Arhitekture paralelnih računara

Izborni seminar - paralelno programiranje

Institut za matematiku i informatiku

Institut za matematiku i informatiku
Prirodno-matematički fakultet, Kragujevac

Novembar 2010. god.

O čemu će biti reči?

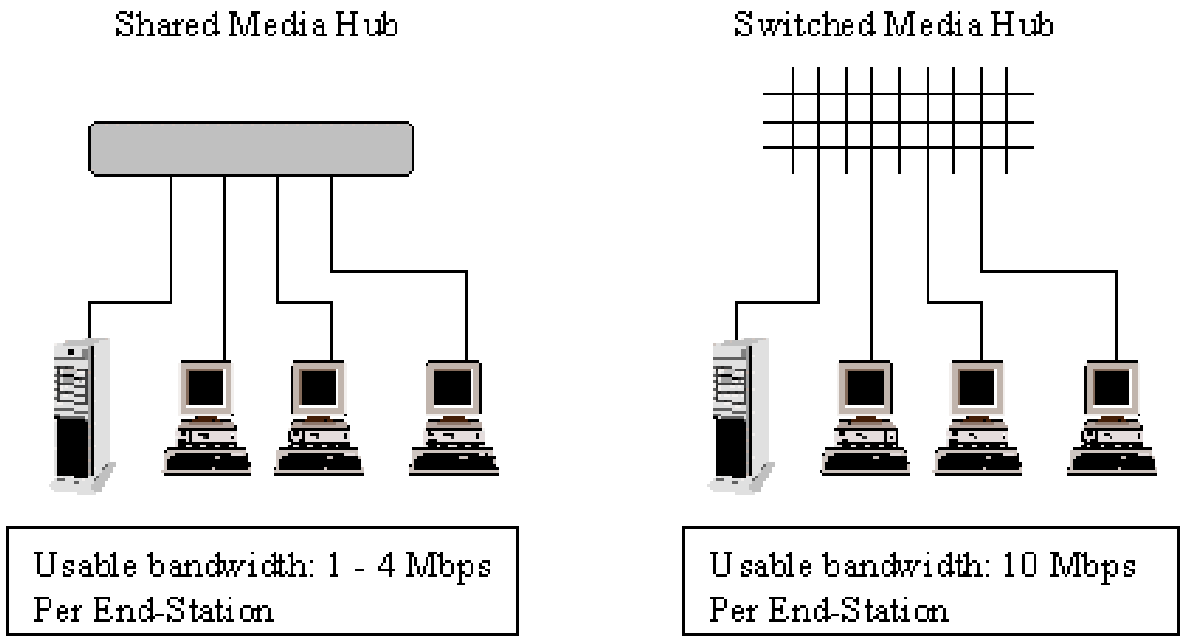
1. Arhitektura mreže
2. Procesorki nizovi
3. Multiprocesori

Uvod

- Od 60-ih do sredine 90-ih proslog veka ispitane su mnoge arhitekture paralelnih sistema
- Osnovna debata: najviše nekoliko desetina vrlo moćnih procesora ili hiljade "običnih" procesora
- Debata je danas rešena: sistemi napravljeni od specijalnih procesora su retkost

Deljeni ili switch-ovani medijum

- Deljeni medijum (recimo mreža preko *hub*-a) dozvoljava transfer samo jedne poruke istovremeno
- *Switch*-ovani medijum dozvoljava da se point-to-point komunikacije odigravaju konkurentno



Topologije switch-ovanih mreža

Topologije

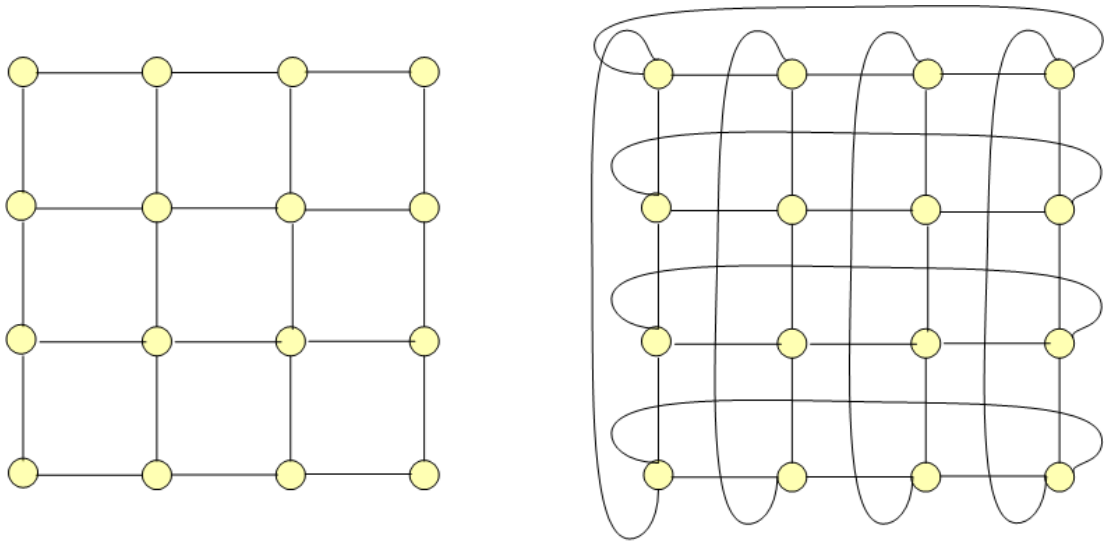
- Direktna topologija - odnos broja *switch* čvorova prema broju procesora je $1:1$
- Indirektna topologija - odnos broja *switch* čvorova prema broju procesora je $> 1:1$

Kriterijumi performansi mreže

1. Dijametar - najveća udaljenost između 2 čvora
2. Bisekciona širina - minimalni broj stranica koje treba prekinuti da bi se mreža podelila na 2 dela
3. Ivica po *switch* čvoru - poželjno konstanta nezavisna od veličine mreže, zbog skaliranja
4. Konstantna dužina ivice - u 3D prostoru, zbog skaliranja nezavisna od veličine mreže

