



# Elosztott adatfeldolgozás

Bujáki Attila  
Bujaki.Attila@gmail.com



Adatbázisok haladóknak 2012.

2012. Október 9.



# Miről lesz szó?

Elosztottságról általában

Adatreplikáció, Georeplikáció általánosan

Georeplikáció a nagyméretű felhő alkalmazásokban

- Nehézségek
- Adat konzisztencia modellek
- Replikációs stratégiák
- Konkrét implementációk

*Kérdések, válaszok*



# Elosztottság - DDBMS

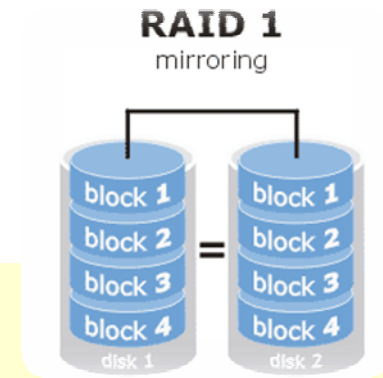
- Elosztottság módozatai
  - Felbontás (Fragmentation)
    - Vízszintes
    - Függőleges
    - Kevert
  - Replikáció
    - Nem teljes
    - Teljes replikáció
    - Teljes redundancia
  - Lehet kombinálni.





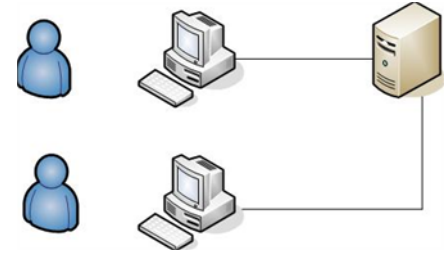
# Adatreplikáció

- Létezik cél HW, cél SW
  - Biztonsági mentések, archiválás, de most DB
- Georeplikáció
  - Nagy távolságok
  - Koordináció jelentős költséget jelent
  - Kritériumok enyhítése?
  - Magában is nagy üzlet (iOra, Syntergy)
    - Hajó-part replikáció
    - Katonai célok





# Konkurens hozzáférésvezérlés



- Soros végrehajtás – jól átlátható
- Konkurens – nem feltétlen
- Ne legyen konkurencia?
- Valamifajta sorrendben...
  - Adat konzisztencia modellek
  - Meghatározzák melyek az elfogadható kimenetek
- Az erős konzisztencia túl költséges a georeplikált rendszerekben. ☹



# Replikáció menedzsment

- A replikáció több célt szolgál
  - Terhelés elosztás, Megbízhatóság növelése
  - Elérési idők csökkennek
  - Hibatűrés biztosítása (akár komoly katasztrófáknál)
- Konzisztencia fenntartása?
  - Látszólag egy példány?
  - Georeplikált környezetben enyhébb kell. ☹
    - Nehezebb programozni
  - CAP



# Tanulság



- Replikáció előnyös
- Konkurens hozzáférés szabályozás előnyös
- Földrajzilag elosztott rendszereknél használjuk mind a kettőt – erős érvek.
- Georeplikáció
  - Magas késleltetési idők
- Hálózat szétszakadása
  - Lehetetlenné teszi az erős konzisztenciát
  - Enyhébbnél viszont nehezebb programozni
- Konzisztencia helyett teljesítmény



# Adat konzisztencia modellek - Feltevések

- Feltevések:
  - Csak regiszterek (Read+Write művelet)
  - Nincs pl. set (kommutatív szemantika)
- Konfliktusok és feloldásuk
  - Last writer wins rule: „Az ír igazán, aki utoljára ír!”
    - Regiszterre ok – de komplexebb objektumokra?
    - Pl.:set kommutatív hozzáadás, szemantikát is figyelni kell
    - Léteznek műveleti szemantikát is figyelembe vevő sémák
      - Set-nél: Add + Add - Merge



