



Adatbáziskezelő-architektúrák

Adatbázisok elmélete 2. előadás

Gajdos Sándor



Tartalom

- Architektúra általában
- Centralizált DBMS
- Kliens-szerver DBMS
- Párhuzamos DBMS
- Elosztott DBMS
- Példák

DBMS architektúra általában I.

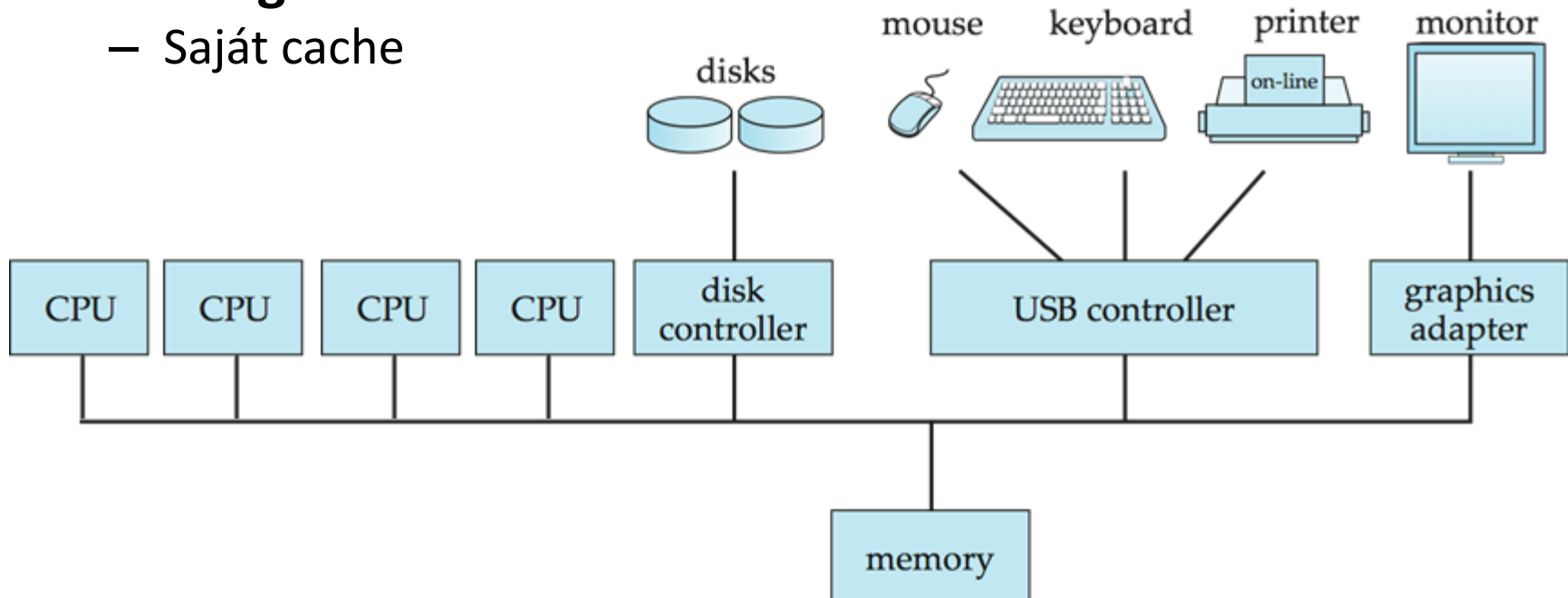
- Definíció:
 - „Rendszertervezési döntés, amely általában nem könnyen változtatható meg”
 - „Fontosabb szempontok, amiket figyelembe kell venni.”
 - Konceptcionális architektúra
 - Adat(konzisztencia) architektúra
 - Eszközarchitektúra (HW, SW)
 - Üzemeltetési architektúra
 - Biztonsági architektúra
 - ...

DBMS architektúra általában II.