2D-Mesh mreža

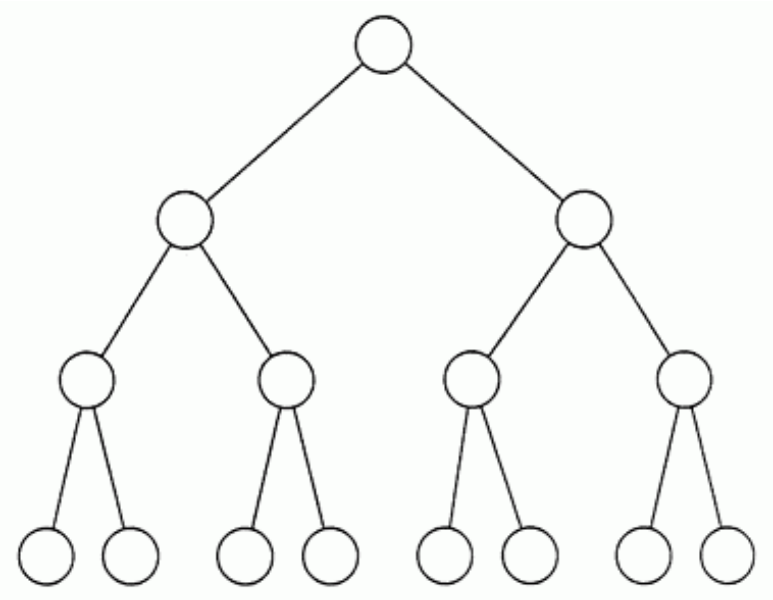
- Direktna topologija
- *Wraparound* varijanta ima konstantan broj ivica po čvoru
- Minimalni dijametar i maksimalna bisekciona širina postižu se topologijom kvadrata
- Tada su oba ova parametra jednaka $\theta(\sqrt{n})$ za n čvorova



(a) (b)

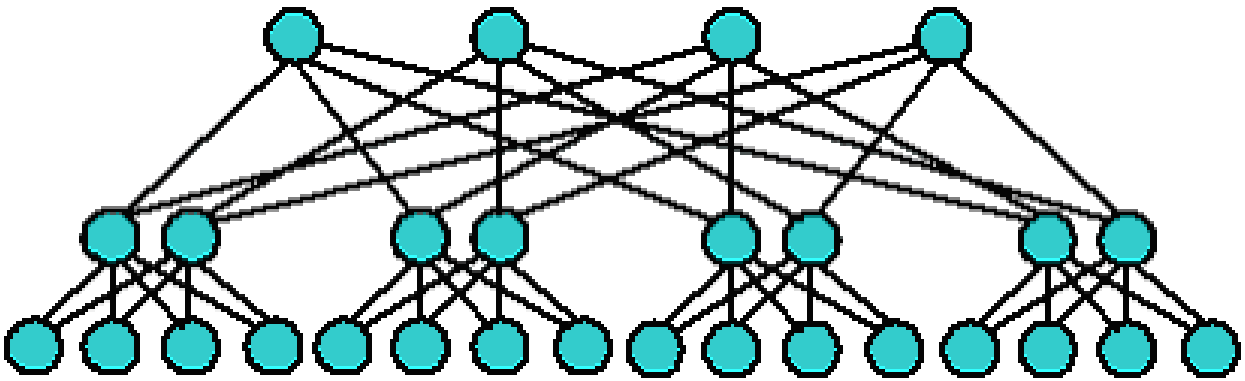
Topologija binarnog stabla

- Indirektna topologija
- Ukupno $n = 2^d$ procesora sa $2n - 1$ svičeva
- Relativno mali dijametar $2log_2 n$
- Međutim, bisekciona širina je vrlo niska, 1
- Prilično nezgodno za postavljanje u prostoru jer nije 3D



Topologija hiperstabla

- Hiperstablo stepena k i dubine d se najlakše može objasniti ako se posmatra iz dva različita ugla
- Sa prednje strane izgleda kao drvo k-tog stepena dubine d
- Sa strane izgleda kao obrnuto binarno stablo dubine d
- 4-struko hiperstablo dubine d može da udomi 4^d procesora sa ukupno $2^d(2^{d+1} - 1)$ svičeva
- Dijametar je $2d$, a bisekciona širina 2^{d+1}
- Broj ivica po svič čvoru nikad nije veći od 6



fat tree architecture

Butterfly topologija

- Indirektna topologija
- $n = 2^d$ procesora je zakačeno sa $n(log_2 n + 1)$ svič čvorova
- Svič čvorovi su podeljeni u $log_2 n + 1$ vrsta, tzv. **rankova**
- (i, j) - j-ti čvor i-tog ranka
- Najduža ivica se povećava kako raste broj čvorova
- Pretpostavimo da su čvorovi ranga 0 i ranga $log_2 n$ isti čvorovi.

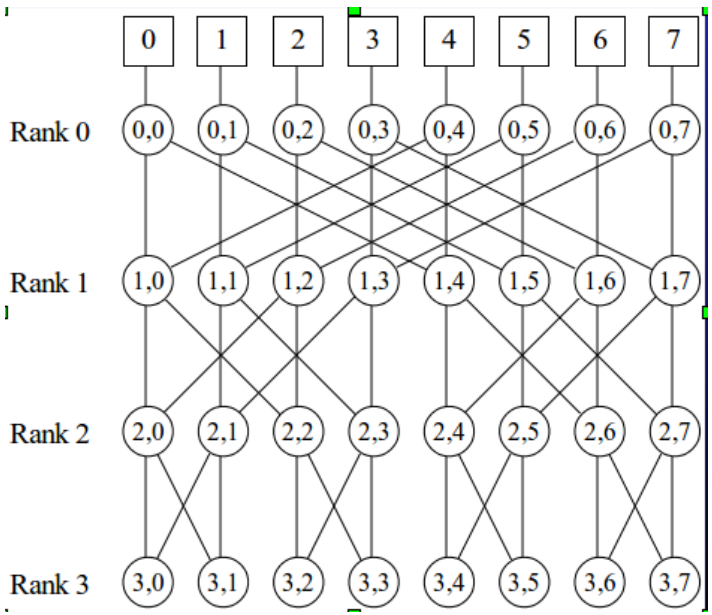
Povezanost čvorova

Čvor (i,j) povezan je sa 2 čvora ranga $i - 1$, i to $(i - 1, j)$ i $(i - 1, m)$, gde je m dobijeno inverzijom i-tog bita po važnosti u binarnoj reprezentaciji j.
Na primer, (2,3) je povezan sa (1,3) i (1,1).