# Adat konzisztencia modellek - Típusok

- Adat konzisztencia modellek
  - Nem tranzakciós
  - Tranzakciós
- Regiszterek a megosztott adatelemek



# Adat konzisztencia modellek - Típusok

- Adat konzisztencia modellek
  - Nem tranzakciós
    - Linearizálhatóság
    - Szekvenciális konzisztencia
    - Rekordonkénti idővonal konzisztencia
    - Kauzális konzisztencia
    - Esetleges konzisztencia
    - Kauzális+ konzisztencia
  - Tranzakciós

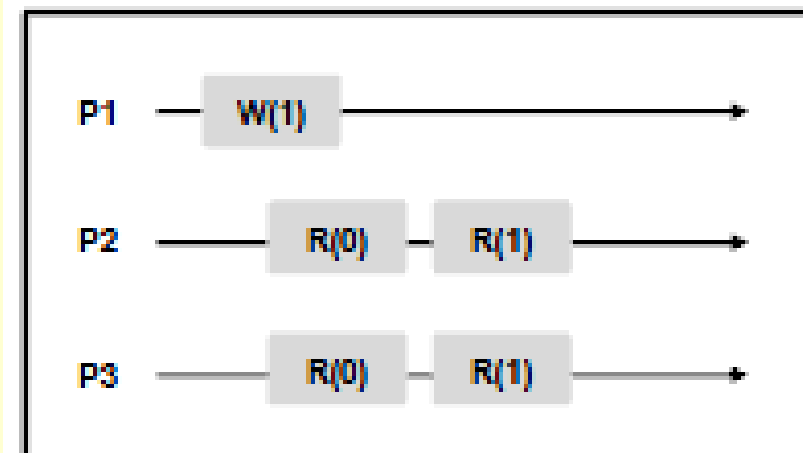


# Adat konzisztencia modellek

Nem tranzakciós:

- **Linearizálhatóság** (Atomi konzisztencia modell)

- Legközelebbi az ideálshoz
- Művelet hatása azonnali minden csúcsban
- Kompózábilítás
- Elosztott nagyon költséges
- Írás után egyből látszik



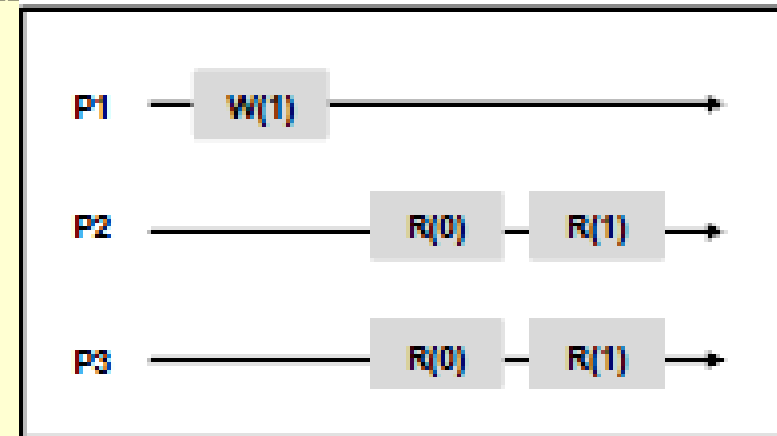


# Adat konzisztencia modellek

Nem tranzakciós:

## •Szekvenciális konzisztencia

- Enyhébb mint a linearizálhatóság
- Nem kompózabilis
- Írás nem kell, hogy egyből mindenhol látszódjon
- Ábra:
  - Írás után régi érték
  - Később már az új





# Adat konzisztencia modellek

Nem tranzakciós:

## • **Rekordonkénti idővonal konzisztencia**

- A szekvenciális enyhítése
- A replikációt felfedi a fejlesztőnek
- Objektumként biztosítja a szekvenciális konzisztenciát
- Objektumonkénti idővonal
- Visszalépés az idővonalon
- Egy adatház, egy objektuma

