Porazdeljeni sistemi

2. Paralelne arhitekture I

Predavatelj: izr. prof. Uroš Lotrič
Asistent: Davor Sluga

Uvod

- 1960 – 1990, vrh 1980
- Preizkušenih mnogo različnih arhitektur
  - Razvoj posebnih procesorjev za paralelne sisteme
  - Dilema: veliko enostavnih procesorjev ali manj bolj kompleksnih
- Danes
  - Zgornjih dilem ni več
  - Najbolj se splača uporabljati masovno proizvedene procesorje
  - Razvoj specializiranih procesorjev je predrag in prepočasen

Pregled paralelnih arhitektur

- Povezovanje procesorjev
  - Polja procesorjev
  - Večprocesorski sistemi
  - Večračunalniški sistemi
- Flynnovo označevanje
  - SISD
  - SIMD (vektorski, GPU)
  - MIMD (shared memory: UMA, NUMA, distributed memory)

Delitev paralelnega računanja

- Flynnovo označevanje
  - Najbolj znana shema označevanja paralelnih računalnikov
  - Večina modernih paralelnih računalnikov pade pod rubriko MIMD kar zmanjšuje uporabnost klasifikacije

		Podatkovni tok	
		Enojni	Večkratni
Ukazni tok	Enojni	**SISD** Običajni procesorji	**SIMD** Procesorska polja Cevovodni vektorski procesorji
	Večkratni	**MISD** Sistolična polja	**MIMD** Večprocesorski sistemi Večračunalniški sistemi

  - Sistolično polje: cevovod več neodvisnih funkcijskih enot, ki delujejo na istem podatkovnem toku

Delitev paralelnega računanja

- Programerski vidiki
  - Podatkovni paralelizem (SIMD)
  - Funkcijski paralelizem (MIMD)
    - Različni programi, različni podatki
  - Isti program, različni podatki (SPMD)
  - Podatkovni tok: cevovodni paralelizem

  - Komentarji
    - SIMD in SPMD:
      - SPMD bolj splošen, na običajnih računalnikih
      - SIMD za vektorske računalnike
    - SPMD in MIMD:
      - sta v bistvu enaka, saj vsak MIMD lahko naredimo SPMD
      - oboje lahko na običajnih računalnikih izvedemo z MPI

Polja procesorjev

- Vektorski računalnik
  - Zna računati na vektorjih in skalarjih
  - Dva tipa:
    - Cevovodna rešitev (starejši, cevovodno računanje arit. operacij)
    - Procesorska polja
      - Sekvenčni računalnik
      - Določene ukaze izvaja na posebnih sinhroniziranih procesnih elementih, ki so zmožni hkrati izvajati izračune na različnih podatkih
      - Razloga za njihov razvoj
        - Visoka cena glavnega procesorja
        - Večina izračunov v znanosti je podatkovno paralelnih