- Standard (hardver-oprendszer-DBMS, mint alkalmazás) és egyedi megvalósítás
- Platform hatása
- Kezdetben: centralizált (egyetlen gép erőforrásai határoznak meg mindent)
- Meghatározó tényezők:
 - Hálózati lehetőségek megjelenése: egyes taskok máshol futhatnak -> kliens/szerver rendszerek
 - Sebesség-/teljesítménynövelés igénye: párhuzamos rendszerek
 - Adatok kezelése és/vagy elérhetőségének biztosítása több helyről, védettség -> elosztott rendszerek

Centralizált DBMS I.

- Def.: Egyetlen gépen fut (PC...mainframe), más rendszerekkel nincs interakció
- Általános szgép egy vagy néhány CPU-val, számos kontrollerrel, mind közös buszon a megosztott memóriával
- **Versengés a memóriáért**
 - Saját cache

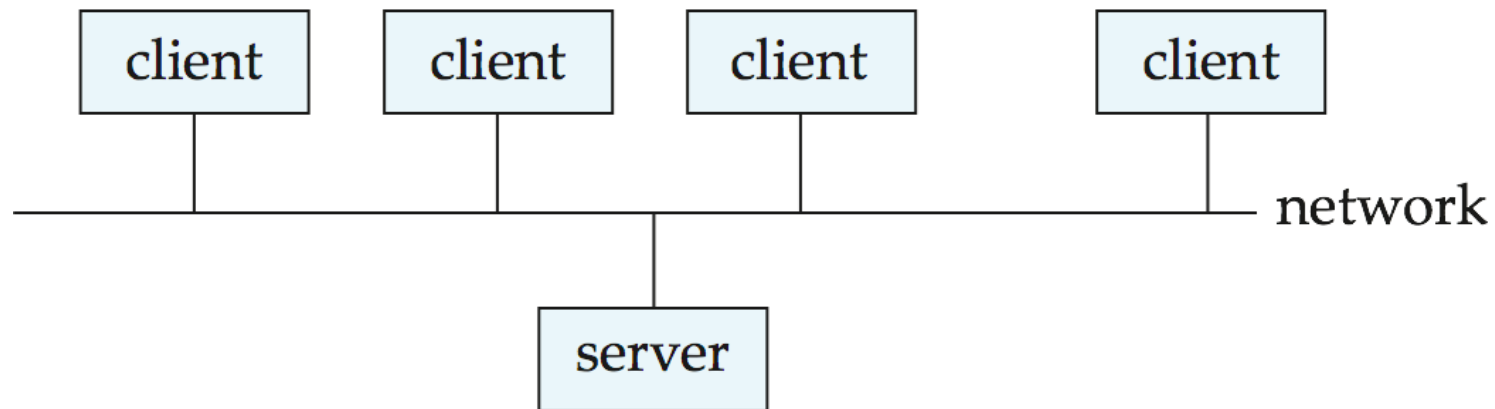


Centralizált DBMS II.

- Lehet egyfelhasználós, többfelhasználós(-> terminálok)
 - Egyfelhasználós rendszer: nem támogatja a konkurens működést, hiányzó v. egyszerű mentési megoldások, *durva granularitású lekérdezési* párhuzamosság
 - Tkp. egyprocesszorosnak érzékelhető
- Párhuzamosítás egyetlen processzoron
 - Processzek időosztásban, látszólagos párhuzamosság
 - Mikor van ennek értelme?
- *Finom granularitású* párhuzamosság: tip. sok processzor mellett