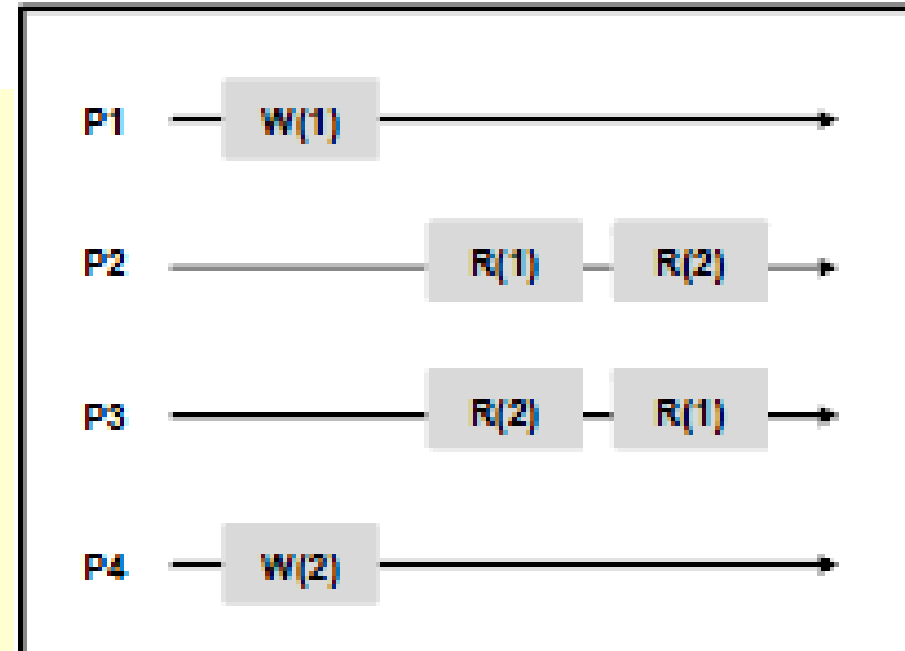


# Adat konzisztencia modellek

Nem tranzakciós:

## •Kauzális konzisztencia

- Az korrekt végrehajtások eredményei mindig konzisztensek a „korábban történt” részleges rendezéssel.
- „Korábban történt” – részleges rendezés - Lamport
- Különböző csúcsokban különböző sorrendben láthatjuk a konkurens írásokat megjelenni



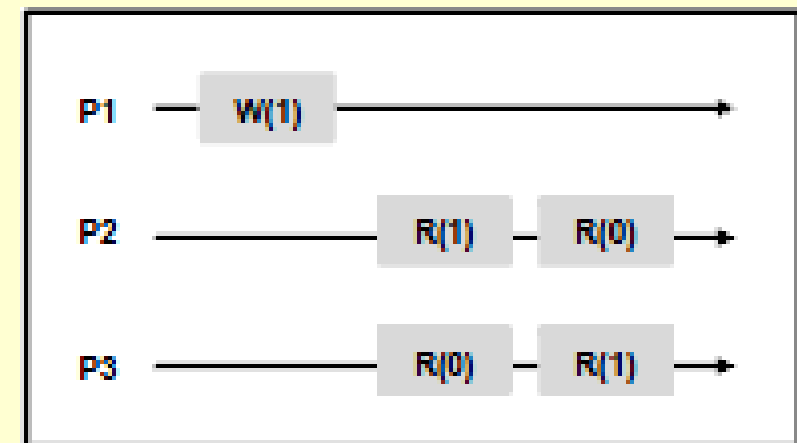
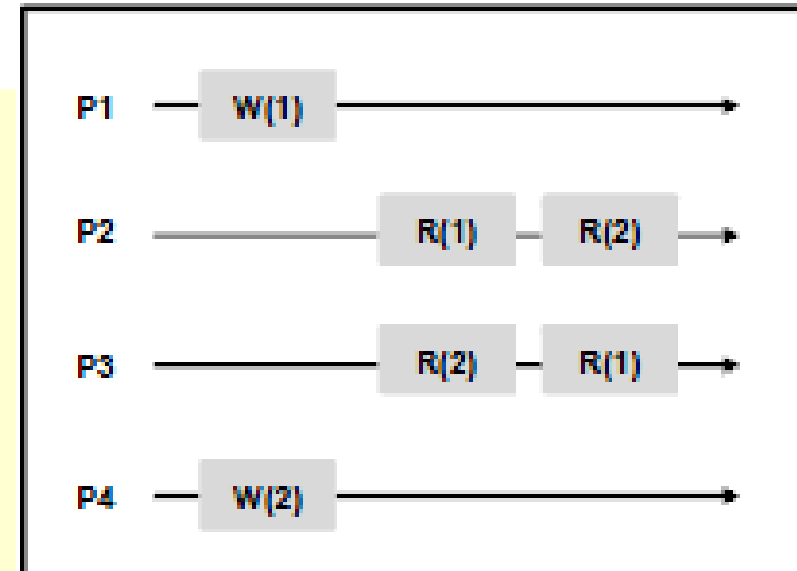


