Distributed Database

Munawar, PhD

REPLICATION

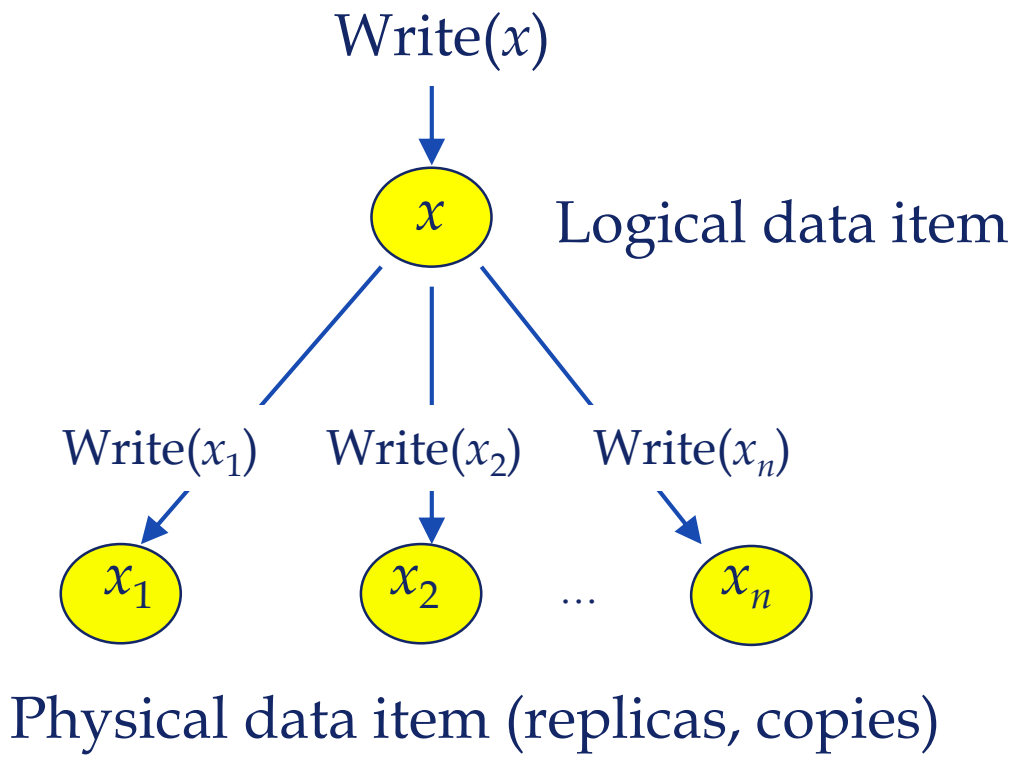
Replication

- Why replicate?
  - System availability
    - Avoid single points of failure
  - Performance
    - Localization
  - Scalability
    - Scalability in numbers and geographic area
  - Application requirements
- Why not replicate?
  - Replication transparency
  - Consistency issues
    - Updates are costly
    - Availability may suffer if not careful

Execution Model (1)

- There are physical copies of logical objects in the system.
- Operations are specified on logical objects, but translated to operate on physical objects.
- One-copy equivalence
  - The effect of transactions performed by clients on replicated objects should be the same as if they had been performed on a single set of objects.

Execution Model (2)



Replication Issues

- Consistency models - how do we reason about the consistency of the "global execution state"?
  - Mutual consistency
  - Transactional consistency
- Where are updates allowed?
  - Centralized
  - Distributed
- Update propagation techniques – how do we propagate updates to one copy to the other copies?
  - Eager
  - Lazy