Butterfly topologija

Algoritam:

1. Svaki svič čvor uzima vodeći bit iz poruke
2. Ako je taj bit 0, ide dole na levi link, a ako je 1 ide na desni link
3. Primer, procesor 2 šalje poruku procesoru 5 kao “101poruka”

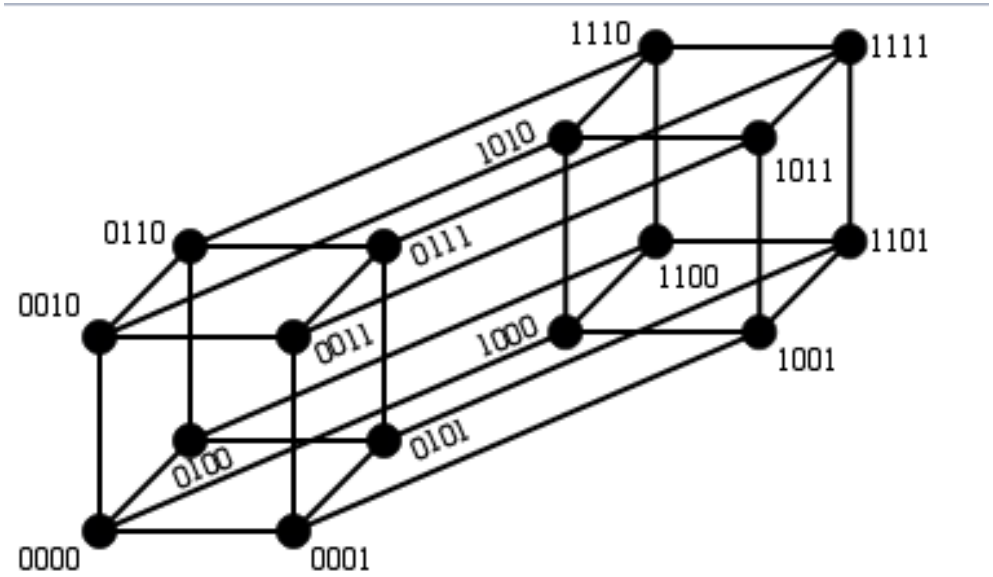


Hiperkocka

- Hiperkocka je *butterfly* topologija u kojoj je svaka kolona svič čvorova kolabirana u jedan čvor
- Binarna n-kocka sastoji se od $n = 2^d$ procesora i isto toliko svičeva (direktna topologija)
- Oni su numerisani kao $0, 1, \cdots 2^d - 1$.
- Susedni su ako im se binarna reprezentacija razlikuje za tačno 1 bit
- Dijametar je $log_2 n$, a bisekciona širina $n/2$
- Ovo je vrlo dobro, ali na račun broja ivica po čvoru ($log_2 n$)

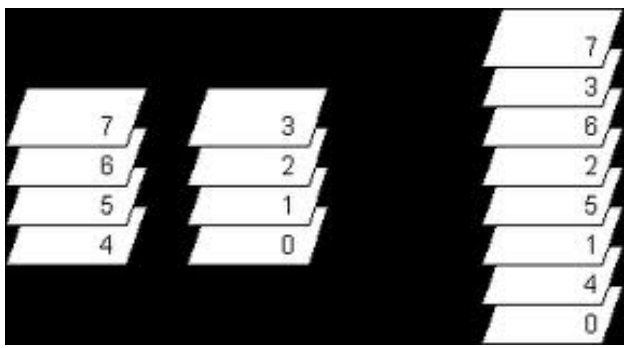
Hiperkocka

- Na primer čvor 0101 sa 1101, 0001, 0111 i 0100.
- Broj bitova u kojima se čvorovi razlikuju određuje njihovo rastojanje
- Primer put rastojanja 2: $0101 \rightarrow 0001 \rightarrow 0011$, a može i $0101 \rightarrow 0111 \rightarrow 0011$



Shuffle exchange topologija

- Ideja je perfektno izmešani špil karata
- Pozicija u izmešanom špilu se dobija rotacijom binarnog zapisa ulevo, npr. sa pozicije 5 (101) na poziciju 3 (011)
- *Shuffle-exchange* topologija je direktna topologija sa $n = 2^d$ procesora $(0, \ldots n-1)$
- Exchange veze - linkuju procesore koji se razlikuju u jednom bitu
- Shuffle veze - linkuju procesore koji se dobijaju opisanom *bitwise* rotacijom

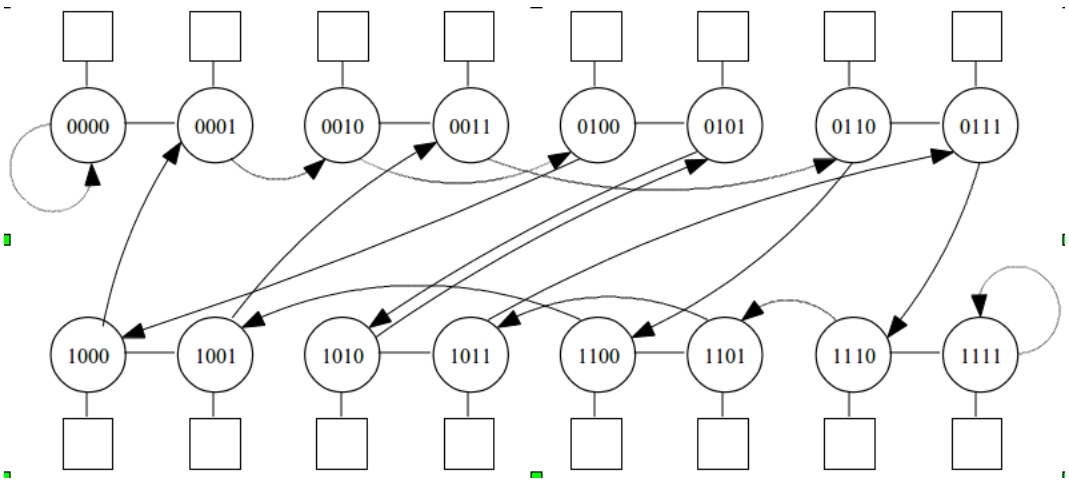


Shuffle exchange topologija

- Konstantan broj ivica po čvoru - 2
- Dužina najduže ivice raste sa veličinom mreže
- Dijametar je $2log_2 n - 1$, a bisekciona širina $\approx n/log_2 n$

Najgori slučaj rutiranje $0 \rightarrow n-1$

$0000 \xrightarrow{E} 0001 \xrightarrow{S} 0010 \xrightarrow{E} 0011 \xrightarrow{S} 0110 \xrightarrow{E} 0111 \xrightarrow{S} 1110 \xrightarrow{E} 1111$



Zaključak

1. Sve arhitekture imaju logaritamske dijametre osim *2D Mesh*
2. Hiperstablo, *butterfly* i hiperkocka imaju bisekcionu širinu $n/2$
3. Sve imaju konstantan broj ivica po čvoru osim hiperkocke
4. Samo *2D Mesh* održava dužine ivica konstantnim kako se veličina mreže povećava

Vektorski procesori

Definicija

Vektorski računar je računar koji pored operacija sa skalarima, implementira i vektorske operacije.

