

A számítógépek felépítése 11.: A párhuzamos gépek alapfogalmai

Markó Tamás
PTE TTK, 2003

2003.12.01.

Markó Tamás, PTE TTK

1

A rádiótelefonokat kérem KIKAPCSOLNI!

2003.12.01.

Markó Tamás, PTE TTK

2

Miért és hogyan párhuzamosítunk?

- Indokok
 - kell a nagyobb teljesítmény
 - a ciklusidő nem csökkenthető akármeddig
 - kisebb teljesítményű komponensek párhuzamosítása reális alternatíva
- Tervezési alapkérdések
 - a feldolgozóelemek minősége, mérete és mennyisége
 - a memória-modulok minősége, mérete és mennyisége
 - a feldolgozóelemek és a memória-modulok összekapcsolási módja

2003.12.01.

Markó Tamás, PTE TTK

3

A feldolgozóelemek

- Ha kicsik (egy chip tötrésze), akkor sok is összeépíthető (akár egymilliónál is több)
- Ha komplett számítógépek, akkor jóval kevesebb
- Gyakori a kommersz processzorokból felépített párhuzamos rendszer

2003.12.01.

Markó Tamás, PTE TTK

4

A memória-modulok

- Egymástól függetlenül működnek
- Egyidejűleg (esetleg) több CPU-ból is elérhetők

2003.12.01.

Markó Tamás, PTE TTK

5

Az összekapcsolás

- Laza együttműködés: kevés CPU lassú kapcsolattal
 - korlátozott kommunikáció
 - durva tagoltságú, egymással alig együttműködő programok párhuzamos futtatására alkalmas
- Szoros együttműködés: sok kicsi komponens nagysebességű interaktív kapcsolattal
 - a finoman tagolt párhuzamosított programokhoz alkalmas

2003.12.01.

Markó Tamás, PTE TTK

6

Kommunikációs modellek

- Ugyanazon feladat különböző részeivel foglalkozó CPU-k kommunikációjáról van szó
- Alapvető megoldások:
 - multiprocesszorok
 - multiszámítógépek
- Mindkettőt alkalmazzák a gyakorlatban

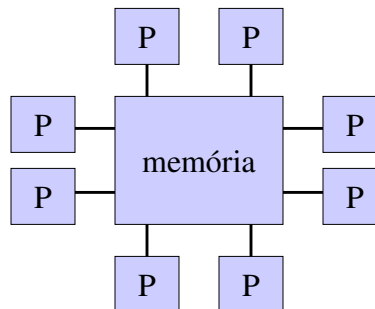
2003.12.01.

Markó Tamás, PTE TTK

7

A multiprocesszorok

- Több CPU közös memóriát használ („közös memóriájú rendszer”)
- Kommunikáció a memórián keresztül
- Jól érthető
- Megépíteni bonyolultabb



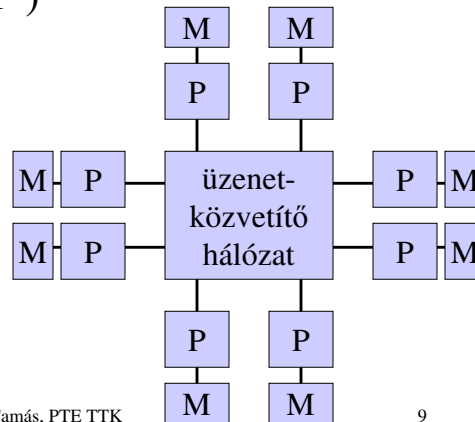
2003.12.01.

Markó Tamás, PTE TTK

8

A multiszámítógépek

- Minden CPU-nak saját memória („osztott memóriájú rendszer”)
- Kommunikáció üzenetekkel
- A programok bonyolultabbak
- Megépíteni egyszerűbb



2003.12.01.

Markó Tamás, PTE TTK

9

A memória megosztási lehetőségei

- A megosztás különböző szinteken valósítható meg:
 - hardverben (multiprocesszoros rendszer)
 - a virtuális memória szintjén
 - az alkalmazás szintjén

2003.12.01.