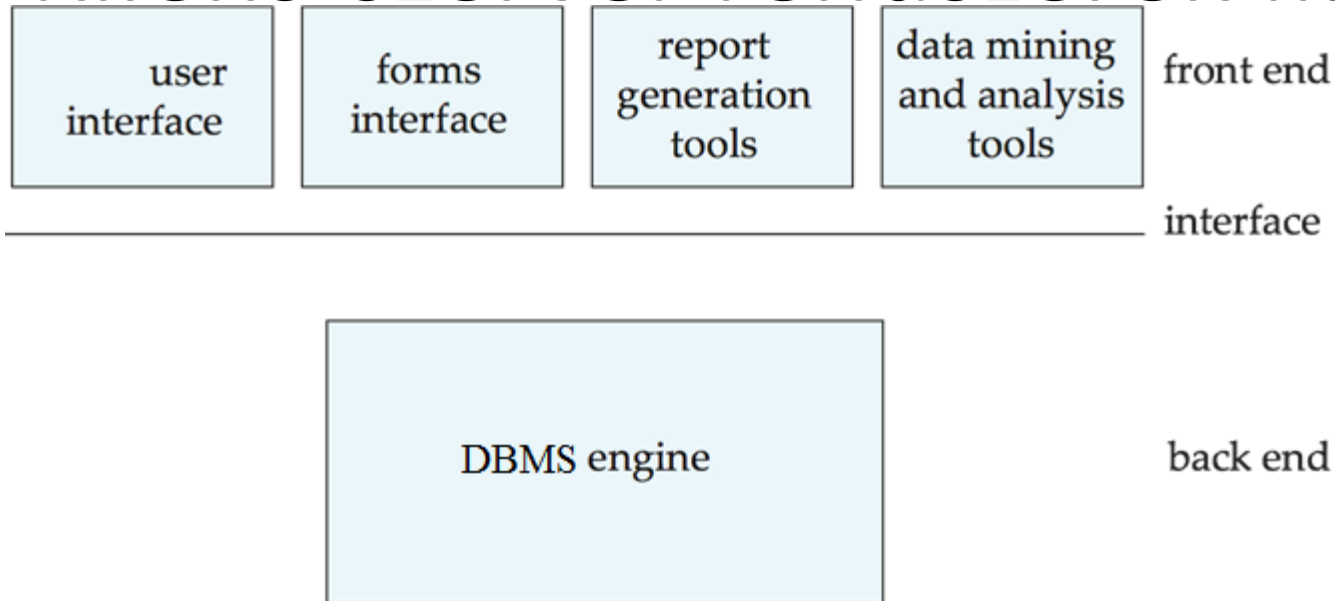
Kliens-szerver rendszerek I.

- Amikor a terminálokat kiszorították a PC-k...



- Back-end: adatok kezelése, lekérdezés feldolgozás, konkurencia és helyreállítás
- Front-end: lekérdezések összeállítása, eredmény megjelenítése
- Interfész: SQL alapú vagy API

Kliens-szerver rendszerek II.



A BE lehet:

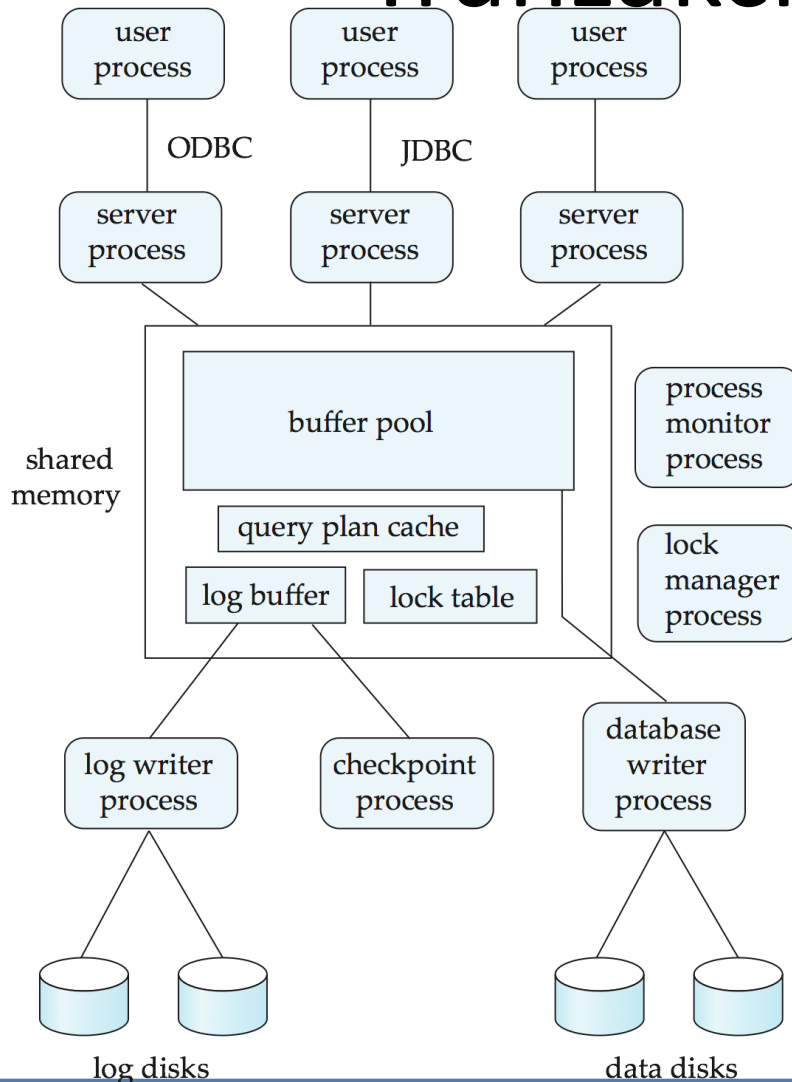
- Tranzakciós szerver (tip: relációs, hálós)
- Adatszerver (tip: OODB)