U opštem slučaju, dva su načina za njegovu implementaciju:

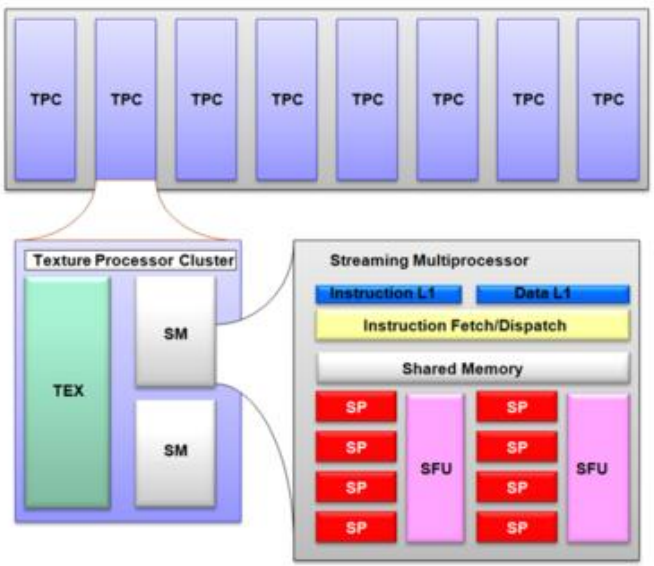
1. *Pipelined* vektorski procesor - poseduju pipeline aritmetičku jedinicu. Rane arhitekture...
2. Procesorski niz - identični sinhronizovani procesorski elementi koji jednu istu operaciju izvode nad različitim podacima

Arhitektura vektorskih računara

Zašto vektorski procesori?

1. Istorijski, visoka cena CPU kontrolne jedinice
2. Naučne aplikacije imaju inherentni paralelizm podataka

- Procesorski niz se sastoji iz više parova procesor-memorija
- Postoji *front-end* procesor i *back-end* procesori



Arhitektura vektorskih računara
Primeri

Primer (1)

Ako sistem poseduje 1024 procesorske jedinice, a vektori A i B po ≤ 1024 elementa, sabiranje $A + B$ može se obaviti u jednoj instrukciji.

Primer (2)

Ako sistem poseduje 1024 procesorske jedinice, vektor od 10000 članova može da se sačuva tako što 784 procesora dobije po 10 elemenata, a 240 procesora po 9 elemenata.

Primer (3)

Niz sadrži 1024 procesora, a svaki od njih može da sabere 2 broja u $1\mu s$. Za sabiranje dva vektora od 1024 člana:

$$Performance = \frac{1024 ops}{1\mu s} = 1024 \times 10^9 ops/s$$

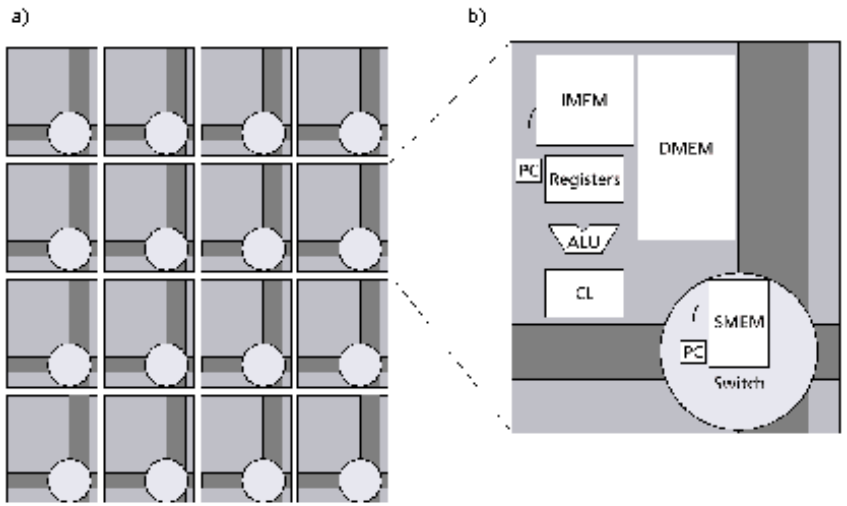
Primer (4)

Niz sadrži 512 procesora, a svaki od njih može da sabere 2 broja u $1\mu s$. Za sabiranje 2 vektora od 600 članova:

$$Performance = \frac{600 ops}{2\mu s} = 3 \times 10^8 ops/s$$

Arhitektura vektorskih računara
Komunikaciona mreža

- Tipična vektorska operacija je mnogo komplikovanija od 1-1 sabiranja. Recimo kod metode konačnih razlika: $a_i \leftarrow (a_{i-1} + a_{i+1})/2$, dakle postoje interprocesorske komunikacije
- Za ovu arhitekturu, komunikacione linije moraju da podržavaju konkurentni transfer poruka (*2D Mesh*)

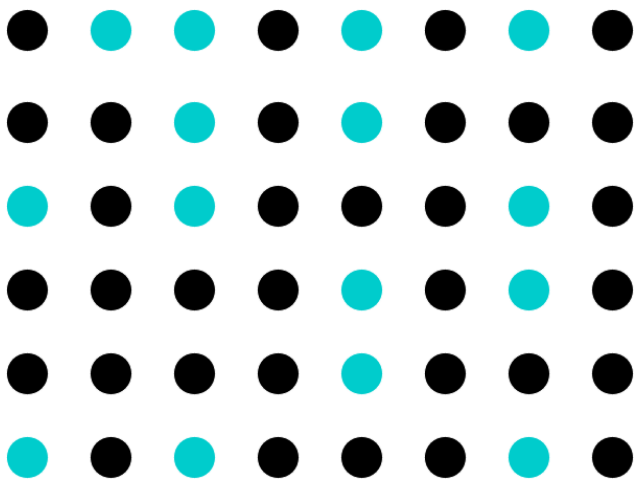


Arhitektura vektorskih računara

Grananje

Primer

Vektor A ima 10 elemenata koji su distribuirani na 10 procesora. Ako je element vektora različit od nule, menja se u 1, a ako je jednak 0, menja se u -1.
Zahteva se uključivanje/isključivanje pojedinih procesora u pojedinim granama, što dovodi do pada performansi.