# Adat konzisztencia modellek

Nem tranzakciós:

- **Esetleges konzisztencia**

- Összefoglaló fogalom
- Inkonzisztencia ablak





# Adat konzisztencia modellek

Nem tranzakciós:

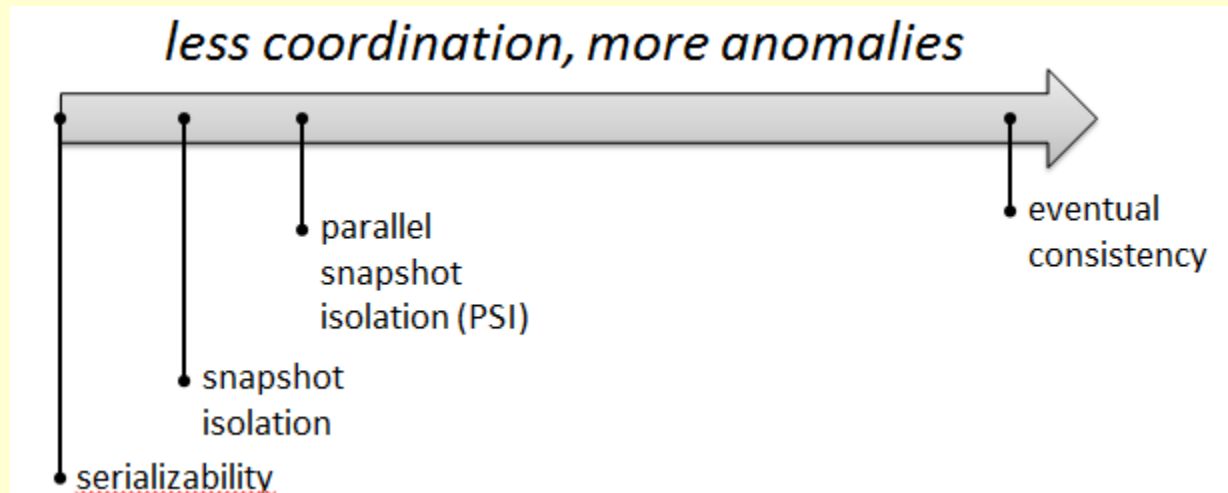
## •Kauzális+ konzisztencia

- A korábban történt rendezés miatt
  - Konkurens írások különböző sorrendben hathatnak a különböző csúcsokon
  - A replikák akár örökké divergálhatnak
  - Konvergens konfliktus kezelés
  - Konvergencia tulajdonság biztosítása – pl.: Last writer wins
  - Szemantikus feloldást is lehetővé tesz



# Adat konzisztencia modellek - Típusok

- Adat konzisztencia modellek
  - Nem tranzakciós
  - **Tranzakciós**
    - Sorosíthatóság
    - Snapshot izoláció
    - Parallel Snapshot Izoláció





# Adat konzisztencia modellek

Tranzakciós:

## •Sorosíthatóság

- Sok szó volt róla
- CAP
  - Georeplikált rendszerben egyszerűen nem megy
  - Túl költséges.