Tranzakciós szerver I.

- Tranzakciós szerver
 - Kliens tr. kéréseket küld a szerver felé, az végrehajtja, eredmény vissza
 - Kérés tip. SQL-ben és RPC-n keresztül, tranzakciós RPC
 - API-k definiálása: ODBC-JDBC, szállítók szétválása
 - Számos processzből áll, melyek az adatokat a megosztott memórián keresztül érik el
 - Lekérdezések fogadása, végrehajtása, eredmény visszaküldése
 - Zár menedzser processz
 - Adatbázis író folyamat (módosult buffer blokkok folyamatos visszaírása a diszkre)
 - Naplózó processz
 - Checkpoint processz
 - Processz monitor processz (más processzek figyelés, helyreállítás)

Tranzakciós szerver II.



- Megosztott adatok a megosztott memóriában
- Zárkezelés kapcsán IPC túlterhelés ellen: zártábla közvetlen elérése
- Zármenedzser processz: pattdetekció



Adatszerver I.

- Tipikus, ha
 - a végrehajtandó műveletek számításigényesek
 - a kliens gépek teljesítménye összevethető a szerverével
 - a kliens és a szerver között nagysebességű adatkapcsolat biztosított.
- Utazik az adat a kliensre, az eredmény pedig vissza
- Kommunikáció költsége: msec vs. 100 nsec
- Fő kérdések: komm. granularitás, zárolás, adat caching, zár caching

Adatszerver II.