Arhitektura vektorskih računara
Nedostaci

1. Nemaju svi problemi paralelnost podataka
2. Grananja usporavaju kôd
3. Ne adaptiraju se dobro na sisteme sa više korisnika
4. Baziraju se na *custom* VLSI procesorima
5. Cena kontrolnih jedinica ja dosta pala

Multiprocesori

- Sistem sa više CPU-ova i deljenom memorijom
- Ista adresa na više CPU-ova referiše na isti podatak
- Izbegavaju se 3 problema proc. nizova:
  1. Može se izgraditi od "običnih" CPU-ova
  2. Prirodno podržava više korisnika
  3. Efikasnost dobra i sa kondicionim kôdom
- Dele se na dve velike podgrupe: centralizovane multiprocesore i distribuirane multiprocesore

Centralizovani multiprocesor

- Jasna ekstenzija uniprocesora
- CPU-ovi se dodaju na magistralu
- Svi dele zajednički RAM
- Performanse pristupa memoriji su iste za sve CPU-ove (UMA - *Uniform Memory Access*)
- SMP - Symmetrical MultiProcessor

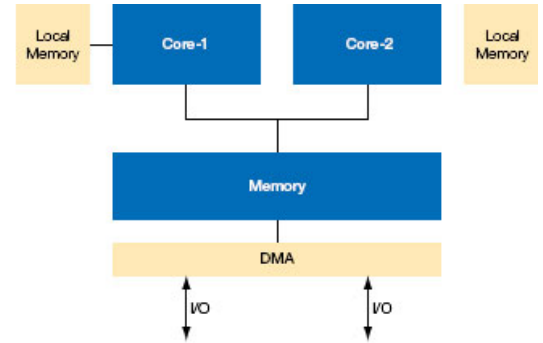


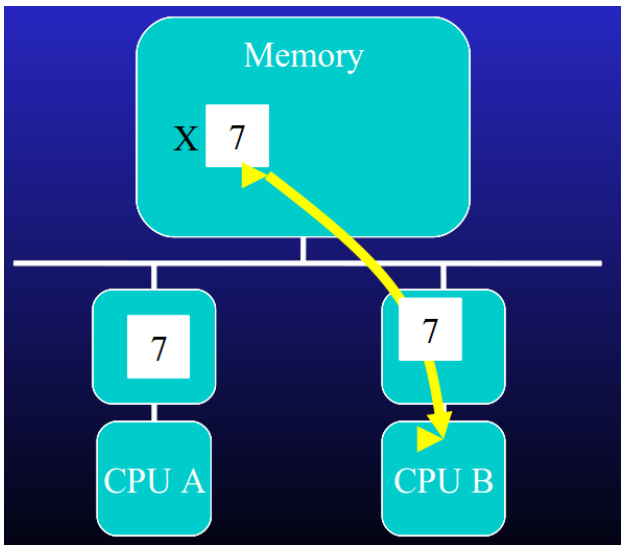
Figure 1 In a simplified SMP architecture, multiple, identical processors have access to common memory, and (optionally) also to private local memory that generally can be accessed with lower latency.

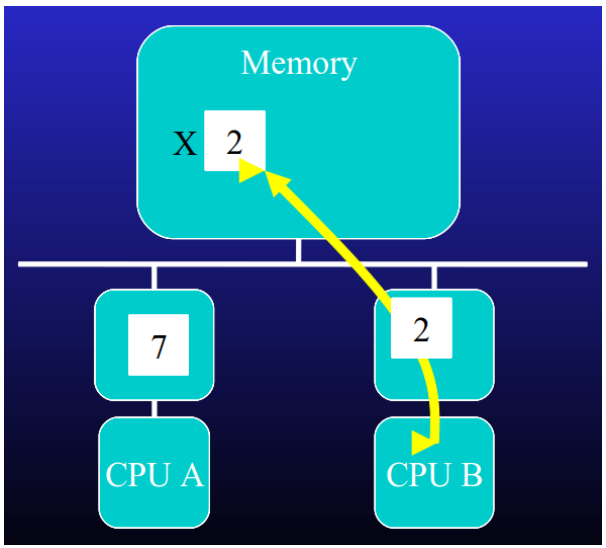
Privatni i deljeni podaci

- Privatni podaci: objekti koji se koriste samo od strane jednog procesora
- Deljeni podaci: objekti koji koristi više procesora
- U multiprocesoru, procesi komuniciraju koristeći deljene podatke!

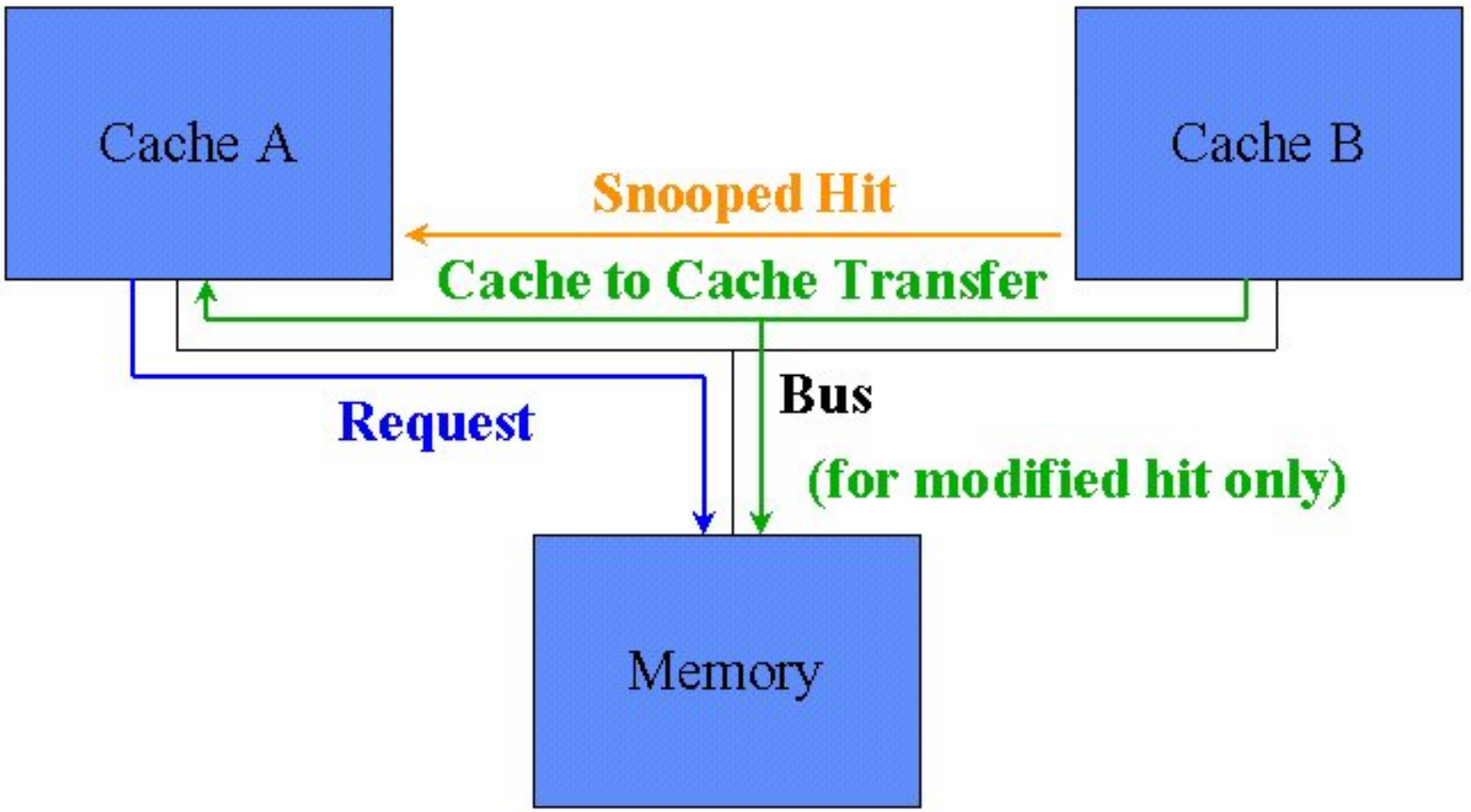
Problemi vezani za deljene podatke

1. Keš koherencija
   - Replikacija podataka u višestrukim keš memorijama redukuje performanse
   - Kako se osigurati da svi različiti procesori imaju isti podatak na istoj memorijskoj adresi?
2. Sinhronizacija
   - *Mutual exclusion - mutex*
   - Barijera