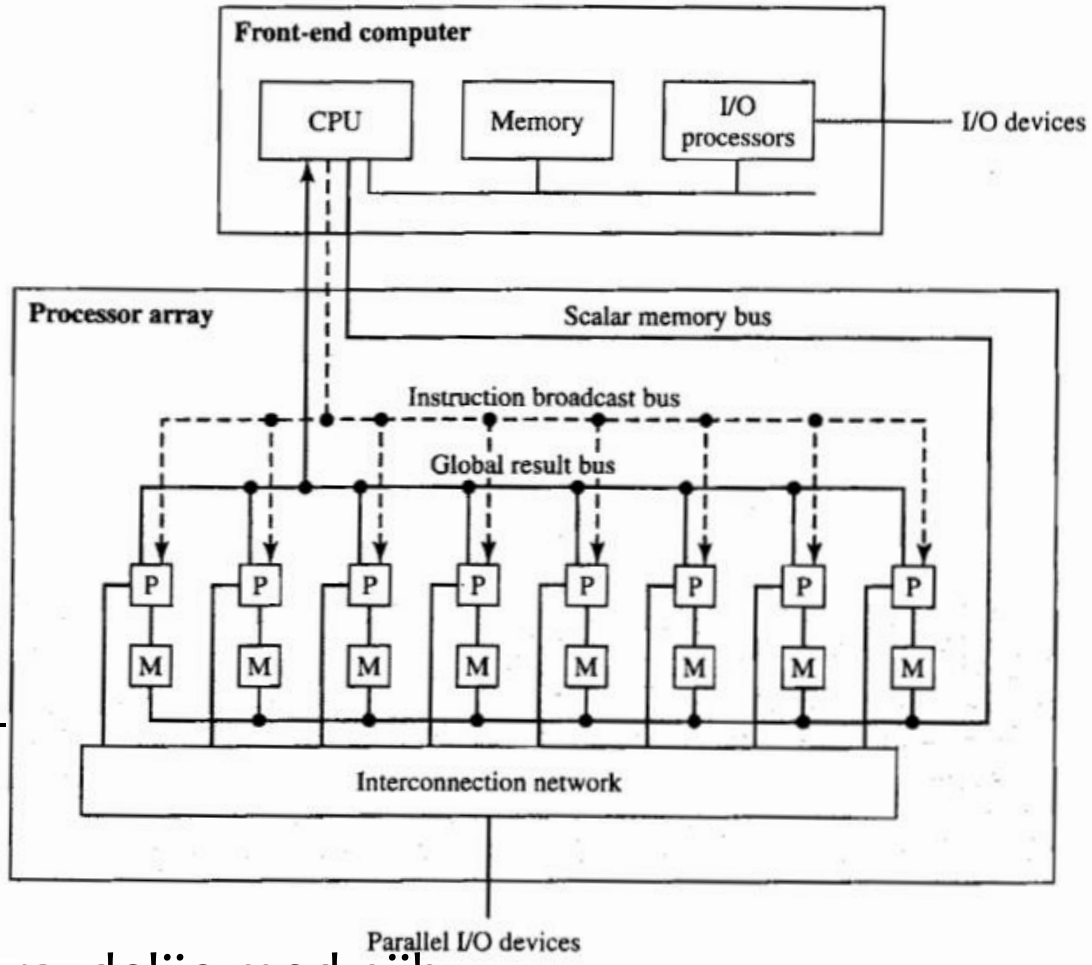
Polja procesorjev

- Vektorski računalnik ima
  - Vektorske registre
  - Vektorske in cevovodne funkcijske enote
  - Vektorski ukazi
  - Pomnilniško prepletanje (zaporedni naslovi v različnih modulih, da več procesorjev dostopa na enkrat do pomnilnika)
  - Poravnan zapis operandov v pomnilniku

Polja procesorjev

- Arhitektura
  - Front – end:

    enojedrni procesor
    - Pomnilnik za ukaze in sekvenčne operande
  - Polje procesorjev:
    - Mnogo parov enostaven procesor – pomnilnik
    - Podatki za paralelne izračune se najprej porazdelijo med njih
    - Posebno vodilo omogoča sinhrono izvajanje na vseh procesorjih



Polja procesorjev

- Računanje vsote vektorjev **c** = **a** + **b** na 128 procesorjih
  - |**a**|, |**b**| = 128
    - i – ti procesor dobi a_i in b_i in izračuna c_i
    - Izračun se sproži z enim ukazom
  - |**a**|, |**b**| = 28
    - 100 procesorjev je brezposelnih – neučinkovito
  - |**a**|, |**b**| = 555
    - 555 = 4 x 128 + 43
      - 43 procesorjev izvede 5 izračunov
      - 85 procesorjev izvede 4 izračune

Polja procesorjev

- Povezovanje osnovnih procesorjev
  - Najpogosteje 2D mreža
  - Računanje diferencialnih enačb v fiziki, strojništvu, dinamiki tekočin je tipa: $a_i = (a_{i-1} + a_{i+1}) / 2$
  - 2D mreža nam da povezavo procesorja s svojimi sosedi
- Pogojno izvrševanje ukazov
  - Posebni ukazi za maskiranje procesorjev
  - Operacije se izvede samo na nemaskiranih procesorjih
    - Primer: računanje predznaka elementa, testiranje, če je število manjše od 0
      - Maskiranje tistih, ki ustrezajo pogoju
      - Postavljanje vrednosti vseh nemaskiranih procesorjev na 1
      - Maskirane procesorje odmaksiramo, nemaskirane zamaskiramo
      - vrednosti nemaskiranih procesorjev postavimo na -1

Polja procesorjev

- Plusi in minusi
  - Velikost vsakega problema se ne ujema z velikostjo procesorskega polja
  - Počasno izvajanje pogojnih ukazov (gnezdeni stavki if, …)
  - Gre za enouporabniške računalnike, nekdo drug ne more izkoristiti prostih virov
  - Namenski procesorji – počasnejši in dražji razvoj kot za komercialne rešitve
- Programiranje
  - Možno avtomatsko prevajanje (zanka for …)
  - Slaba skalabilnost za velike probleme