Markó Tamás, PTE TTK

10

A memória megosztása hardverben

- Ez a multiprocesszoros rendszer



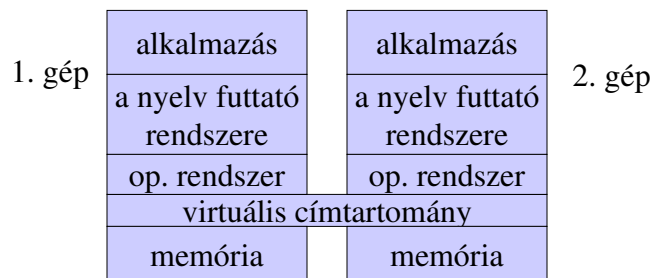
2003.12.01.

Markó Tamás, PTE TTK

11

A memória megosztása a virtuális memória szintjén

- DSM, Distributed Shared Memory
- Laponként változik, hogy ki használja a virtuális memóriát



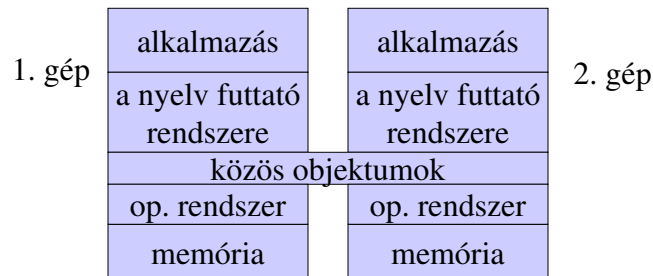
2003.12.01.

Markó Tamás, PTE TTK

12

A memória megosztása az alkalmazás szintjén

- A nyelvben lehet közös objektumokra hivatkozni
- A megvalósítás a fordító és a futtató rendszer dolga



2003.12.01.

Markó Tamás, PTE TTK

13

Az összekötő hálózat

2003.12.01.

Markó Tamás, PTE TTK

14

Az összekötő hálózat komponensei

- CPU-k
- memória-modulok
- interfészek
 - a CPU-knál és a memória-moduloknál lévő eszközök, amik az üzeneteket küldik és fogadják
- kapcsolatok
 - fizikai csatornák, amiken a bitek mozognak
- kapcsolók
 - több bemeneti és kimeneti kapuval rendelkező eszközök: a bemeneten megjelenő csomagot valamelyik kimenetre továbbítják

2003.12.01.

Markó Tamás, PTE TTK

15

Az összekötő hálózatok topológiája 1.

- Gráfokkal modellezhető
 - élek: kapcsolatok
 - csomópontok: kapcsolók
- A legfontosabb jellemzők:
 - egy csomópont fokszáma: a befutó élek száma (hibatűrés!)
 - két csomópont távolsága: azon élek minimális száma, amiken keresztül el lehet jutni egyikből a másikba
 - a gráf átmérője: a két legtávolabbi csomópont távolsága (legrosszabb haladási idő)
 - dimenzió: a forrásból a célba való eljutás választási lehetőségeinek száma (csak egy lehetőség: 0 dimenzió)
 - kettévágott sáv szélesség

16

A kettévágott sávszélesség

- Meghatározása:
 - A hálózatot két egyenlő (azonos számú csúcsot tartalmazó) részre bontjuk
 - Vesszük a megszüntetett kapcsolatok sávszélességeinek összegét
 - Minden lehetséges kettévágásra vesszük ennek a minimumát
- Ennél nem lehet kisebb a két fél közötti kommunikáció sebessége

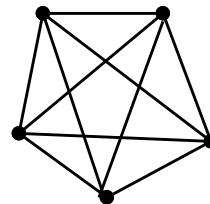
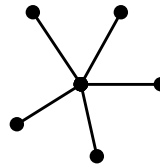
2003.12.01.