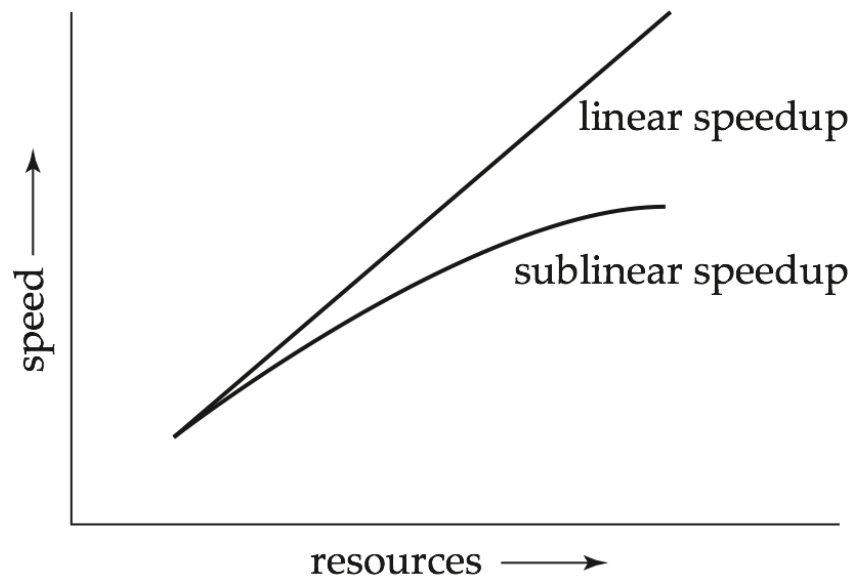
- Adatkommunikáció granularitása
 - Objektumonként költséges -> érdemes többet -> teljes lapok, kb. prefetch
- Zárolás
 - Szerver ítéli meg az átküldött objektumokra
 - Szükségtelen zárok
 - Zár visszahívható a szerver által, a kliens pedig visszaküldheti
 - *Zár de-eszkaláció*: a lapon lévő zárok deeszkálálhatók a lapon levő olyan objektumokhoz tartozó zárokra, amelyek ténylegesen szükségesek
- Adat cache
 - Kliens oldalon így más tranzakció is tudja az obj-t használni
 - *Cache koherencia* ellenőrzése használat előtt, ehhez üzenetcsere a szerverrel
- Zár cache
 - T.f.h. nagymértékben particionált adathasználat a kliensek között
 - A kliens a zárat is megtarthatja magánál két tr. között
 - Az obj. kommunikáció nélkül is használható más tranzakciókból
 - A szervernek követnie kell a zár cache-eket, szükség esetén mindet *visszahívni*
 - Hasonló, mint a zár deeszkaláció, csak tranzakciók között.

Párhuzamos DBMS I.

- Cél: növelni az
 - eredő I/O diszk sávszélességet
 - feldolgozási sebességettöbb diszk és CPU egyidejű használatával.
- Centralizált? Kliens-szerver?
- Durva és finom granularitású párhuzamos rendszerek
 - Durva: kevés, de egyenként nagyobb teljesítményű processzort működtetnek
 - Finom: akár több ezer (kisebb) processzossal dolgoznak
- Fő “performancia” mértékek: *tranzakciós teljesítmény, válaszidő*
- Miért jó? Ld. sok kis, ill. nagy tranzakciók esete
- Fő „jósági” mértékek: *speedup, scaleup*

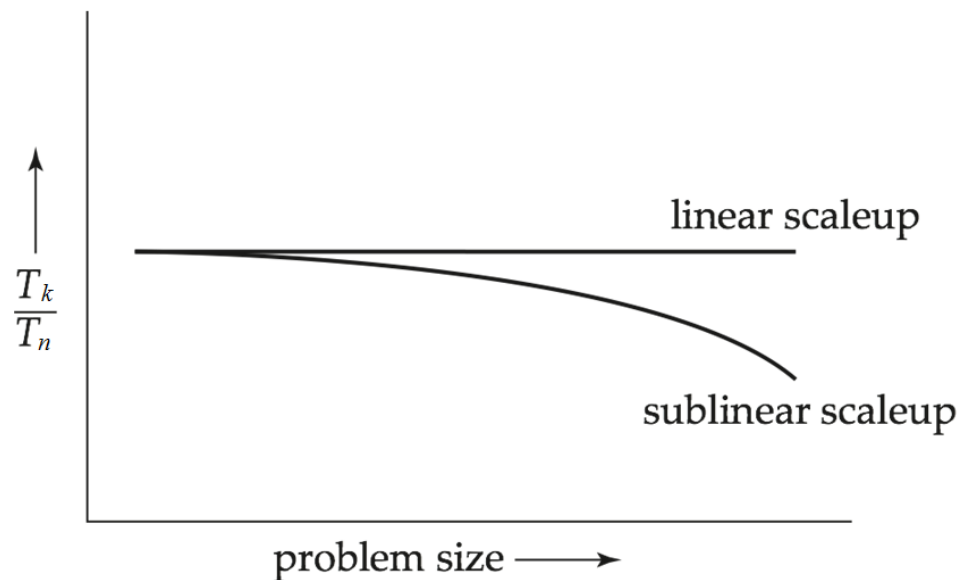
Párhuzamos DBMS II.