Polja procesorjev

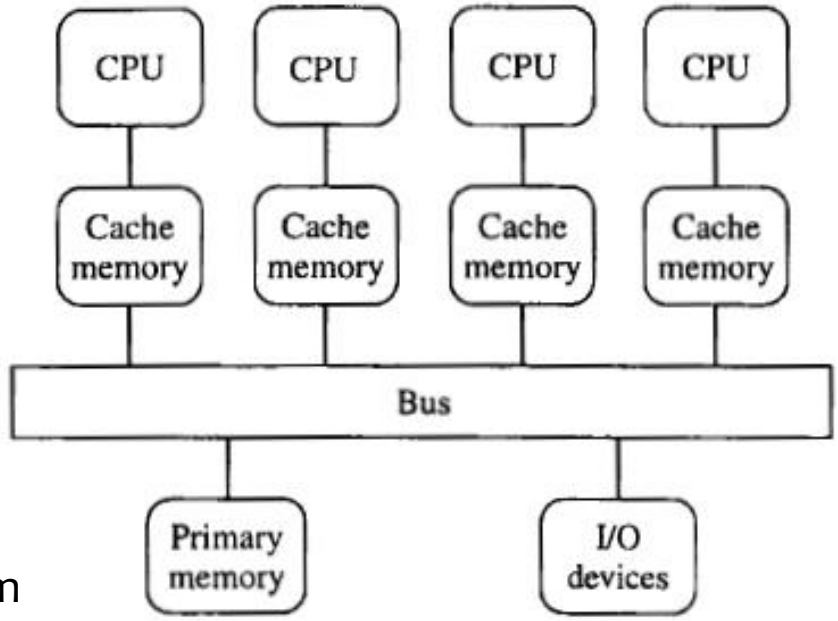
- GPU
  - hierarhična zasnova
  - Mutiprocesorji, senčilni procesorji
  - Senčilni procesorji izvajajo množico nitk
  - SIMD na senčilnih procesorjih
  - SPMD na multiprocesorjih
  - SIMT – single instruction multiple thread

Večprocesorski sistemi

- Gre za računalnike z več procesorji in deljenim pomnilnikom
- Prednosti pred procesorskimi polji:
  - Sestavljeni so iz komercialnih komponent
  - Običajno imajo vgrajeno podporo za večopravilnost
  - Ne izgubljajo na učinkovitosti pri izvajanju pogojne kode
- Tipi
  - Centralizirani večprocesorski sistemi
  - Porazdeljeni večprocesorski sistemi
- Dva velika problema
  - sovisnost (koherenca) podatkov
  - sinhronizacija

Večprocesorski sistemi

- Centralizirani večprocesorski sistemi
  - Naravna razširitev enoprocesorskega sistema
  - Procesorji so s skupnim pomnilnikom povezani preko vodila
  - UMA (Uniform Memory Access) ali SMP (Symmeric MultiProcessor)
  - Današnji večjedrni računalniki spadajo v to skupino
  - Procesor ima običajno nekaj lokalnega predpomnilnika, da se zmanjšajo prenosi po vodilu
    - Stalno branje spremenljivk s strani vseh procesorjev bi vodilo k mnogim prenosom in s tem celoten sistem silno upočasnilo



