Distributed Systems (ICE 601)

Replication & Consistency - Part 3

Dongman Lee

ICU

Class Overview

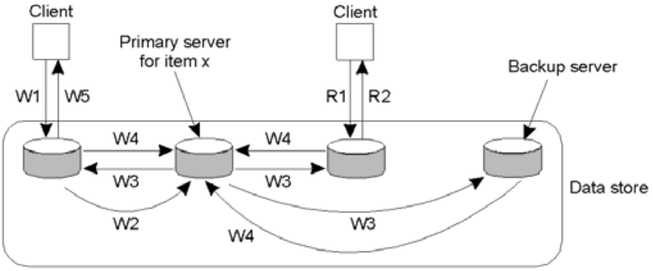
- Introduction
- Replication Model
- Request Ordering
- Consistency Models
- Consistency Protocols
- Case study
  - Transactions with Replicated Data
  - Lazy replication
  - ISIS

Consistency Protocols

- Description
  - describe an implementation of a specific consistency model
- Classification
  - primary-based protocols
    - remote-write protocols
    - local-write protocols
  - replicated-write protocols
    - active replication
    - quorum-based protocols

Primary-based Remote-Write Protocols

- All write operations are performed at a (remote) fixed server
  - read operations are allowed on a local copy while write operations are forwarded to a fixed primary copy



W1. Write request
W2. Forward request to primary
W3. Tell backups to update
W4. Acknowledge update
W5. Acknowledge write completed

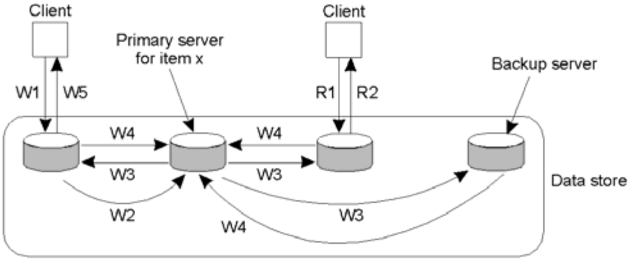
R1. Read request
R2. Response to read

Primary-based Remote-Write Protocols (cont.)

- Issues
  - update can be a performance bottleneck if implemented as a blocking operation
    - but guarantees sequential consistency (most recent write as the result of a read)
    - if implemented as a non-blocking, the protocol provides no guarantee of sequential consistency and fault tolerance

Primary-based Local-Write Protocols

- All write operations are performed locally and forwarded to the rest of replicas
  - primary copy migrates between processes that wish to perform a write operation
  - Multiple, *successive* writes can be done locally (via non-blocking protocol)
  - can be exploited in mobile computing

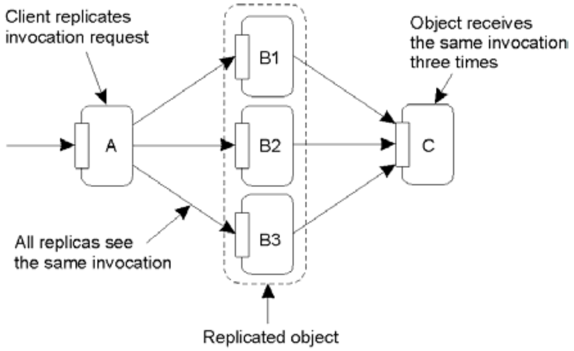


W1. Write request
W2. Forward request to primary
W3. Tell backups to update
W4. Acknowledge update
W5. Acknowledge write completed

R1. Read request
R2. Response to read

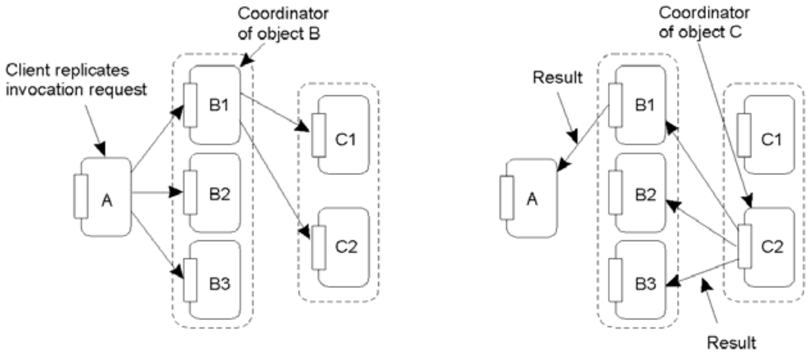
Active Replication

- Each replica performs update operations and propagates them (or the results) to the others
  - requires totally ordered multicast
- Replicated invocation problem


Client replicates invocation request
B1
A
B2
B3
C
Object receives the same invocation three times
All replicas see the same invocation
Replicated object

Active Replication (cont.)