Markó Tamás, PTE TTK

17

Az összekötő hálózatok topológiája 2.

- Csillag
 - 0 dimenzió
 - nem hibatűrő
 - a központ lehet a szűk keresztmetszet
- Teljes gráf
 - sokdimenziós
 - maximális hibatűrés
 - 1 átmérő
 - maximális kettévágott sávszélesség
 - hátrány: túl sok kapcsolat ($\sim n^2$)



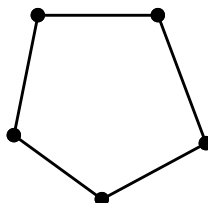
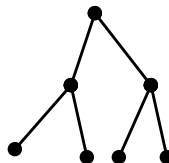
2003.12.01.

Markó Tamás, PTE TTK

18

Az összekötő hálózatok topológiája 3.

- Fa
 - 0 dimenzió
 - nem hibatűrő
 - a gyökere közelében szűk keresztmetszet
- Gyűrű
 - 1 dimenzió



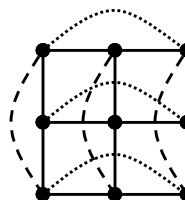
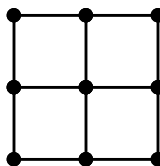
2003.12.01.

Markó Tamás, PTE TTK

19

Az összekötő hálózatok topológiája 4.

- Rács (háló)
 - 2 dimenzió
 - könnyű növelni
 - az átmérő $\sim \sqrt{n}$
 - gyakori
- Kettős tórusz
 - 2 dimenzió
 - széleivel összekapcsolt rács
 - jobb hibatűrés
 - kisebb átmérő



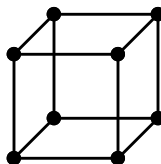
2003.12.01.

Markó Tamás, PTE TTK

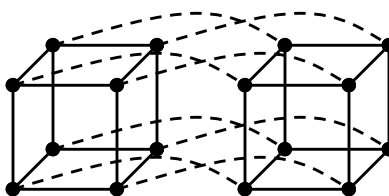
20

Az összekötő hálózatok topológiája 5.

- Kocka
 - 3 dimenzió



- Hiperkocka
 - itt 4 dimenzió (több is lehet)
 - nagyteljesítményű rendszereknél gyakori



2003.12.01.

Markó Tamás, PTE TTK

21

A kapcsolás

- Áramkörkapcsolás
 - a csomag csak akkor indul el, ha a teljes útja le van már foglalva
 - hosszabb előkészítés, gyors továbbítás
- „Tárold és továbbítsd” kapcsolás
 - a csomag a következő kapcsolóig megy, ott tárolódik, amíg a következő útszakasz szabaddá válik
 - rövid előkészítés, hosszadalmas és kiszámíthatatlan továbbítás

2003.12.01.

Markó Tamás, PTE TTK

22

Útvonalválasztó algoritmusok

- A jó algoritmus
 - szétesztja a terhelést több alternatív összeköttetésre
 - kerüli a holtponatok kialakulását
- Forrásszintű útvonalválasztás
 - előre ismert a telje útvonal, az egyes kapuknál követendő irányt egy listaként a csomaghoz kapcsolják
 - minden kapu a lista első eleme szerint küldi tovább, ezt az adatot pedig levágja a listáról
- Osztott útválasztás
 - minden kapcsoló maga dönti el, hogy merre továbbít

2003.12.01.

Markó Tamás, PTE TTK

23

Párhuzamos gépek és a szoftver

2003.12.01.

Markó Tamás, PTE TTK

25

A teljesítmény szoftveres mértéke

- Hányszor gyorsabban fut egy adott program, mint egy egyprocesszoros gépen
- A processzorszám függvényében ábrázolható
- Mindig a lineáris gyorsítás alatt marad, de függ a problémától
 - az n-test probléma vizsgálata jó
 - nagy mátrix invertálása rossz (ötszörösnél nagyobb gyorsítás nem érhető el)
- **Skálázható rendszer: a CPU-k számának növelésével arányosan nő a teljesítmény**

2003.12.01.