Večprocesorski sistemi

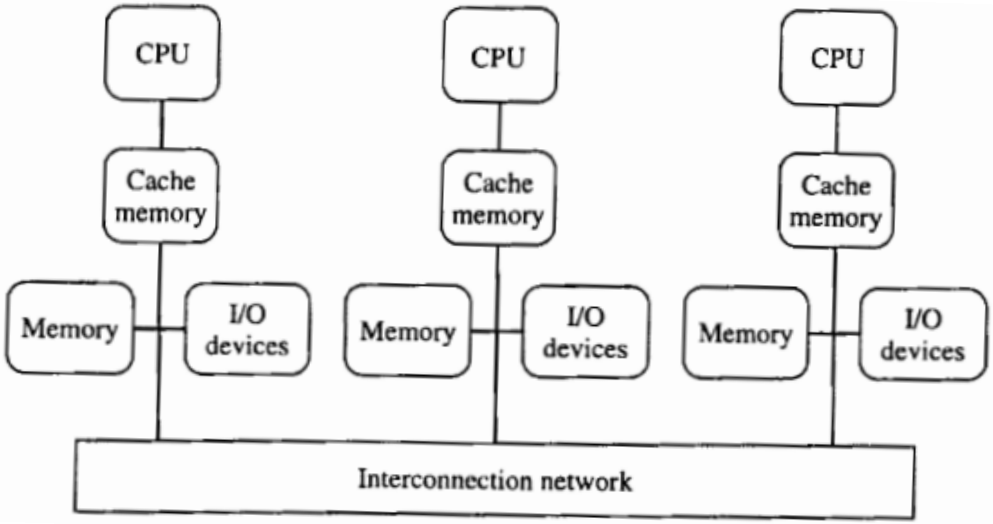
- Centralizirani večprocesorski sistemi
  - Pomnilnik je običajno razdeljen na več modulov
    - Do enega modula lahko dostopa samo en procesor na enkrat
    - Več modulov → več hkratnih dostopov
    - Prepletanje
      - Visokonivojsko:
        - zaporedni naslovi so v istem pomnilniku
        - Primer: 0-255, 256-511, 512-767, 768-1023 v štirih modulih
      - Nizkonivojsko:
        - Zaporedni naslovi so v modulih z zaporednimi oznakami
        - M0: 0, 4, 8, …, M1: 1, 5, 9, …, M2: 2, 6, 10, …, M3: 3, 7, 11, …
        - Super za vektorske operacije, če jih izvajajo različni procesorji (procesorska polja)

Večprocesorski sistemi

- Centralizirani večprocesorski sistemi: delovanje
  - Ločimo lokalne in deljene podatke
  - Procesorji med seboj komunicirajo preko deljenega pomnilnika

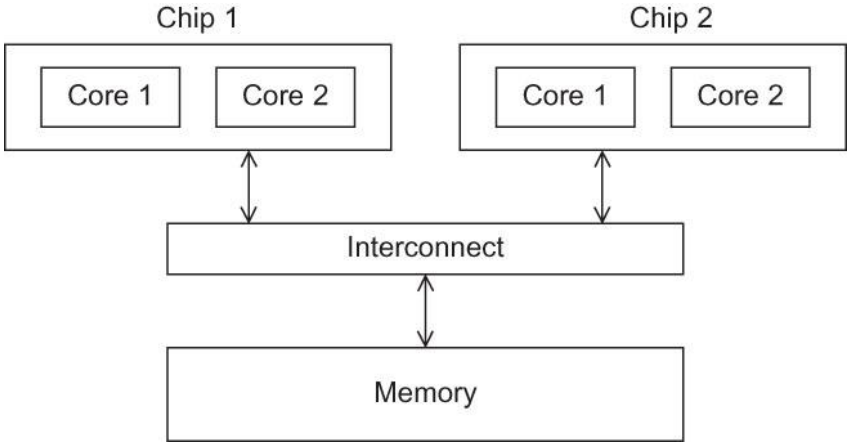
Večprocesorski sistemi

- Porazdeljeni večprocesorski sistemi
  - Porazdeljen je tudi pomnilnik
  - Dostop do lokalnega pomnilnika je bistveno hitrejši kot do deljenega
  - Govorimo tudi o NUMA (Non-Uniform Memory Access)
  - Imamo lahko več procesorjev kot pri UMA (boljša raztegljivost)
  - Vsi procesorji si delijo isti naslovni prostor
    - isti naslov na različnih procesorjih se nanaša na isti kos pomnilnika

