Scalable Replication and Data Consistency

Kai Shen

Dept. of Computer Science, University of Rochester

Scalable Replication and Data Consistency

- Replication improves data availability and durability
- Replication introduces consistency concern
  - Basic consistency: all replica agrees on the same data at any stabilization point
  - Strong consistency: results of reads are as if all operations (reads and writes) are serially ordered
- In database, to achieve strong replication consistency:
  - Two-phase commits (writes become barriers)
  - Strong primary-secondary replication (writes initiated at the primary, reads replied at the primary)
- Both sacrifice performance or scalability

Scalable Distributed Data Structures [Gribble et al. OSDI2000]

- Two-phase commits do not necessarily sacrifice performance/scalability.
- Abstract data management (scalability/availability/consistency) into data structures.
- If the data access semantics is simple enough – in case of hash tables
  - conflicts are rare on fine-granularity data units, so a two-phase commit won't hold up two many reads.

Commutable Writes [Saito et al. SOSP1999]

- If all writes are commutable, then they can be executed at different replicas in any order.
- Assumption: there is a global total order of all writes
- Appends
- All total object overwrites
  - a write proceeds if it follows all committed writes on the object
  - a write is ignored if it precedes any already committed write on the object
- Only satisfies the basic consistency, not strong consistency

Google File System
[Ghemawat et al. SOSP2003]

- Primary-secondary replication
  - writes initiated at the primary
  - reads replied anywhere
- Only satisfies the basic consistency, not strong consistency

- Interesting features:
  - separate control from data flow
  - data flow in fixed order

Chain Replication
[van Renesse and Schneider OSDI2004]

- All replicas organized in a chain:
  - writes go to the head, and then flow through the chain
  - reads services at the tail
- Satisfy strong consistency

- Simple failure management
- Always add at the tail