Consistency

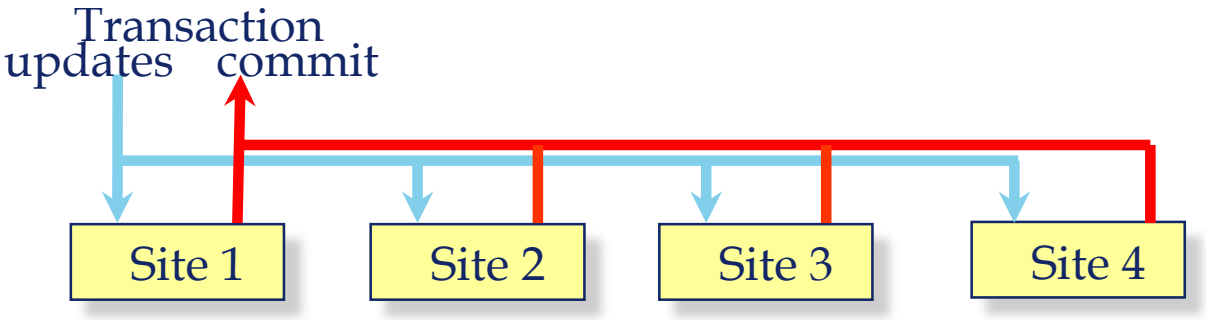
- Mutual Consistency
  - How do we keep the values of physical copies of a logical data item synchronized?
  - Strong consistency
    - All copies are updated within the context of the update transaction
    - When the update transaction completes, all copies have the same value
    - Typically achieved through 2PC
  - Weak consistency
    - Eventual consistency: the copies are not identical when update transaction completes, but they eventually converge to the same value
    - Many versions possible:
      - Time-bounds
      - Value-bounds
      - Drifts

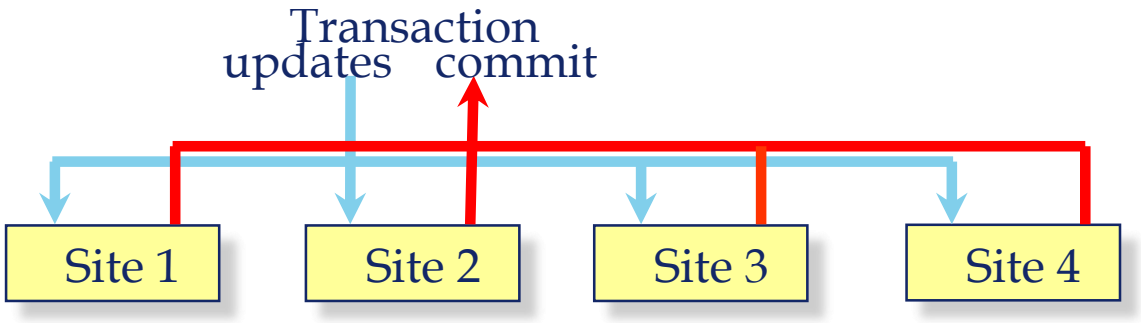
Transactional Consistency

- How can we guarantee that the global execution history over replicated data is serializable?
- One-copy serializability (1SR)
  - The effect of transactions performed by clients on replicated objects should be the same as if they had been performed *one at-a-time* on a single set of objects.
- Weaker forms are possible
  - Snapshot isolation
  - RC-serializability

Distributed

- Changes can be initiated at any of the copies. That is, any of the sites which owns a copy can update the value of the data item.





Forms of Replication

Eager

+ No inconsistencies (identical copies)
+ Reading the local copy yields the most up to date value
+ Changes are atomic
− A transaction has to update all sites
  − Longer execution time
  − Lower availability

Lazy

+ A transaction is always local (good response time)
− Data inconsistencies
− A local read does not always return the most up-to-date value
− Changes to all copies are not guaranteed
− Replication is not transparent

Centralized

+ No inter-site synchronization is necessary (it takes place at the master)
+ There is always one site which has all the updates
− The load at the master can be high
− Reading the local copy may not yield the most up-to-date value

Distributed

+ Any site can run a transaction
+ Load is evenly distributed
− Copies need to be synchronized

Replication Protocol

- The previous ideas can be combined into 4 different replication protocols:

	Centralized	Distributed
Eager	Eager centralized	Eager distributed
Lazy	Lazy centralized	Lazy distributed

Replication Strategies

	Centralized	Distributed
Eager	+ Updates do not need to be coordinated + No inconsistencies - Longest response time - Only useful with few updates - Local copies are can only be read	+ No inconsistencies + Elegant (symmetrical solution) - Long response times - Updates need to be coordinated
Lazy	+ No coordination necessary + Short response times - Local copies are not up to date - Inconsistencies	+ No centralized coordination + Shortest response times - Inconsistencies - Updates can be lost (reconciliation)

KEMAMPUAN AKHIR YANG DIHARAPKAN

Mahasiswa secara aktif, kreatif dan dapat ber-inovasi dalam menanyakan mengenai proses perkuliahan dalam mengerti dan memahami tentang Data Replication

Thank You !

Munawar, PhD