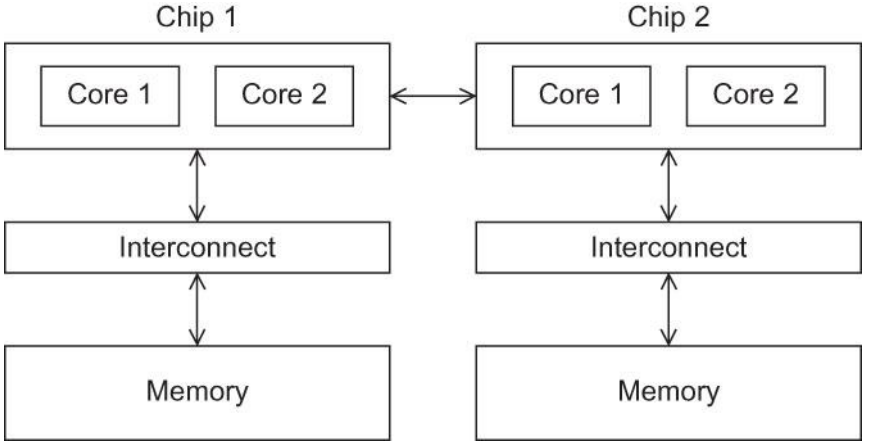


Večprocesorski sistemi

- Centralizirani večprocesorski sistemi
  - UMA večjedrnik

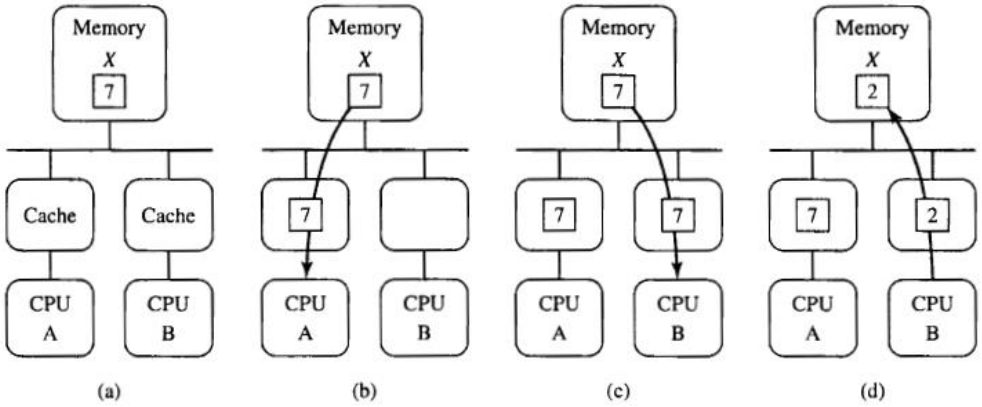


  - NUMA večjedrnik



Večprocesorski sistemi

- Centralizirani večprocesorski sistemi: sovisnost podatkov
  - Procesorji vidijo podatke preko predpomnilnika
  - Paziti je treba, da se podatki med procesorji ne razlikujejo
  - Primer



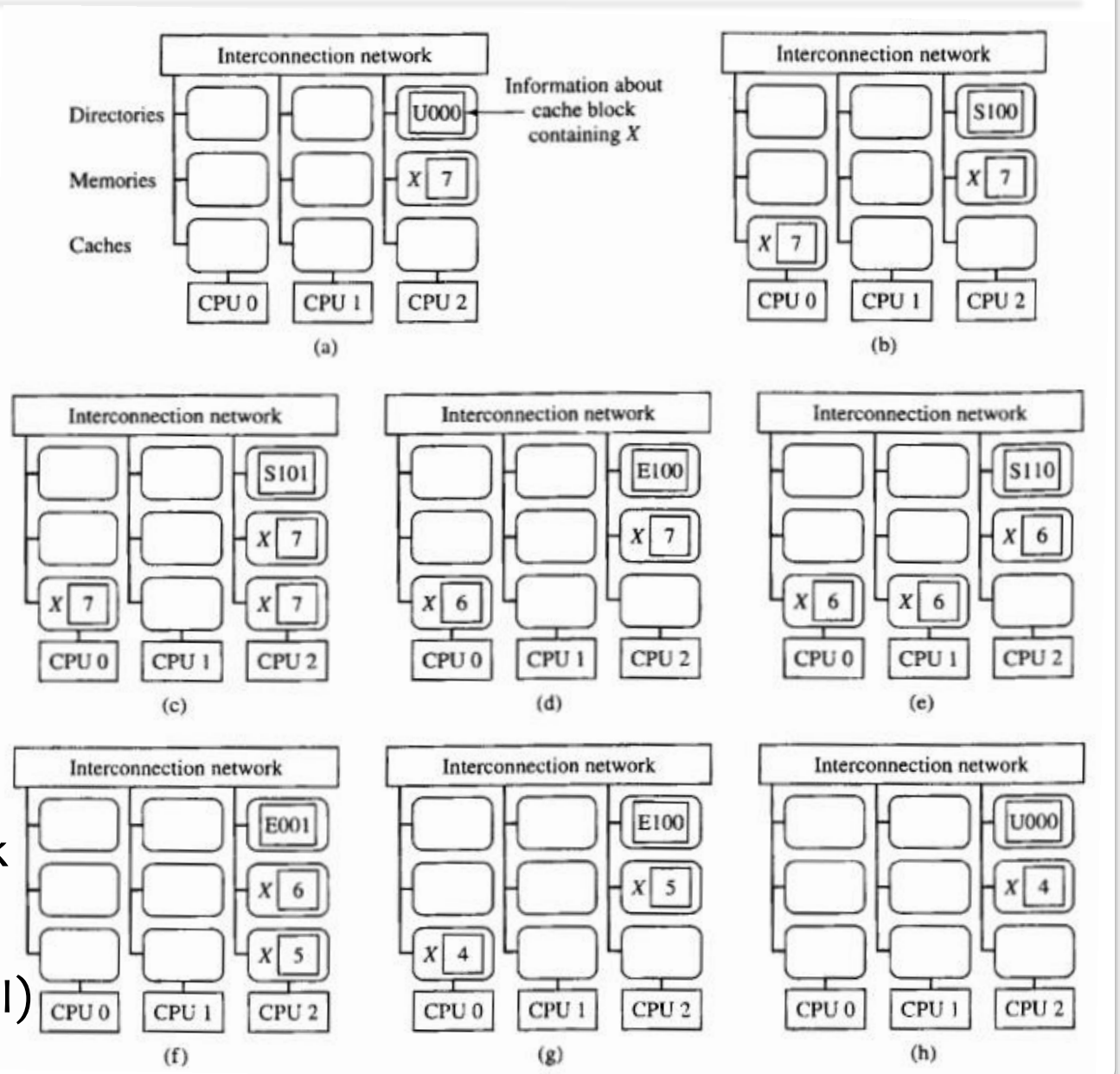
  - Rešitev
    - vohljanje (snooping) v kombinaciji s pisanjem skozi
    - Direktorijski protokoli (MESI, …) v kombinaciji s pisanjem nazaj