Markó Tamás, PTE TTK

26

A hatékony szoftver

- Szükség van a párhuzamos hardver lehetőségeit kihasználó programokra is
- Sokféle fejlesztés történik
- Általában a meglévő programozási nyelveket bővítik új szerkezetekkel

2003.12.01.

Markó Tamás, PTE TTK

27

A párhuzamos számítógépek osztályozása

2003.12.01.

Markó Tamás, PTE TTK

29

Egy lehetséges osztályozás

Az utasítás-áramok (=utasításszámlálók) és az adatáramok száma alapján

- SISD: Single Instruction stream, Single Data stream
 - klasszikus Neumann-elvű gép, soros működés
- SIMD: Single Instruction stream, Multiple Data stream
 - egyetlen vezérlőegység, több ALU
- MISD: Multiple Instruction stream, Single Data stream
 - nem ismert ilyen gép
- MIMD: Multiple Instruction stream, Multiple Data stream
 - multiprocesszor, multiszámítógép

2003.12.01.

Markó Tamás, PTE TTK

30

A SIMD gépek

- Vektorokkal és tömbökkel végzett (tudományos) számításokhoz
- Egyetlen vezérlő egység, soros utasítás-végrehajtás
- Minden utasítás több adatelemen hajtódik végre (ez jelenti a párhuzamosságot)

2003.12.01.

Markó Tamás, PTE TTK

31

SIMD gépek - a tömbprocesszor

- Egy vezérlő, több **független** ALU
- A feldolgozó egységek mérete az 1 bitestől a lebegőpontos aritmetikai egységig bármi lehet
- A feldolgozó egységek jellemző összekötési módja a rács
- A feldolgozó egységeknek belső autonómiája is lehet
 - egyes rendszerekben eldönthetik, hogy végrehajtanak-e egy utasítást, vagy nem
- A tömbprocesszorok jövője bizonytalan

2003.12.01.

Markó Tamás, PTE TTK

32

SIMD gépek - a vektorprocesszor

- Ide tartoznak a Cray gépek is
- Ugyanaz a művelet egy vektor minden elemére
- Szokásos művelet-típusok:
 - $A_i = f(B_i)$ pl. elemenkénti négyzetgyökvonás
 - $x = f(A)$ pl. a vektor elemeinek összege
 - $A_i = f(B_i, C_i)$ pl. elemenkénti összeadás
 - $A_i = f(x, B_i)$ pl. skalárral való szorzás

2003.12.01.

Markó Tamás, PTE TTK

33

MIMD - multiprocesszorok 1.

- A közös memóriában egyetlen közös operációs rendszer
- A memória közös használatára többféle szabály vonatkozhat
- A perifériák lehetnek közösek (SMP, symmetric multiprocessor), vagy tartozhatnak csak bizonyos CPU-khoz

2003.12.01.

Markó Tamás, PTE TTK

34

MIMD - multiprocesszorok 2.

- UMA (Uniform Memory Access)
 - minden memória-modul ugyanolyan gyorsan érhető el
- NUMA (Non-Uniform Memory Access)
 - a közeli modulok elérése gyorsabb
 - fontos, hogy a kód és az adat hol helyezkedik el
- COMA (Cache-Only Memory Access)
 - minden CPU a saját memóriáját úgy használja, mintha az cache lenne

2003.12.01.

Markó Tamás, PTE TTK

35

MIMD - multiszámítógépek

- MPP (Massively Parallel Processors)
 - nagyszámú CPU szoros kapcsolatban egy nagysebességű hálózattal
 - ilyenek a drága szuperszámítógépek
- COW (Cluster Of Workstations)
 - szokásos PC-k vagy munkaállomások hagyományos hálózattal összekötve

2003.12.01.