- Speedup (kb. gyorsítás)
 - Kiindulási futási idő T_k . Adjunk a rendszerhez N -szer több erőforrást, ezután a futási idő T_n .
Gyorsítás= T_k/T_n



Párhuzamos DBMS III.

- Scaleup (kb. skálázás)
 - Feladat méret: Q_k és Q_n , utóbbi „N-szer nagyobb”. T.f.h. Q_k végrehajtása egy adott M_k gépen T_k ideig tart, miközben a Q_n végrehajtása egy M_n gépen – ami N-szer nagyobb, mint M_k – T_n ideig tart. Skálázás= T_k/T_n .





Párhuzamos DBMS IV.

- Batch és tranzakciós skálázás
 - Batch: az adatbázis mérete nő, a taskok futási ideje az adatbázis méretével arányosan nő. Jellegzetes: full table scan esetén (DSS, szimulációk)
 - Tranzakciós skálázás: egyre nagyobb gyakoriságú tranzakciók, az adatbázis mérete a tranzakciósebességgel arányos. Jellegzetes: banki környezet.
- A skálázhatóság általában fontosabb, mint a gyorsítás. (Miért?)
- A lineárisan skálázódó rendszer teljesítménye alacsonyabb lehet...



Párhuzamos DBMS V.

Hatékonyságot csökkentő tényezők

- Processz indítási költségek
- Interferencia: megosztott erőforrásokért való versengés következtében
- Aszimmetria: A teljes futási időt a leglassabb rész futási ideje fogja meghatározni, miközben a darabolás nemigen tud egyforma lenni.

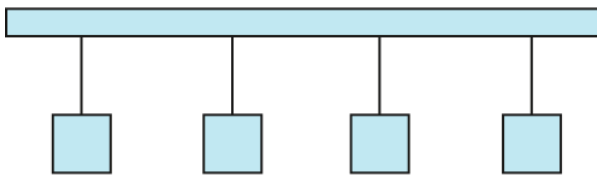


Párhuzamos DBMS VI.

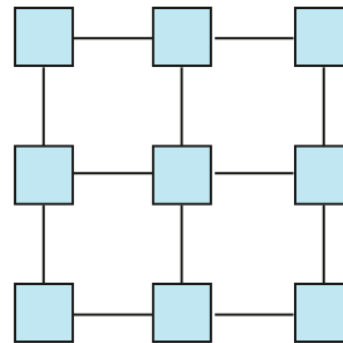
Összekötő hálózat (*interconnect*)

- busz: minden komponens minden üzenetet a közös buszra küld, és onnan kap.
 - Ethernet vagy párhuzamos kapcsolatot megvalósító belső összeköttetés.
 - Kevés processzor/komponens esetén.
- hálószerű: a komponensek egy rács csomópontjaiban vannak elhelyezve.
 - Közvetett kommunikáció routingon keresztül
 - A kommunikációs linkek száma nő a komponensek növekedésével
 - növekvő párhuzamosság esetén jobban skálázódik
- hiperkocka: a komponenseket binárisan számozva, a komponensek egymáshoz kapcsolva, ha számuk pontosan egy bitben különbözik.
 - a kommunikációs késleltetés lényegesen kisebb, mint hálószerű esetben

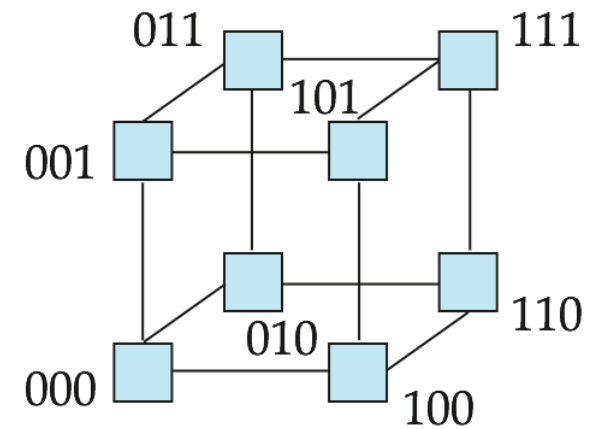
Párhuzamos DBMS VII.