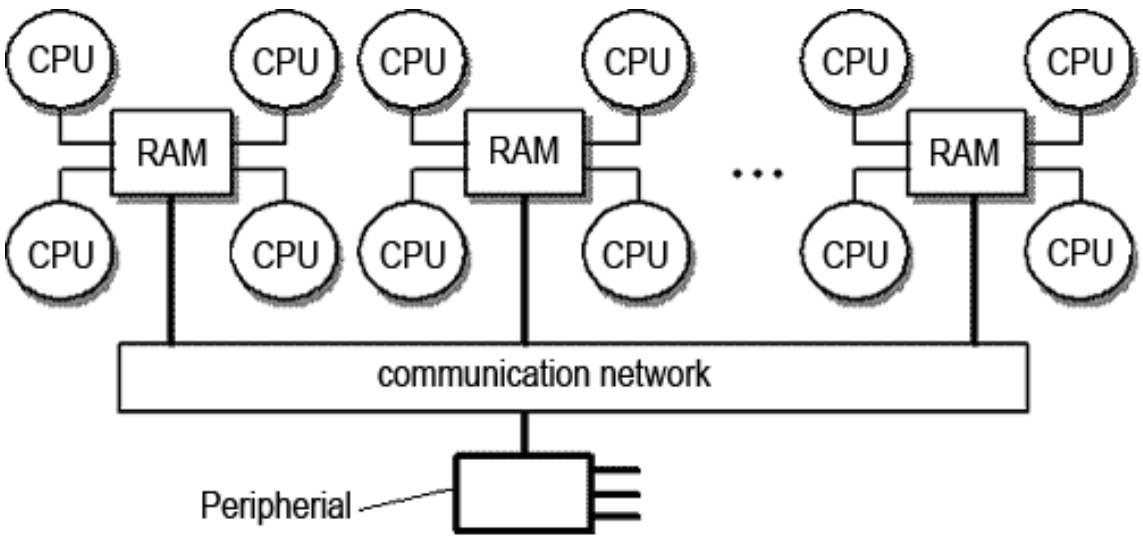
Snooping protocol - write invalidate protocol



Distribuirani multiprocesor

Karakteristike

- Propusna moć magistrale ograničava SMP na par desetina procesora
- Alternativa je distribuirati primarnu memoriju po procesorima, ali da svi imaju zajednički adresni prostor
- NUMA - Non-uniform memory access



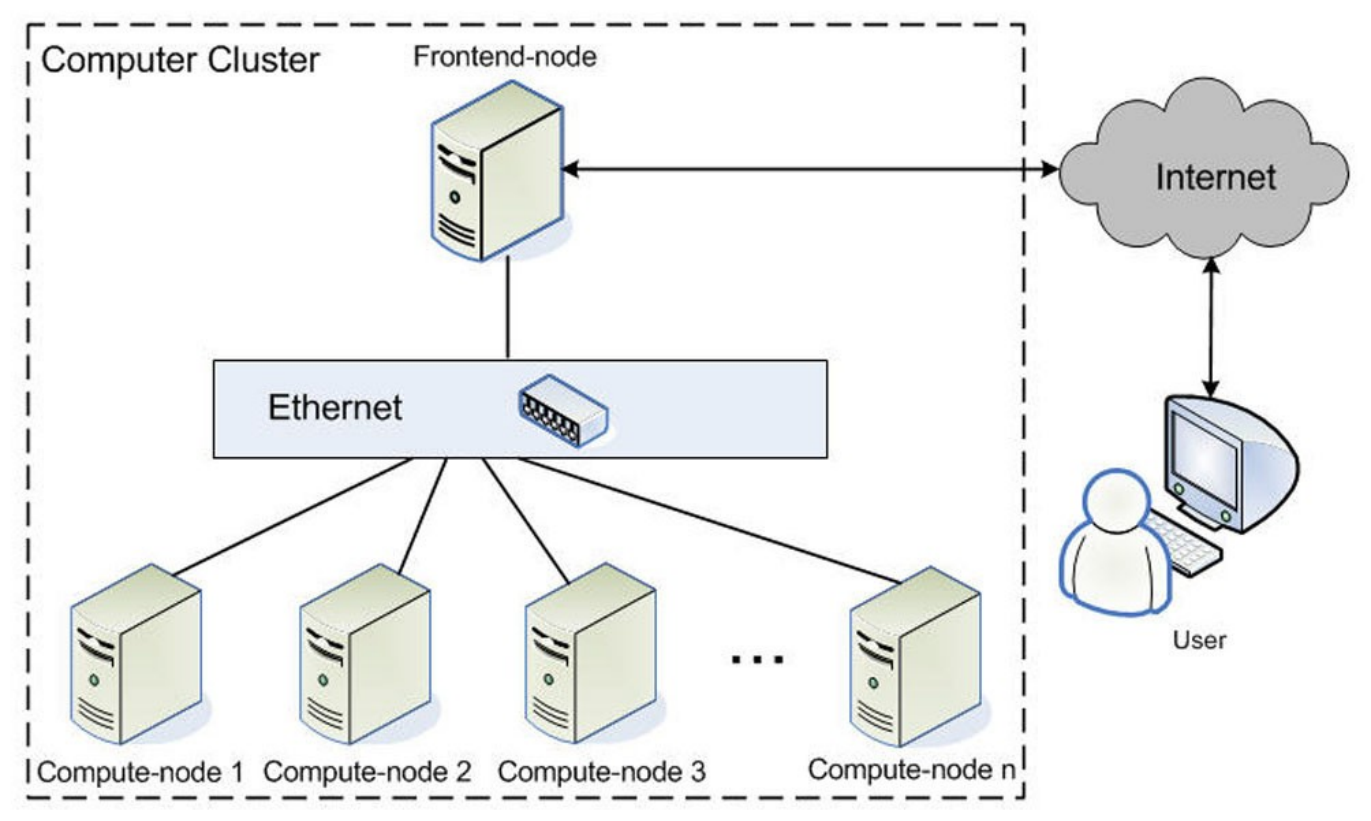
Outline

- Multicomputers
- Flynn's taxonomy

Multicomputer

- Distributed memory multiple-CPU computer
- Same address on different processors refers to different physical memory locations
- Processors interact through message passing
- Commercial multicomputers iPSC I, II, Intel Paragon, Ncube I, II
- Commodity clusters – e.g., Cheetah