Markó Tamás, PTE TTK

36

Az UMA sínrendszerű architektúrák problémái

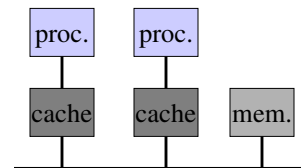
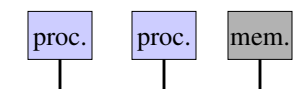
2003.12.01.

Markó Tamás, PTE TTK

38

A sín a szűk keresztmetszet

- A CPU-k versengenek a használatáért
- Már 32 CPU-val is használhatatlan
- A processzoronkénti cache csökkenti a busz forgalmát
- Probléma: sérülhet a cache konzisztenciája



- ugyanaz a sor több cache-ben is megvan
- az egyikben módosul \Rightarrow a másikban érvénytelenné válik

39

A szaglászó gyorsítótár

- Biztosítja a gyorsítótárak konzisztenciáját
- A cache írása írásátesztő (write-through), a memória is mindig frissül
- A többi cache figyeli az ilyen eseményeket
- Ha megvan benne az illető sor, akkor
 - vagy érvényteleníti
 - vagy rögtön frissíti a memóriából

2003.12.01.

Markó Tamás, PTE TTK

41

A MESI protokoll

- A cache sorainak négyféle állapota lehet:
 - **Modified**: érvényes bejegyzés, a memória **nincs** frissítve
 - **Exclusive**: másik cache nem tartalmazza ezt a sort, a memória frissítve van
 - **Shared**: több cache is tartalmazhatja, a memória frissítve van
 - **Invalid**: érvénytelen adatok
- Az állapotokat a sín forgalmának figyelése alapján tudják váltani

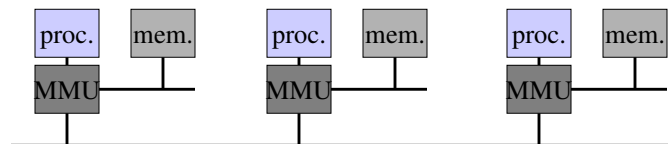
2003.12.01.

Markó Tamás, PTE TTK

42

NUMA multiprocesszorok

- Az egyes CPU-knál helyi memória
- A **címtartomány közös**, minden CPU-ból látható az egész
- A program szintjén nincs különbség a helyi és a távoli memória-modulok elérése között
 - távolra is a LOAD és a STORE használható, az MMU elintézi



2003.12.01.

Markó Tamás, PTE TTK

44

MPP: masszív párhuzamos rendszerek

- Több ezer CPU is összeköthető
- Általában kommersz processzorokat használnak (Pentium, UltraSPARC (Sun), RS/6000 (IBM), Alpha (DEC, Compaq, HP))
- Nagyteljesítményű hálózat: nagy sávszélesség, kis késleltetés
- Nagy I/O-kapacitás a háttértárak felé
- Hibatűrés
 - több ezer CPU-nál heti néhány hiba elkerülhetetlen
 - speciális hardver és szoftver figyel és elhárítja
 - pl. Cray T3E: minden 128 CPU-hoz egy tartalék, ami szükség esetén „beugrik”

46

COW: munkaállomások klasztere

- Néhány száz PC vagy munkaállomás, kommersz hálózat
- Olcsó - fokozatosan kiszorítja az MPP-ket
- Központosított COW:
 - egy helyiségben összezsúfolva
 - csak erre használt gépek, homogén géppark
 - kevés periféria
- Széttagolt COW:
 - egy intézmény gépei helyi hálózatban, heterogén géppark
 - naponta csak néhány órán keresztül használhatók így
 - a gépen futó központi feladat elvándoroltatható, ha a felhasználónak kell a gépe

47

A világ leggyorsabb számítógépei

- Félévente frissített lista:
<http://www.top500.org>
- Többségük sok olcsó processzorból (pl. Pentium) áll, masszív párhuzamos rendszerek
- Jelentős a vektorprocesszoros rendszerek aránya is
- A teraflop (10^{12} FLOP) sebességet már átlépték, 2010-ig várják a petaflop (10^{15} FLOP) gépet