CS743 - Principles of Database Management and Use

Distribution, Replication, and CAP

Ken Salem

David R. Cheriton School of Computer Science
University of Waterloo

Fall 2014
Data Partitioning
Data Partitioning

![Diagram showing data partitioning in a distributed database system with multiple DBMS instances.]
Two Phase Commit (2PC)

1. UPDATE R
Two Phase Commit (2PC)

1. UPDATE R
2. UPDATE S

Strict 2PL at each site plus 2PC ensures global serializability.
Two Phase Commit (2PC)

1. UPDATE R
2. UPDATE S
3. UPDATE X
Two Phase Commit (2PC)

1. UPDATE R
2. UPDATE S
3. UPDATE X
4. COMMIT
   - 2PC phase 1
Two Phase Commit (2PC)

1. UPDATE R
2. UPDATE S
3. UPDATE X
4. COMMIT
   - 2PC phase 1
Two Phase Commit (2PC)

1. UPDATE R
2. UPDATE S
3. UPDATE X
4. COMMIT
   - 2PC phase 1
   - 2PC phase 2
Two Phase Commit (2PC)

1. **UPDATE R**
2. **UPDATE S**
3. **UPDATE X**
4. **COMMIT**

- 2PC phase 1
- 2PC phase 2

Strict 2PL at each site plus 2PC ensures **global serializability**.
Data Replication

DBMS
Data Replication

DBMS

DBMS DBMS DBMS
1-Copy Serializability (1SR)

- correctness criterion suitable for replicated databases
- system behaves as if there is a single copy of each object on which transactions appear to execute sequentially in some order
Eager Read One, Write All (ROWA) Replication

- to read $R$, read local replica of $R$
Eager Read One, Write All (ROWA) Replication

- to read $R$, read local replica of $R$
- to update $R$, update all replicas of $R$
Eager Read One, Write All (ROWA) Replication

- to read $R$, read local replica of $R$
- to update $R$, update all replicas of $R$
- each local site has a local concurrency controller
Eager Read One, Write All (ROWA) Replication

- to read $R$, read local replica of $R$
- to update $R$, update all replicas of $R$
- each local site has a local concurrency controller
- use 2PC to atomically commit transaction updates
Eager Read One, Write All (ROWA) Replication

- to read $R$, read local replica of $R$
- to update $R$, update all replicas of $R$
- each local site has a local concurrency controller
- use 2PC to atomically commit transaction updates

Global Serializability

Local strict two-phase locking + 2PC for commit coordination is sufficient to ensure global 1SR.
CAP

Consistency: serializability

Availability: nodes that are up should eventually respond to requests

Partition-Tolerance: system should remain consistent and available even if it partitions
Consistency: serializability
Availability: nodes that are up should eventually respond to requests
Partition-Tolerance: system should remain consistent and available even if it partitions

Brewer’s CAP Conjecture (PODC 2000)
It is impossible build a [distributed database] system that provides consistency, availability, and partition-tolerance.