Večprocesorski sistemi

- Porazdeljeni večprocesorski sistemi: sovisnost
  - Zaradi večjega števila procesorjev vohljanje postane ozko grlo
  - Potrebni so boljši protokoli
    - lista o razporeditvi pomnilniških blokov po predpomnilnikih
    - Primer protokola:
      - Stanje vodi pomnilnik, ki je lastnik določenega bloka
      - U(ncached) – pomnilniški blok ni uporabljen na nobenem predpomnilniku
      - S(hared) – zapis je uporabljen v predpomnilnikih (branje)
      - E(xclusive) –zapis v predpomnilnik, samo en predpomnilnik ima pravo vrednost

Večprocesorski sistemi

- Porazdeljeni večprocesorski sistemi: sovisnost
  - Primer protokola
    - 1/0 se nanaša na uporabljeni predpomnilnik
    - (h) = flush
    - Pentium (MESI)



Večprocesorski sistemi

- Porazdeljeni večprocesorski sistemi: sovisnost
  - Pentium:
    - M(ultiple)– večkratni zapis v predpomnilnik, samo en procesor ima veljavno kopijo
    - E(xclusive) – prvi zapis v predpomnilnik, samo en procesor ima veljavno kopijo
    - S(hared) – blok ima kopije v več predpomnilnikih
    - I(nvalid) – zapisani blok ni ustrezen ali ni v predpomnilniku

Večprocesorski sistemi

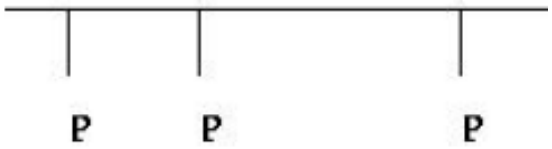
- Večprocesorski sistemi: sovisnost
  - Lažni deljeni podatki
    - Spremenljivki Z in W se nahajata v istem pomnilniškem bloku
    - Ko preberemo Z, se v predpomnilnik prenese tudi W
    - Ko spremenimo Z, bo blok na vseh ostalih procesorjih označen kot neustrezen E(xclusive)
    - S tem bo kot neustrezna označena tudi spremenljivka W, čeprav se njena vrednost ni spremenila
    - Pri spreminjanju Z na enem in W na drugem procesorju se bo ves čas osveževal predpomnilnik → zmanjšanje zmogljivosti