- Solutions to the replicated invocation problem
  - group coordinator
  - sender-driven vs. receiver-driven

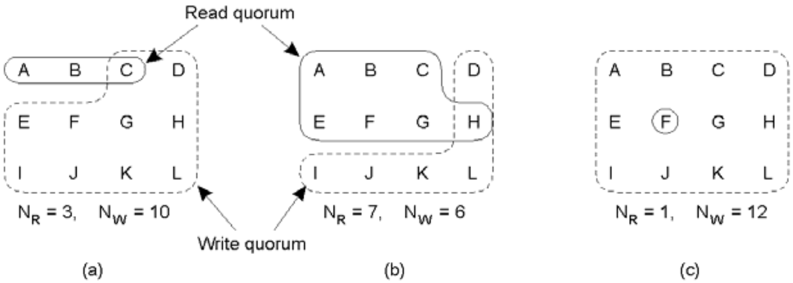


Quorum-based Protocols

- Require clients to request and acquire the permission of multiple servers before any operation on replicas
  - quorum set
    - W > half the total votes
    - R + W > total number of votes for group
      - any pair of read quorum and write quorum must contain common copies, so no conflicting operations on the same copy
    - read operations
      - check if there is enough number of copies >= R
      - perform operation on up-to-date copy
    - write operations
      - check if there is enough number of up-to-date copies >= W
      - perform operation on all replicas

Quorum-based Protocols (cont.)

- Examples



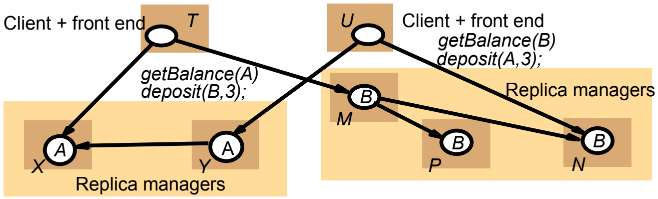
a) A correct choice of read and write set
b) A choice that may lead to write-write conflicts since W <= N/2
c) A correct choice, known as ROWA (read one, write all)

Transactions with Replicated Data

- Replicated transactions
  - transactions in which a physical copy of each logical data item is replicated at a group of servers (replicas)
- One-copy serializability
  - effects of transactions performed by various clients on replicated data items are the same as if they had been performed one at a time on single data item
  - to achieve this
    - concurrency control mechanisms are applied to all of replicas
    - 2PC protocol becomes two level nested 2PC protocol
      - phase 1
        - a worker forwards "ready" message to replicas and collects answers
      - phase 2
        - a worker forward "commit" message to replicas
  - primary copy replication: concurrency control is only applied to primary

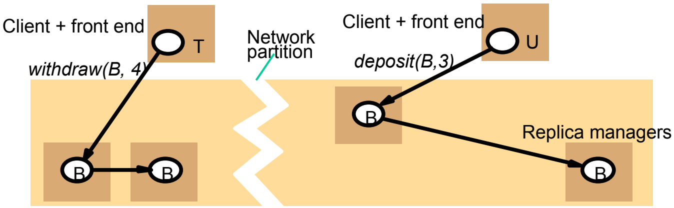
Transactions with Replicated Data (cont.)

- Available copies replication
  - designed to allow for some replicas being allowed unavailable
  - client's Read operation is performed on any of available copy but Write operation on all of available copies
  - failures and recoveries of replicas should be serialized to support one-copy serializability
  - ➔ local validation
    - a transaction checks for any failures (and recoveries) of replica managers of objects it has accessed before it commits



Transactions with Replicated Data (cont.)

- Network partition
  - can separate a group of replicas into subgroup between which communications are not possible
  - assume that partition will be repaired
  - resolutions
    - optimistic approach
      - available copies with validation
    - pessimistic approach
      - quorum consensus
      - virtual partition



Transactions with Replicated Data (cont.)

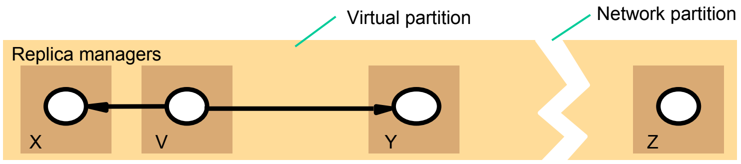
- Available copies with validation
  - available copies algorithm is applied to each partition
  - after partition is repaired, possibly conflicting transaction is validated
    - version vector can be used to check validity of separately committed data items
    - precedence graphs can be used to detect conflicts between Read and Write operations between partitions
    - only feasible with applications where compensation is allowed

Transactions with Replicated Data (cont.)

- Quorum consensus
  - operations are only allowed when a certain number of replicas (i.e. quorum) are available in the partition
    - possible only one partition can allow operations committed so as to prevent transactions in different partitions from producing inconsistent results
  - performed using Quorum-based protocol
- Virtual partition
  - combination of quorum consensus (to cope with partition) and available copies algorithm (inexpensive Read operation)
  - to support one-copy serializability, a transaction aborts if replica fails and virtual partition changes during progress of transaction
  - when a virtual partition is formed, all the replicas must be brought up to date by copying from other replicas

Transactions with Replicated Data (cont.)

- Virtual partition (cont.)
  - virtual partition creation
    - phase 1
      - initiator sends Join request to each potential replica with logical timestamp
      - each replica compares timestamp of current virtual partition
        - if proposed time stamp is greater than local one, reply yes
        - otherwise, no
    - phase 2
      - if initiator gets sufficient Yes replies to form read and write quora and send confirmation message with list of members
      - each member records timestamp and members


Virtual partition
Network partition
Replica managers
X
V
Y
Z