# Adat konzisztencia modellek

Tranzakciós:

## •Snapshot Isolation

- Enyhébb modell
- Elterjedten (Oracle, SQLServer)
- Adott időpontban összes korábban Commit-ált tranzakció hatása látszik.
- A tranzakciók mindig érvényes állapoton dolgoznak
- Nem függünk „félkész” tranzakcióktól
- Itt nem tükrözi a rendszer az időbeli **TELJES** rendezést – nem minden korábbi látszik.



# Adat konzisztencia modellek

Tranzakciós:

## •Parallel Snapshot Isolation

- SI Enyhítés georeplikált rendszereket célozva
- Egy csúcson belül Snapshot Isolation
- Csúcsok közt enyhébb
  - Csúcsonként külön snapshot



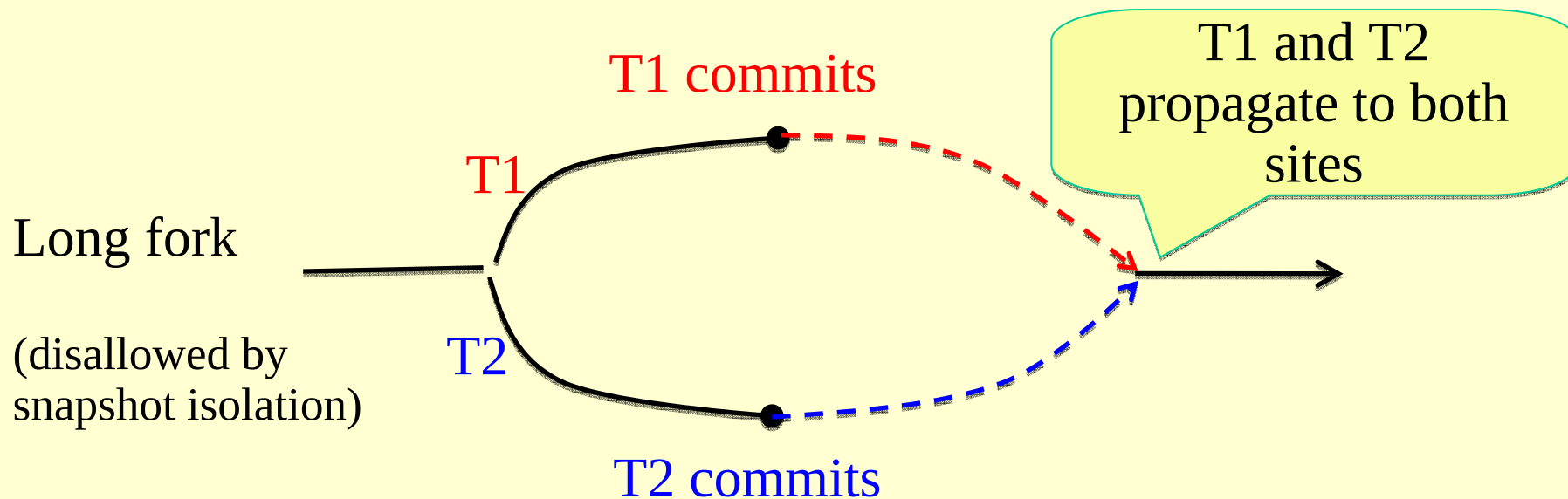
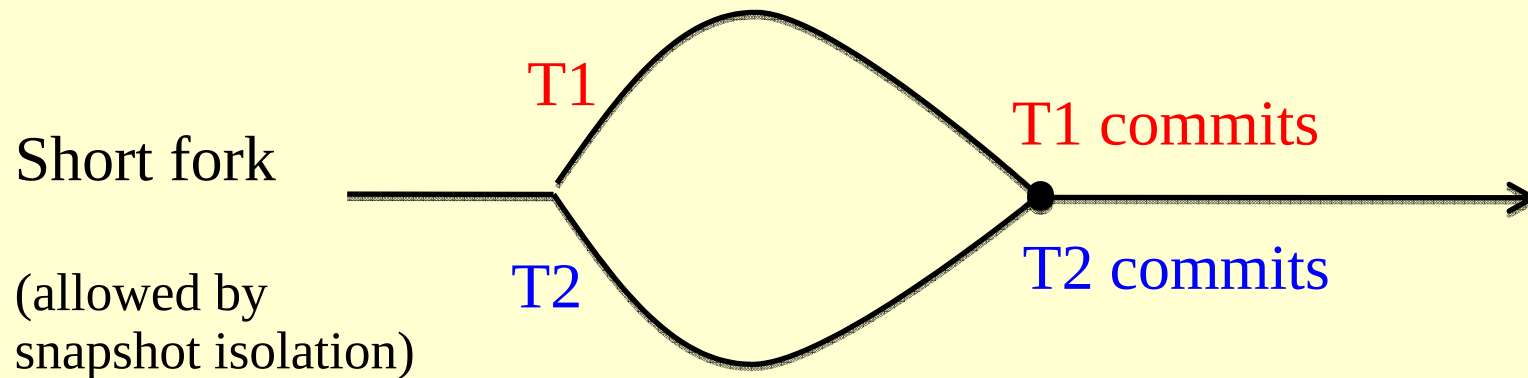
# Parallel Snapshot Isolation - Anomáliák

