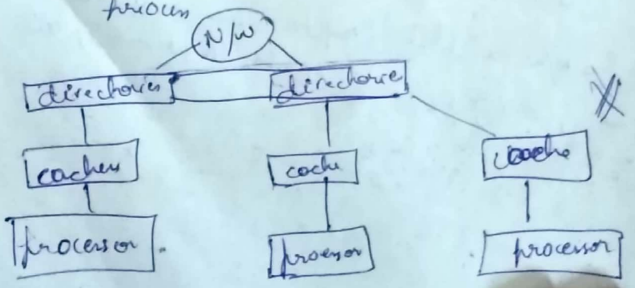
→ Symmetric multiprocessor → Processor have equal access to all peripheral

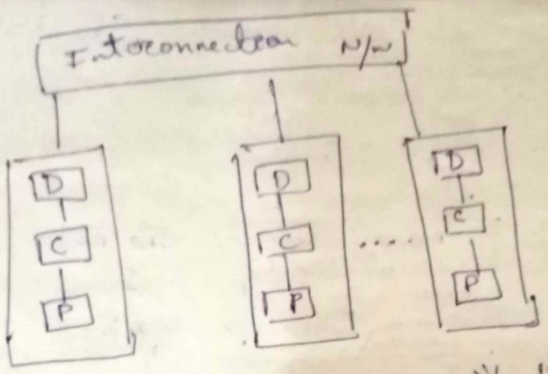
Asymmetric multiprocessor → Only 1 or subset processor can have I/O or executive capability
eg: master slave

Remaining processor (slaves) don't have I/O capability
(Attached processor)

NUMA → shared memory is divided into small memories or divided processor divided into clusters.

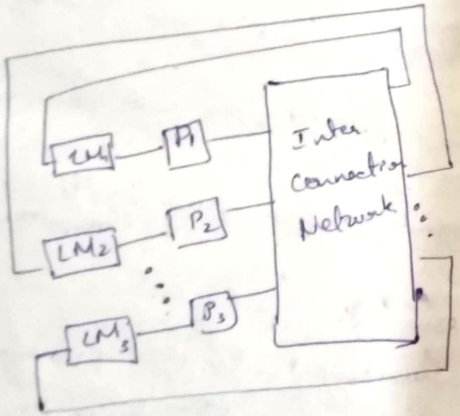
COMA



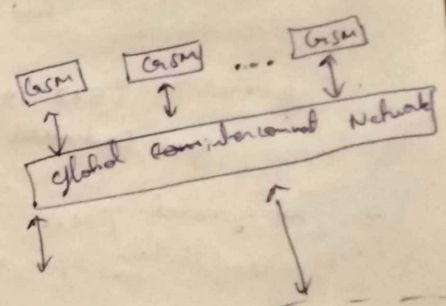


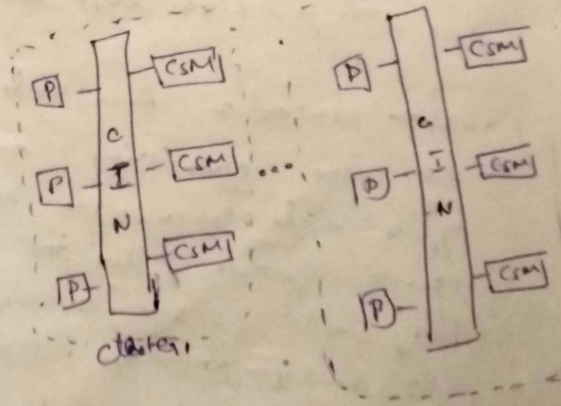
D → directory
C → cache
P → Processor

Non uniform memory access (NUMA)



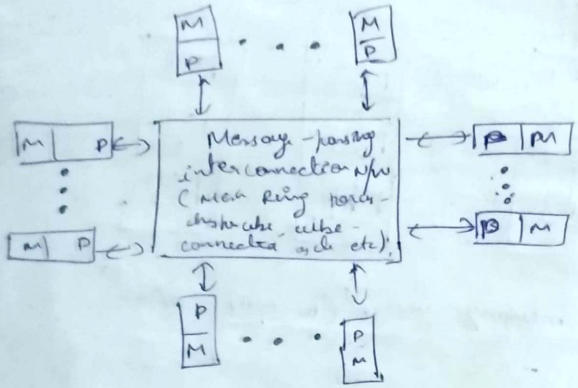
n way





CSM → cluster shared memory
GSM → global / not-shared memory
CIN → cluster Interconnect N/W
P → Processor

distributed Memory Multi computer (Stand alone)



→ 1st generation (1983-1987) were based on processor technology, "s/w controlled msg switching"

→ 2nd generation (1988-1992) was implemented with mesh connected architecture, h/w message routing.

→ 3rd gen (1993-1997) was based on fine grain multi computers

Problem based on Amdahl law

→ we are considering an enhancement to the processor of a web server. The new CPU is 20 times faster on search queries than the old processor. The old processor is busy with search queries 70% of the time, what is the speedup gained by integrating enhanced CPU

$$\text{Speedup} = \frac{1}{(1-0.7) + \frac{0.7}{20}}$$

$$= \frac{1}{(0.3) + 0.035} = \frac{1}{0.335} = 2.985$$

UNIT-
Pipelin
where
J is
Each