# Replication and Consistency: Being Lazy Sometimes Helps

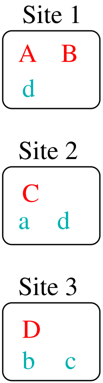
Presentation of a paper by Yuri Brietbart and Henry F. Korth from the 1997 ACM Symp. on Principles of Database Systems (PODS'97)

Ken Salem

David R. Cheriton School of Computer Science
University of Waterloo

September 27, 2006

# System Model

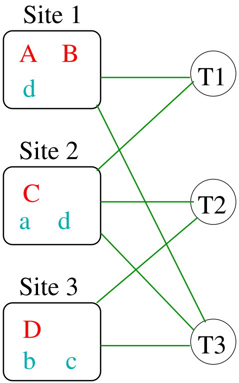



- lazy/master replication, each object has a designated master copy
- partial replication allowed
- no distributed transactions

**Goal**

Ensure global serializability.

# Simplified Global Serializability (GS) Protocol

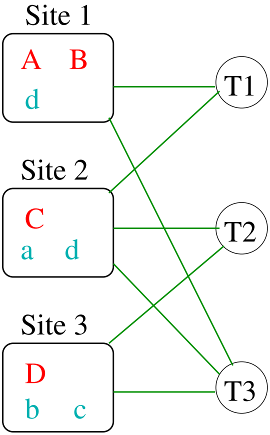



- $T_1$: read $B, d$, write $A$ (Site 1)
- later, update $a$ at Site 2
- $T_2$: read $C, a, d$, write $C$
- later, update $c$ at Site 3
- $T_3$: read $b, c$, write $D$
- later, update $d$ at Sites 1,2
- replication graph

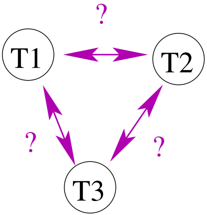
## Main Idea

Acyclic replication graph implies global serializability.

# GS Graphs and Serialization Graphs




replication graph



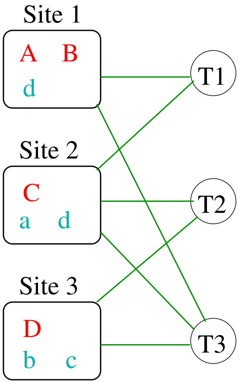

serialization graph

# Generalization #1: Dynamics

- GS protocol maintains replication graph dynamically
- when $T$ writes data, try to add required arcs to replication graph
  - if graph would be acyclic, $T$ proceeds
  - otherwise, $T$ aborts or is delayed
- Question: when can $T$ be removed from the replication graph?
- Answer: when all parts of $T$ are committed, and no other transactions in the graph precede $T$

Similar to condition for removing transactions from dynamically maintained serialization graphs.

# Generalization #2: False Conflicts

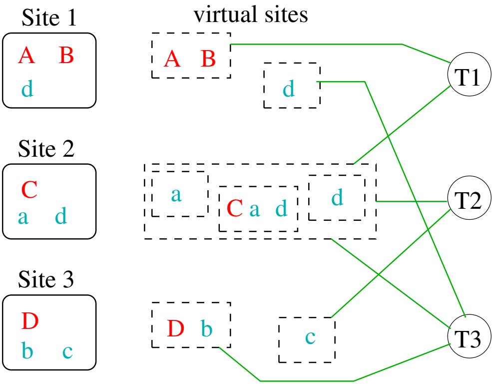



- $T_1$: read $B$,~~d~~, write $A$
- $T_2$: read $C, a, d$, write $C$
- $T_3$: read $b$,~~c~~, write $D$
- replication graph is same as original example, but these transactions can always be serialized.

## False Conflicts

Simplified protocol conservatively assumes that read/write conflicts exist, though they may not.

# Virtual Sites




- $T_1$: read $B$, ~~$d$~~, write $A$ (Site 1)
- $T_2$: read $C$, $a$, $d$, write $C$
- $T_3$: read $b$, ~~$c$~~, write $D$

Virtual sites reduce false conflicts.

# Comparisons to Other Papers

vs. Ganymed:

- centralized updates in Ganymed
- more complex and demanding global concurrency control in GS protocol
- serializability vs. SI

vs. Postgres-R:

- eager vs. lazy replication
- Postgres-R needs underlying group communication mechanism
- no local DBMS mods for GS

# Discussion

- demands on local DBMS
  - expose serialization order (for managing replication graph)
  - expose transaction readset and writeset (for managing virtual sites)
- performance issues
  - distributed deadlocks are possible
  - even purely local transactions experience overhead, aborts
  - even local read-only transactions experience overhead, aborts
- similar protocol for SI? Would it perform better?
- physical design problem
  - given workload description and a set of sites, decide where to place primary copies and replicas
  - constrained optimization problem
  - objectives: balance load, minimize aborts/delays