| Anomaly             | Serializability | Snapshot Isolation | PSI | Eventual |
|---------------------|-----------------|--------------------|-----|----------|
| dirty read          | No              | No                 | No  | Yes      |
| non-repeatable read | No              | No                 | No  | Yes      |
| lost update         | No              | No                 | No  | Yes      |

|                  |    |     |     |     |
|------------------|----|-----|-----|-----|
| short fork       | No | Yes | Yes | Yes |
| long fork        | No | No  | Yes | Yes |
| conflicting fork | No | No  | No  | Yes |



# Parallel Snapshot Isolation - Fork



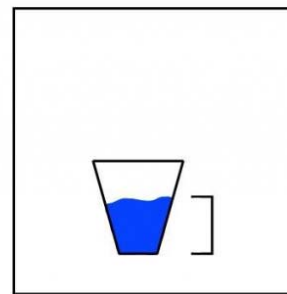


# Konkurens hozzáférés kezelő mechanizmusok

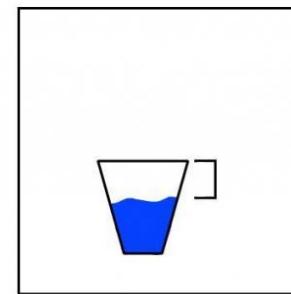
- **Lehetőségek**
  - Pesszimista vs. Optimista
- Pesszimista
- Optimista
- Valóság:
  - Attól függ.



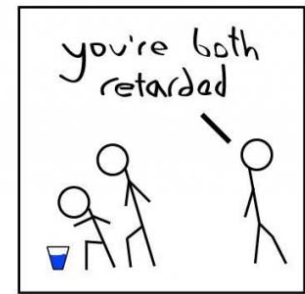
## Optimism vs pessimism



WHAT OPTIMISTS SEE



WHAT PESSIMISTS SEE



WHAT REALISTS SEE



# Adat replikációs stratégiák



- **Aktív replikáció**
  - Determinált, szinkronizált végrehajtás
  - Atomi üzenetszórás kell hozzá
    - Azonos sorrendben, mindenki megkapja
- Elsődleges példány replikáció
- Multi-master replikáció