Večprocesorski sistemi

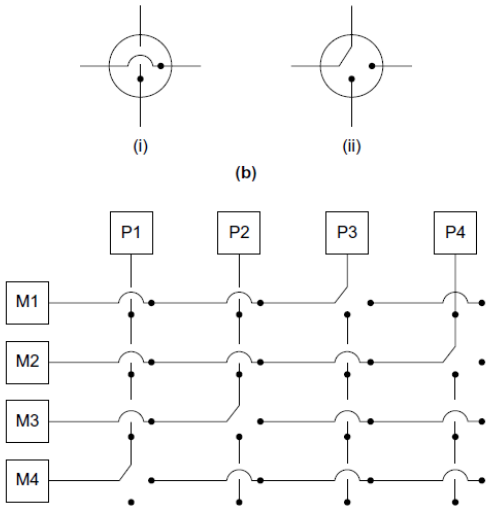
- Večprocesorski sistemi: sinhronizacija
  - Medsebojno izključevanje:
    - določene operacije lahko na enkrat dela samo en procesor
    - Primer: premikanje kazalca na naslednjo nalogo
      - Prvi procesor vzame prvo nalogo in kazalec premakne na naslednjo
      - Drugi vzame nalogo na katero kaže kazalec in premakne kazalec preden naslednji procesor prebere nalogo …
  - Sinhronizacija s preprekami
    - Dokler vsi procesorji ne pridejo do istega dela programa, označenega kot prepreka, nobeden ne sme nadaljevati izvajanja

Povezovalna omrežja

- Večprocesorski sistemi
  - vodila
    - žice si delijo vse naprave,
    - nizka cena, fleksibilnost
    - samo dve napravi na enkrat lahko komunicirata,
    - Več kot je procesorjev, bolj čakajo na dostop do pomnilnika



  - Mreža
    - Preklopni sistem
    - Poti med napravami se nastavlja s stikali
      - Kvadrati – n x procesor / n x pomnilnik
      - Stikala - nxn možnih poti
      - Za manjše sisteme
      - n hkratnih povezav



Pohitritev

- Definicija:

$$S(n,p) = \frac{t_s(n)}{t_p(n,p)}$$

  - čas izvajanja sekvenčnega programa $t_s(n)$
  - čas izvajanja paralelnega programa $t_p(n,p)$
  - velikost problema n
  - število paralelnih procesov p

  - wall-clock time!

Pohitritev

- Paralelni algoritmi so sestavljeni iz treh skupin operacij:
  - operacije, ki jih je potrebno izvajati zaporedno, delež $\sigma(n)$,
  - operacije, ki jih lahko paraleliziramo, delež $\varphi(n)$ in
  - komunikacija med procesi, delež $\kappa(n,p)$.
- Če predpostavimo, da se paralelne operacije enakomerno porazdelijo med procese, velja

$$S(n,p) = \frac{t_s(n)}{t_p(n,p)} = \frac{\sigma(n) + \varphi(n)}{\sigma(n) + \varphi(n)/p + \kappa(n,p)}$$

Učinkovitost

- Definicija:

$$E(n,p) = \frac{t_s(n)}{p \cdot t_p(n,p)} = \frac{S(n,p)}{p}$$

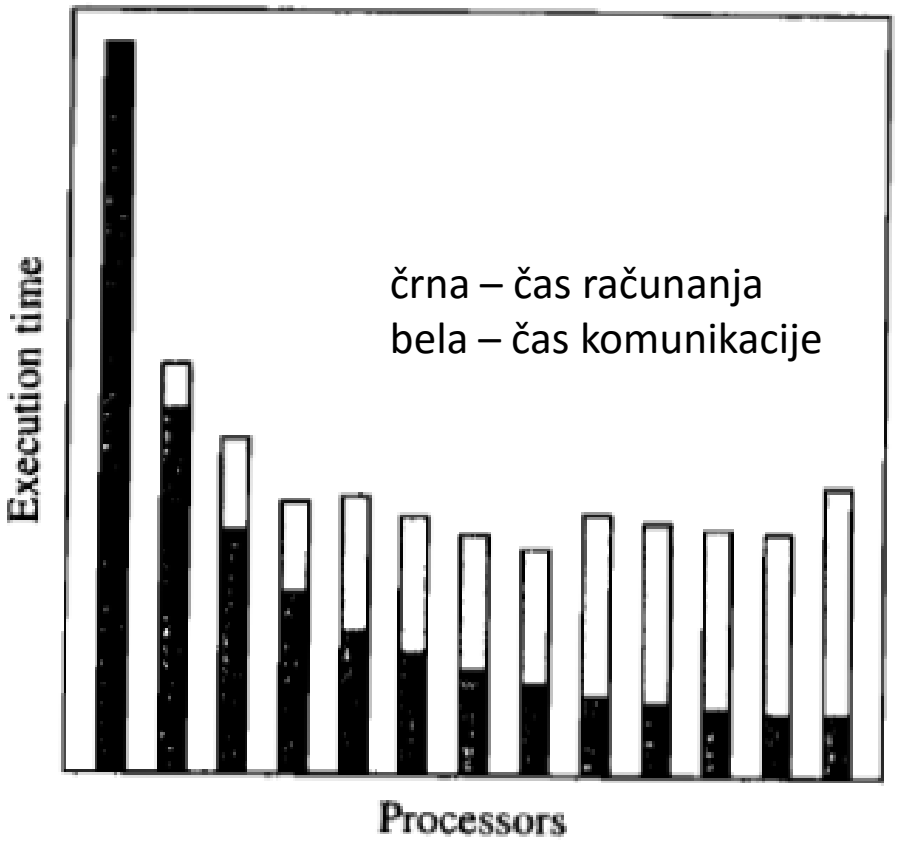
  - Je mera izrabe procesorjev
  - Pove nam kolikšen delež časa so procesi p polno izkoriščeni

