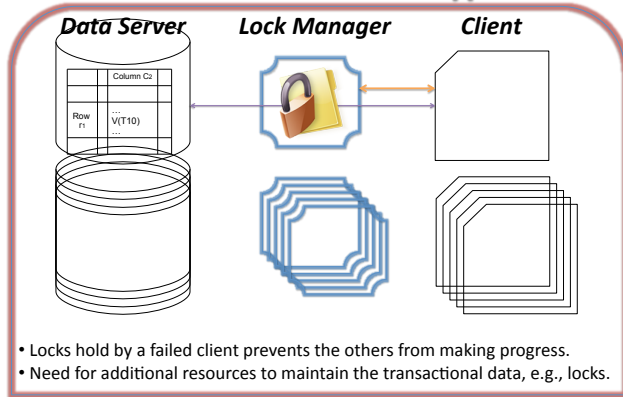
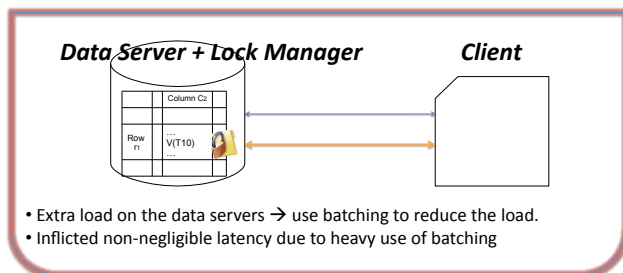


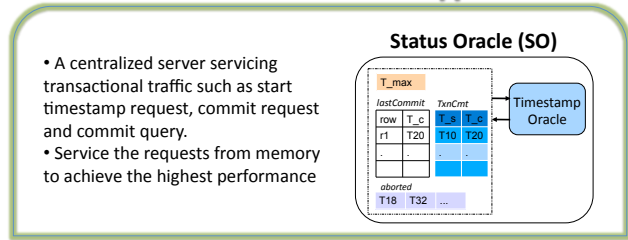
Lock-based Transactional Support



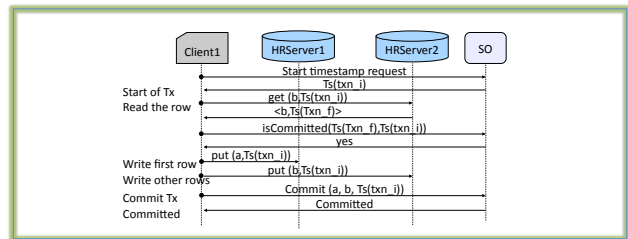
Google Percolator



Lock-free Transactional Support



Sequence Diagram

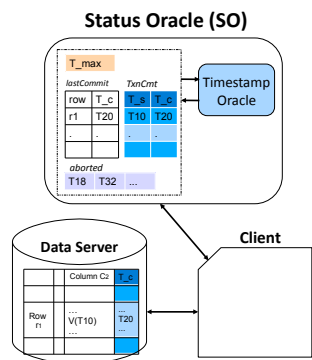


Challenges & Solutions

- Partial data in memory → remember T_{max} , the maximum removed timestamp from memory
- Reliability of in-memory data → Write-ahead logging via BookKeeper
- Limited CPU power → read-only replicas of SO at data servers or clients

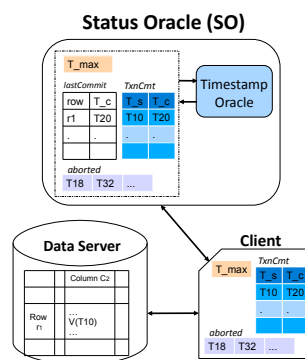
SO Data Replication

Replication on Data Servers



By maintaining a read-only copy of SO data on some other nodes, some part of transactional traffic (queries about commit timestamps) could be answered by the read-only replicas, reducing the load on processing load on the SO. The read-only replicas could be stored in some new servers, the same data servers, or the clients. Our approach lightly replicates the SO data on clients.

Replication on Clients



- Piggyback Δ SO on Start Timestamp Response message
- No modification into data server → run on top of any key-value store
- Close-to-zero overhead on data servers
- A prototype is implemented on top of Hbase
- Scalable up to
- 50,000 write transactions per second
- 1000 client connections