# Adat replikációs stratégiák



- Aktív replikáció
- **Elsődleges példány replikáció**
  - Primary-backup replication
  - Egyetlen replikát módosítunk mindig (elsődleges példány) - AdatB
  - Lazy vs. Eager
  - Bottleneck
- Multi-master replikáció



# Adat replikációs stratégiák



- Aktív replikáció
- Elsődleges példány replikáció
- **Multi-master replikáció**
  - Több adatpéldány módosítható
  - Több szinkronizáció és koordináció szükséges



# Létező rendszerek

- A leírt modellek konkrét implementációi
  - Cluster of Order-Preserving Servers (COPS)
    - Causal+ (függőségekkel)
    - Princeton Uni., Intel, Carnegie Mellon Uni.
    - Jól skálázható „wide-area” elosztott méretekben is
    - Műveletek függőségei
    - Első Causal+, ami georeplikált szinten is jól skálázódik
    - Get-tranzakciók
      - Pl.



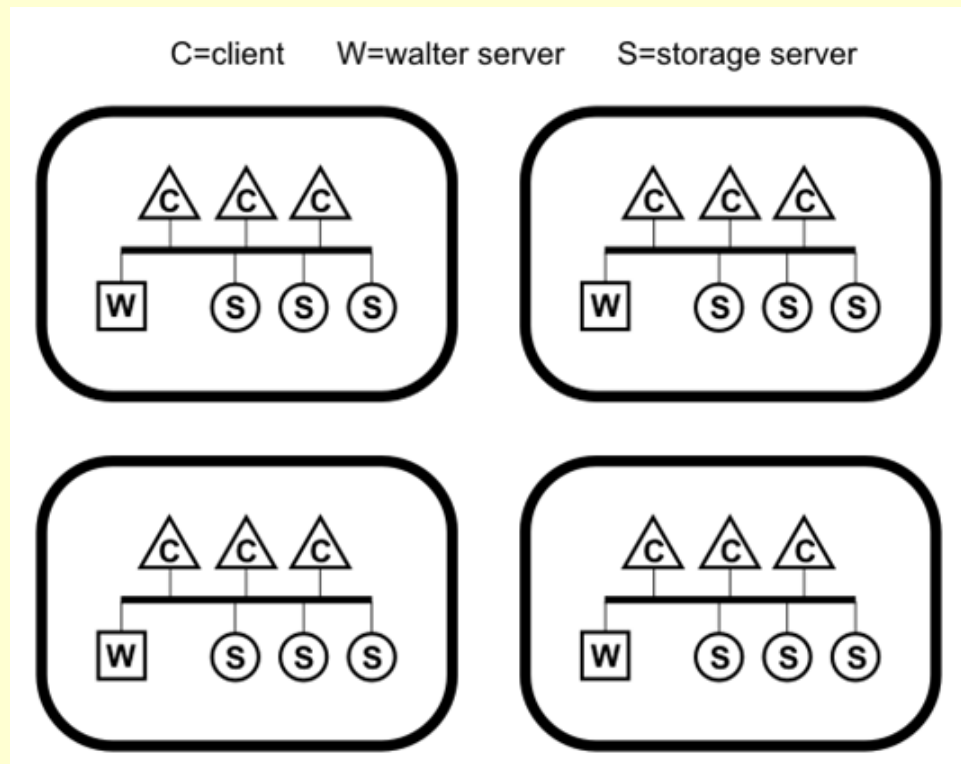
# Létező rendszerek

- A leírt modellek konkrét implementációi
  - Walter
    - Friss – georeplikált környezetre
    - Tranzakciók – hatásai asszinkron módon terjednek (PSI)
    - Key-value store (Egyszerű API)
    - Parallel Snapshot Isolation
      - Timestamp vector (Tranzakciók elején, külön minden csúcshoz)
      - Preferred site
      - Fast commit vs. Slow commit
      - Counting set - Fast commit (kommutatív műveletekre, am. nem)
    - Problémák: Site center (+megoldás), Slow commit ☹