Asymmetrical Multicomputer



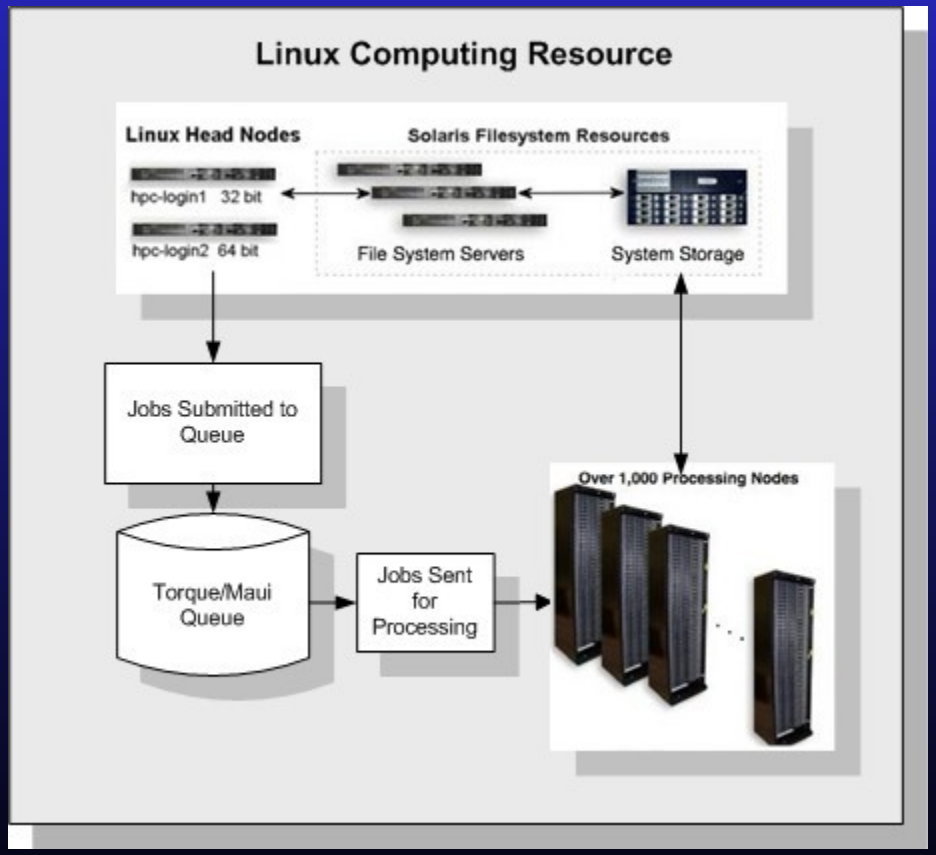
Asymmetrical MC Advantages

- Back-end processors dedicated to parallel computations $\Rightarrow$ Easier to understand, model, tune performance
- Only a simple back-end operating system needed $\Rightarrow$ Easy for a vendor to create

Asymmetrical MC Disadvantages

- Front-end computer is a single point of failure
- Single front-end computer limits scalability of system
- Primitive operating system in back-end processors makes debugging difficult
- Every application requires development of both front-end and back-end program

Symmetrical Multicomputer



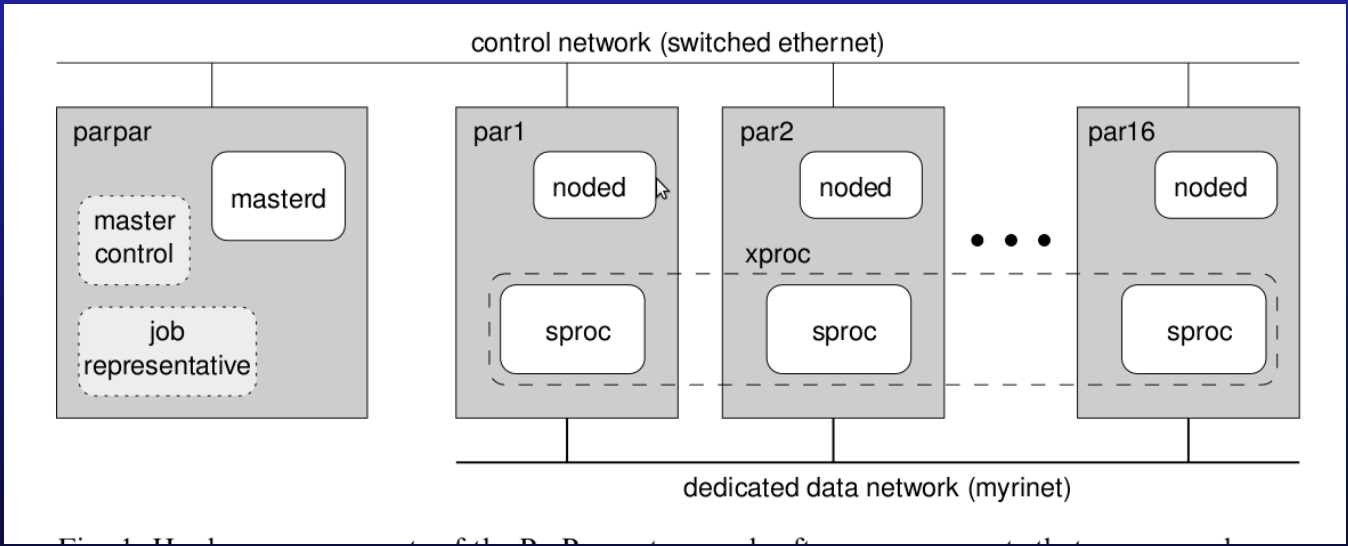
Symmetrical MC Advantages

- Alleviate performance bottleneck caused by single front-end computer
- Better support for debugging
- Every processor executes same program

Symmetrical MC Disadvantages

- More difficult to maintain illusion of single "parallel computer"
- No simple way to balance program development workload among processors
- More difficult to achieve high performance when multiple processes on each processor

ParPar Cluster, A Mixed Model



Commodity Cluster

- Co-located computers
- Dedicated to running parallel jobs
- No keyboards or displays
- Identical operating system
- Identical local disk images
- Administered as an entity

Network of Workstations

- Dispersed computers
- First priority: person at keyboard
- Parallel jobs run in background
- Different operating systems
- Different local images
- Checkpointing and restarting important