(a) bus



(b) mesh

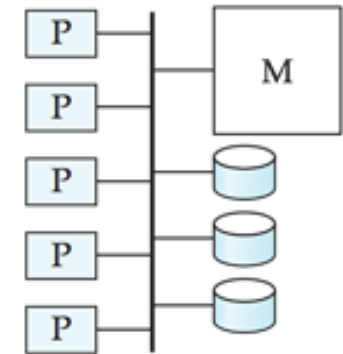


(c) hypercube

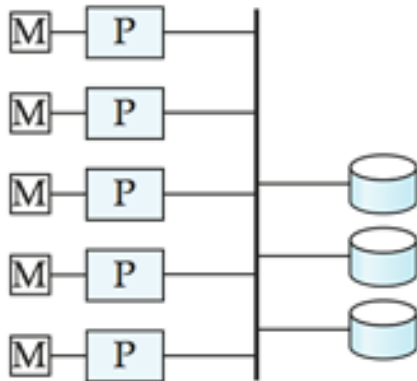
Párhuzamos DBMS VIII.

Párhuzamos architektúrák

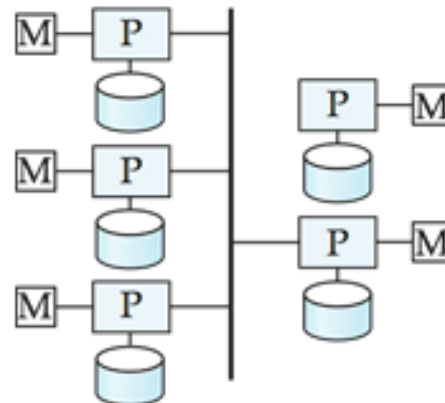
- megosztott memória („shared everything”)
- megosztott diszk (cluster, „fürt”)
- semmi nem megosztott („shared nothing”)
- hierarchikus: előbbieik hibridje.



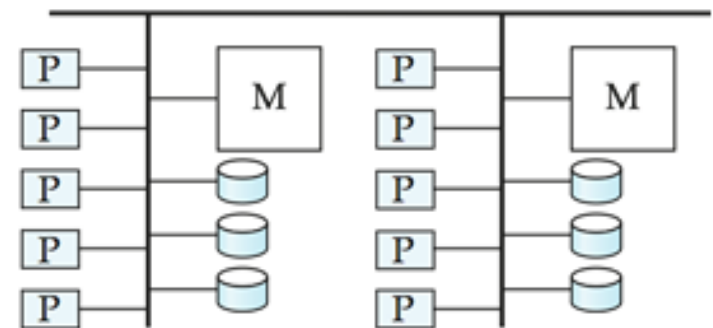
(a) shared memory



(b) shared disk



(c) shared nothing



(d) hierarchical

Párhuzamos DBMS IX.

Megosztott memória

- a CPU-k és a diszkek a közös memóriát egy buszon vagy egy interconnection hálózaton keresztül tudják elérni
- kommunikáció a processzorok között különösen hatékony – nem kell szoftveres adatmozgatás
- 32 proc. felett már nemigen skálázható a hálózat miatt
- Minden processzor lehetőleg nagy cache-t használ
- Elterjedt 4-8 processzor mellett
- Példa: Oracle a rapidon (Dell PowerEdge 2850 2x3,2 GHz CPU, 8G RAM, 1TB HDD)

Párhuzamos DBMS X.