# Létező rendszerek

- A leírt modellek konkrét implementációi
  - Walter (folytatás)





# Létező rendszerek



- A leírt modellek konkrét implementációi
  - Yahoo – PNUTS
    - Rekordonkénti idővonal konzisztencia
    - Rekord alapú – de nincsenek tranzakciók ☹
    - Minden rekordhoz elsődleges példány
    - Elsődleges példány adaptív kiválasztása
    - PNUTS API – konzisztencia modell választás, nem szép
    - Hálózat szétesését nem tudja megfelelően tolerálni
      - Elsődleges nem érhető el ezért?
        - Új elsődleges választása (két elsődleges – nem OK) vagy
        - Nincs frissítés ahol nem érhető el az elsődleges



# Létező rendszerek

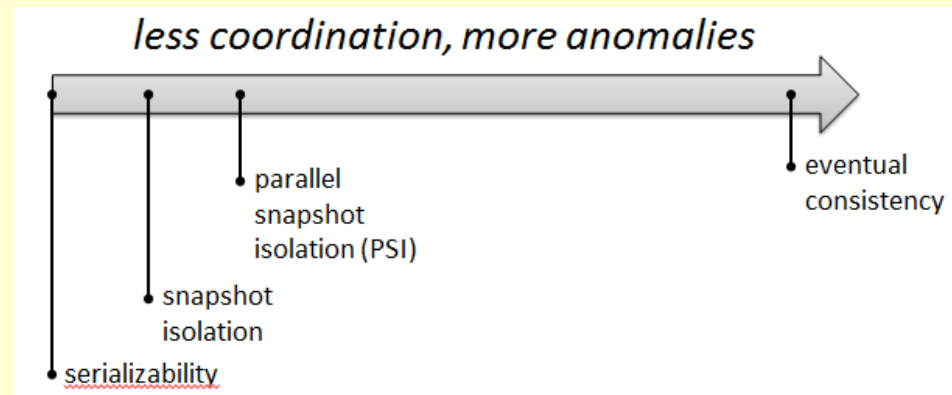


- A leírt modellek konkrét implementációi
  - Amazon – Dynamo
    - Key-value store
    - Elosztott Hash-tábla
    - Esetleges konzisztencia
      - Hibák vagy magas terhelés esetén
      - Ilyenkor szavazásos írás, szavazásos olvasás (konfigurálható)
      - DE:  $R\# + W\# > \text{Replikák száma}$
    - Verziózás - Írásnál új verzió, „vektor óra”
    - Ez kivételesen az írás megbízhatóságára tervezték
      - Írás-Írás konfliktus lehet – Automatikus kezelés vagy alkalmazás



# Létező rendszerek

- Összefoglalás és összehasonlítás



| Systems           | Consistency Model   | Concurrency Control |             | Replication Approach |                |              |
|-------------------|---------------------|---------------------|-------------|----------------------|----------------|--------------|
|                   |                     | Optimistic          | Pessimistic | Active               | Primary-Backup | Multi-Master |
| Chain Replication | Linearizability     |                     | ✓           |                      | ✓ (eager)      |              |
| COPS              | Causal+             | ✓                   |             |                      |                | ✓ (lazy)     |
| Walter            | PSI                 | ✓                   |             |                      | ✓ (lazy)       |              |
| PNUTS             | Per-record timeline |                     | ✓           |                      | ✓ (lazy)       |              |
| Lazy Replication  | Causal              | ✓                   |             |                      |                | ✓ (lazy)     |
| Dynamo            | Eventual            | ✓                   |             | ✓                    |                |              |



# Kérdések...

- ...válaszok (?)



# Köszönöm a figyelmet!

Marton József Ernő  
Bujáki.Attila@gmail.com



Adatbázisok haladóknak 2012.

2012. Október 9.