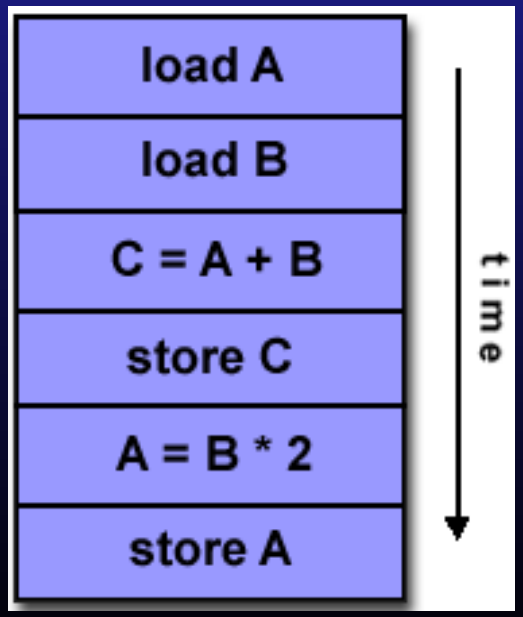
Flynn's Taxonomy

- Instruction stream
- Data stream
- Single vs. multiple
- Four combinations

S I S D Single Instruction, Single Data	S I M D Single Instruction, Multiple Data
M I S D Multiple Instruction, Single Data	M I M D Multiple Instruction, Multiple Data

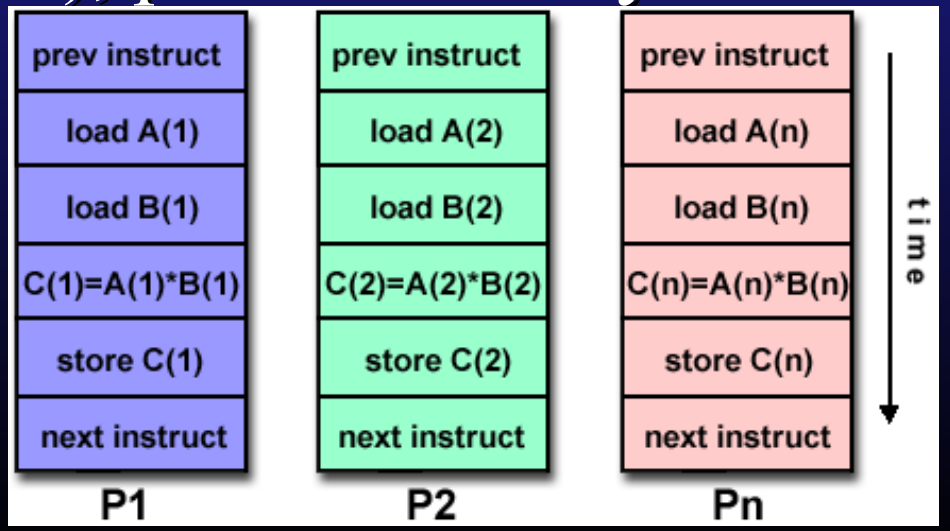
SISD

- Single Instruction, Single Data
- Single-CPU systems
- Note: co-processors don't count
  - Functional
  - I/O
- Example: PCs



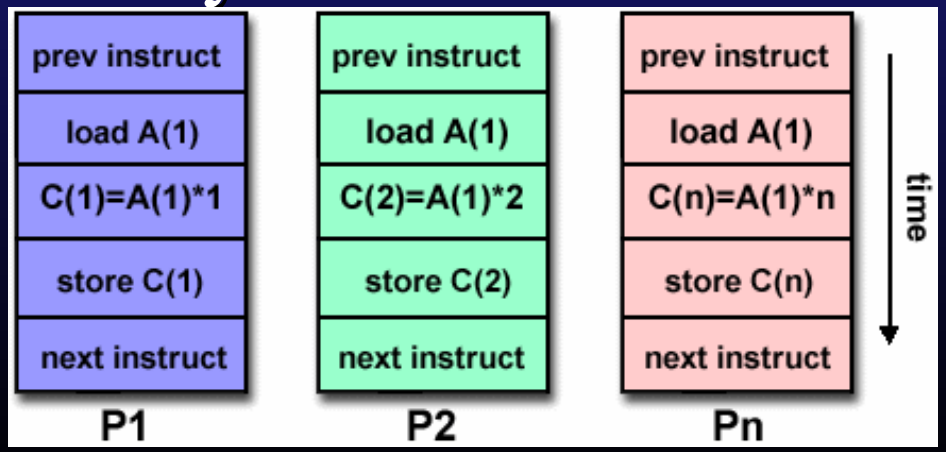
SIMD

- Single Instruction, Multiple Data
- Two architectures fit this category
  - Pipelined vector processor (e.g., Cray-1), processor array



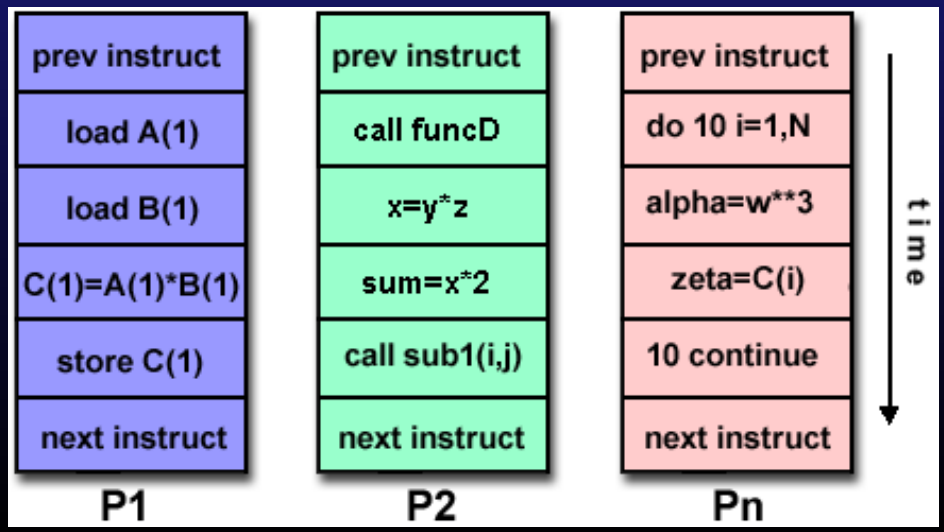
MISD

- Multiple Instruction, Single Data
- Example: systolic array??



MIMD

- Multiple Instruction, Multiple Data
- Multiple-CPU computers
  - Multiprocessors
  - Multicomputers



Summary

- Commercial parallel computers appeared in 1980s
- Multiple-CPU computers now dominate
- Small-scale: Centralized multiprocessors
- Large-scale: Distributed memory architectures (multiprocessors or multicomputers)