$$E(n,p) = \frac{\sigma(n) + \varphi(n)}{p\sigma(n) + \varphi(n) + p\kappa(n,p)}$$

  - Običajno velja $0 \leq E(n,p) \leq 1$

Računanje in komunikacija

- Dodajanje procesorjev zmanjšuje čas računanja
- V neki točki se zgodi, da je strošek komunikacije večji od zmanjšanje časa računanja. Takrat se začne čas izvajanja programa povečevati.



$$S(n,p) = \frac{\sigma(n) + \varphi(n)}{\sigma(n) + \varphi(n)/p + \kappa(n,p)}$$

Računanje in komunikacija

- Linearna in nad-linearna pohitritev
  - Če je $E(n,p) = 1$ ali $S(n,p) = p$ govorimo o linearni pohitritvi
  - Če je $E(n,p) > 1$ ali $S(n,p) > p$ govorimo o nad-linearni pohitritvi
    - Do nad-linearne pohitritve lahko pride v praksi, na primer, zaradi vpliva organizacije pomnilnikov na izvajanje programa
      - Sekvenčni program:
        velik problem → premalo glavnega pomnilnika → uporaba navideznega
      - Paralelni program:
        problem je razdeljen → na vsakem procesorju je dovolj pomnilnika → uporaba navideznega pomnilnika ni potrebna

Cena računanja

- Definicija

$$P(n,p) = pt_p(n) = \frac{pt_s(n)}{S(n,p)} = \frac{t_s(n)}{E(n,p)}$$

  - Bolj kot je program učinkovit, bolj so procesorji izkoriščeni, zato izračun poteka manj časa in je cena nižja

Amdahlov zakon

- Še enkrat pohitritev

$$S(n,p) = \frac{\sigma(n) + \varphi(n)}{\sigma(n) + \varphi(n)/p + \kappa(n,p)}$$

  - Vzemimo, da poznamo delež operacij, ki jih moramo izvajati zaporedno

$$f = \frac{\sigma(n)}{\sigma(n) + \varphi(n)}$$

Amdahlov zakon

- Še enkrat pohitritev
  - ker je $\kappa(n,p) \geq 0$, velja

$$S(n,p) = \frac{\sigma(n) + \varphi(n)}{\sigma(n) + \varphi(n)/p + \kappa(n,p)} \leq \frac{\sigma(n) + \varphi(n)}{\sigma(n) + \varphi(n)/p}$$

  - Če iz enačbe za f izrazimo $\varphi(n)$ in ga vstavimo v neenakost za $S(n,p)$, dobimo Amdahlov zakon

$$S(n,p) \leq \frac{1}{f + (1-f)/p}$$

  - Amdahlov zakon predpostavlja, da je:
    - velikost problema konstantna
    - fokus paralalizacije na skrajšanju časa izvajanja programa

Amdahlov zakon

- Prekletstvo parametra *f*
  - delež kode, ki ga ne moremo izvajati paralelno
    - koda, ki je ne moremo paralelizirati,
    - čas, potreben za komunikacijo,
    - čas, ki ga procesi porabijo za čakanje na pošiljanje/prejemanje podatkov,
    - čas, potreben za čakanje ostalih procesov, da zaključijo z delom,
    - čas, potreben za klicanje funkcij sistema MPI.
  - … če je tega samo 5% je največja pohitritev samo dvajsetkratna

sekvenčni delež	največja pohitritev
0 %	∞
5 %	20
10 %	10
20 %	5
25 %	4