

Replication

March 16, 2001

Replication

What is Replication?

- A technique for increasing availability, fault tolerance and sometimes, performance
- Replication can occur at many levels:
 - Data
 - Computation
 - Communication
- Issues: Consistency and Coordination

2

Replication, cont.

Replication Examples

- Atomic writes via file logging
 - Performance hit - more than double the time to write
- DNS vs. Grapevine
 - Consistency limits performance and scaling
- RAID
 - Replication increases performance, but costs money
- AFS
 - Possibility that merge is not practical

3

Replication, cont.

Replication of Data in Transactions

- One-copy semantics
 - To a transaction, a set of replicated objects should behave as one
- Big Tradeoff - Performance under less than perfect conditions
 - Network failures
 - Server downtime

4

Replication Model

- Simple: Read-one/write all
 - Use locks across all replicas to ensure consistency
- Problem - availability
 - If a replica is down, its objects cannot be updated until it comes back up again
 - Worse - if the network is partitioned, a whole set of objects will be affected

5

Network Failures

- Failures Partition a set of replicas
- You have a variety of options to choose from:
 - Optimistic
 - Pessimistic
- Key metrics for choice:
 - Resource contention
 - Frequency and duration of network failures

6

Optimistic Approach

- Allow any replica to act as a proxy for the whole
- As partitions dissolve, merge updates
 - Use normal commit semantics if you can always merge safely
 - Alternative approach: manage inconsistencies outside of the transaction
 - e.g., ATMs, date book

7

Example - Available Copies Replication

- Each object that is replicated is managed by a replica manager
 - Any available replica manager can process requests
- Reads from any one replica OK, writes propagate to available replicas
 - Transactions can lock objects locally at replicas
 - Two-phase commit at replica level to ensure consistency among replicas
- Requires ability to detect server failures

8

Pessimistic Approach - Quorum

- One partition can be the proxy for the whole
- Other partitions must wait for network failure to be resolved
- Each object replica has a value + timestamp when the value was last updated
- To read or write, you must have a quorum:
 - Writes - more than half the votes
 - Reads - more than (total votes minus minimum needed to write)

9

Virtual Partition

- Combine Quorum with available copies replication
- Transaction acts in a virtual partition, artificially created
 - Virtual Partition is smallest quorum
- Assumption - Virtual Partition small enough to practically allow available copies to succeed

10

Putting it all together

- Transactions
 - Atomicity
 - Recoverability
- Implemented on top of:
 - Locks (mutual exclusion algorithms)
 - Non-volatile storage with atomic writes (logging)
- To achieve performance and availability
 - Distribution and Replication

➡ Transaction implementation come in many flavors 11