Megosztott diszk/klaszter

- a processzoroknak saját memóriája van
- minden processzor minden diszkkal tud kommunikálni valamely interconnect hálózaton keresztül
- Memória busz nem szűk keresztmetszet
- Hibatűrés megvalósítása
- Hátrány:
 - interconnect a diszk alrendszer felé szűk keresztmetszet,
 - processzorok közötti kommunikáció lassúbb
- Példa: Oracle RAC (2001-)



Párhuzamos DBMS XI.

Semmi nem megosztott

- Csomópontokból áll (CPU, mem., diszk)
- Minden csp. szerver a saját adatai vonatkozásában
- Diszk I/O nagy része nem kerül az interconnect hálózatra, de:
 - Lekérdező parancsok
 - Nem-lokális diszkekhez való hozzáférések
 - Eredmények
- Interconnect is skálázhatóra készül -> kiváló eredő skálázhatóság (több ezer processzor)
- Hátrány:
 - processzorok közötti kommunikáció,
 - nemlokális diszkek elérése,mivel szoftver interakció kell az adatküldéshez mindkét végen.
- Példák: Oracle Exadata, Greenplum, Paraccel, Netezza, Teradata (egyik első), Vertica, Apache Cassandra, Apache Hbase, MongoDB, Riak, SAP HANA

Párhuzamos DBMS XII.

Hierarchikus

- kombinálja a korábbi három jellemzőit
- A felső szinten shared nothing, egy interconnect kapcsolja össze, nem osztanak meg diszket vagy memóriát
- A csp-ok lehetnek megosztott memóriájú vagy megosztott diszk
- Több hierarchiaszint is lehetséges
- Programozási komplexitás csökkentése: elosztott virtuális memória architektúrával (NUMA)



Elosztott DBMS I.

- Ismétlés, DDBMS, definíció, site/node...
- Vö: párhuzamos architektúrák
- A DDBMS csp-jai:
 - általában földrajzilag is szeparáltak,
 - külön adminisztráltak,
 - a kapcsolatok lassúbbak (WAN), valamint
 - lokális és globális tranzakciók...

Elosztott DBMS II.

Fő előnyök

- Rendelkezésreállítás
- Adatmegosztás
- Autonómia

Hátrányok

- Nagyobb komplexitás (ld. Adatb algoritmusai...)
- Nagyobb feldolgozási overhead

Elosztott DBMS III.

- Homogén
 - Minimum: azonos adatmodellt implementáló DBMS-ek
 - Közös, globális séma
 - (tip: mind támogassa a 2PL-t, csak azonos gyártó rendszerei, sőt azonos „típusú”)
 - Pl. Oracle, MySQL, SQL Server,...
- Heterogén (multiadatbázis, federated DB, szövetséges ~)
 - Különböző DBMS-ek
 - Közös séma különböző csp-ok között (multiadatbázis réteg)
 - Globális tranzakciók nehézkesek
 - Létező rendszerek összekapcsolására
 - Kevésbé elterjedt
 - Pl. IBM DataJoiner (relációs, 2001-), DataBroker (OODB)



Példa I: Oracle Exadata

Shared nothing/MPP DBMS

- Elemei:
 - Database servers, (SPARC, Oracle Linux)
 - Oracle Exadata Storage Servers,
 - InfiniBand storage networking (40 Gbit/sec).
- Kiegészítve:
 - Adatbázis tömörítés, titkosítás
 - Felhő szolgáltatás (2015-től az Oracle Cloud-ban előfizetéssel is elérhető)
- Változatos kiépítés (1 rack RAM: 7 TB-ig, CPU: 216 magig, HDD: 500 TB-ig, 97 GB/sec-ig)
- 2008-tól, kb. évente frissítik, legutóbbi 2016 (6. generáció)



Példa II: Facebook

- Külön előadás...😊



Irodalom

- Silberschatz, Korth, Sudarshan: Database System Concepts, 6th Ed. 2010.
- A Technical Overview of the Oracle Exadata Database Machine and Exadata Storage Server, Oracle 2012