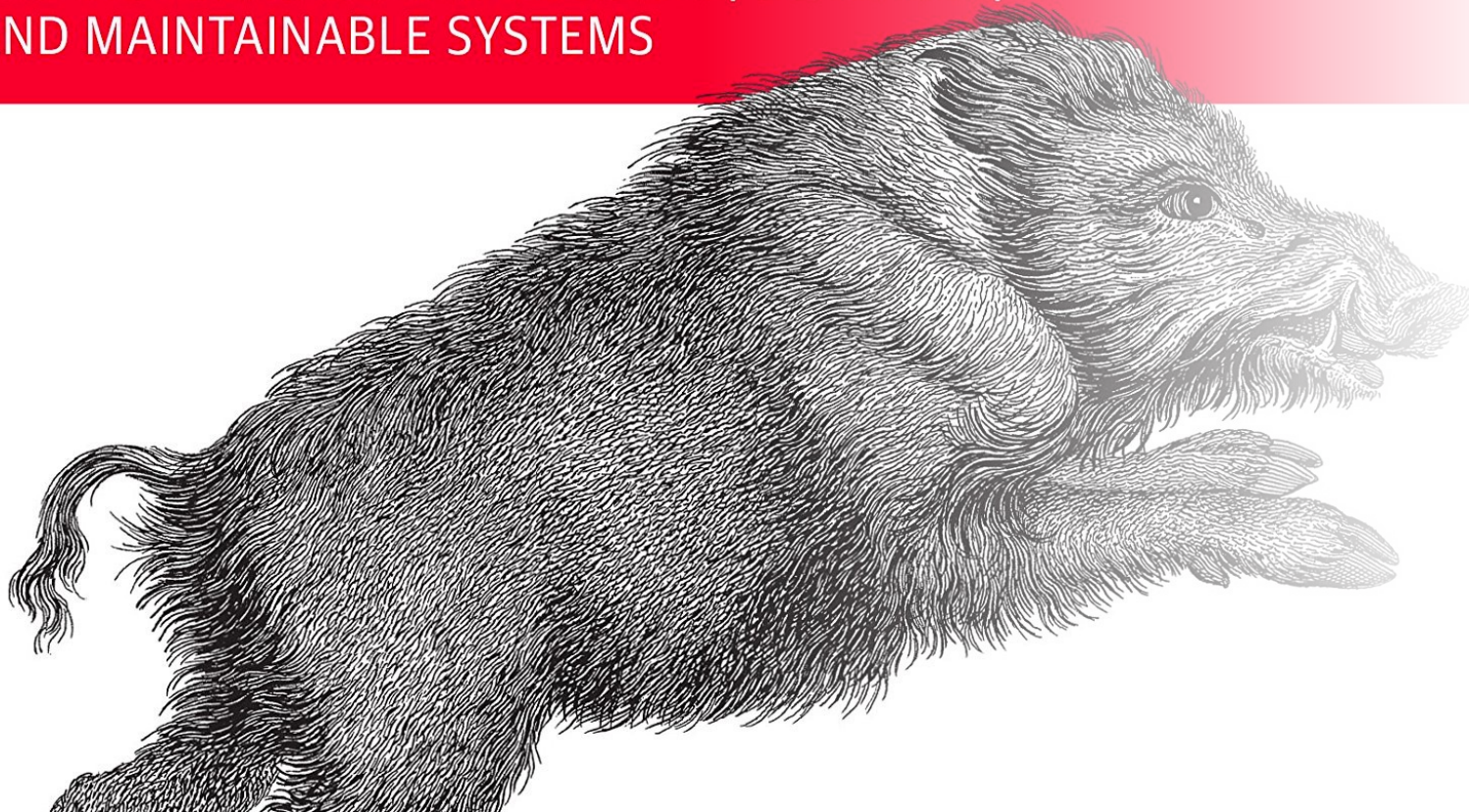



Data-Intensive Applications

THE BIG IDEAS BEHIND RELIABLE, SCALABLE,
AND MAINTAINABLE SYSTEMS



Chapter 5 : Replication Part 2: Multi Leader Replication

Replication



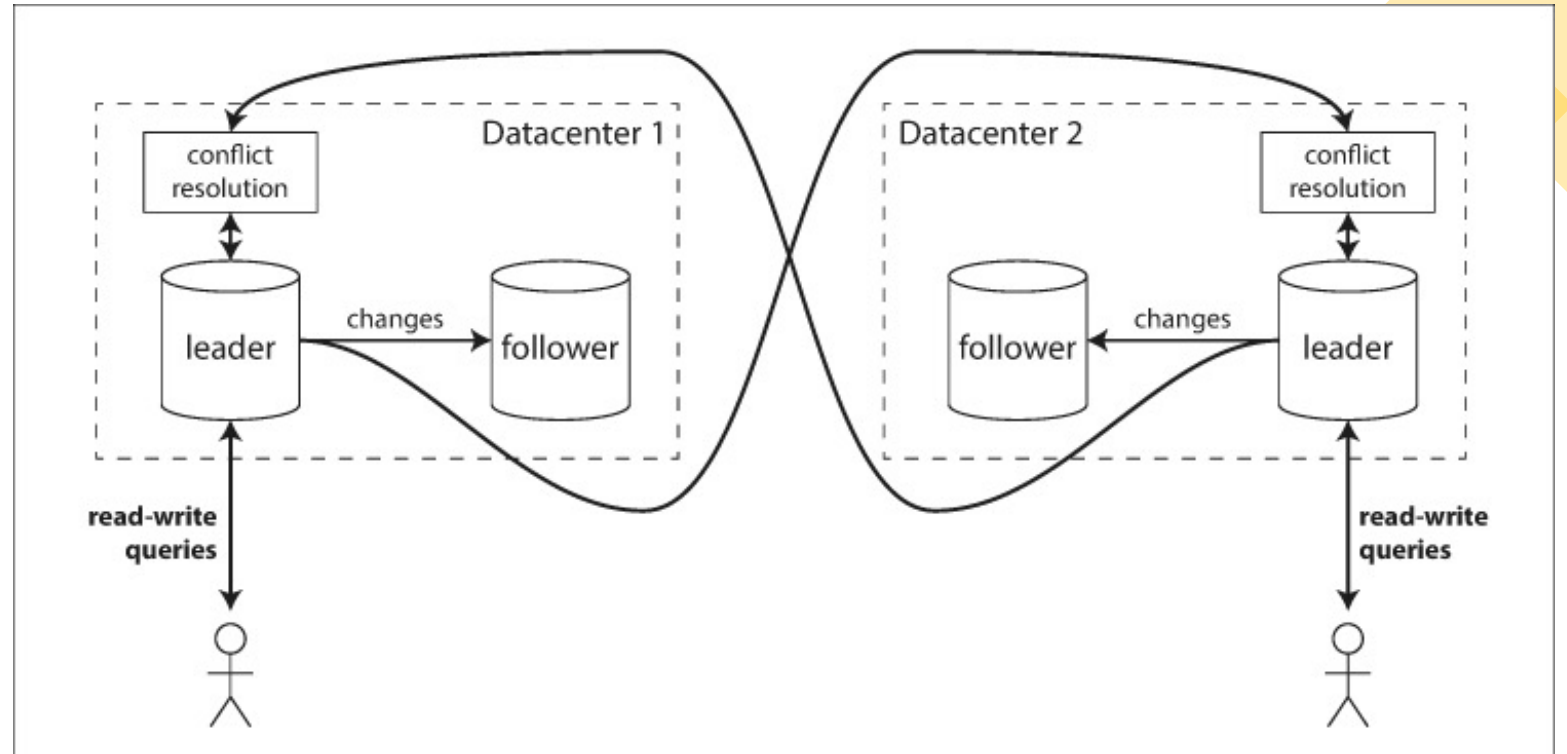
Single
Leader

Multi
Leader

Leaderless

Multi Leader Replication

- Problem with Single leader – single point of failure for write operations.
- Multi leader in single data center.
- Multi leader in multiple data center.



Multi Leader and Single leader in multiple data center

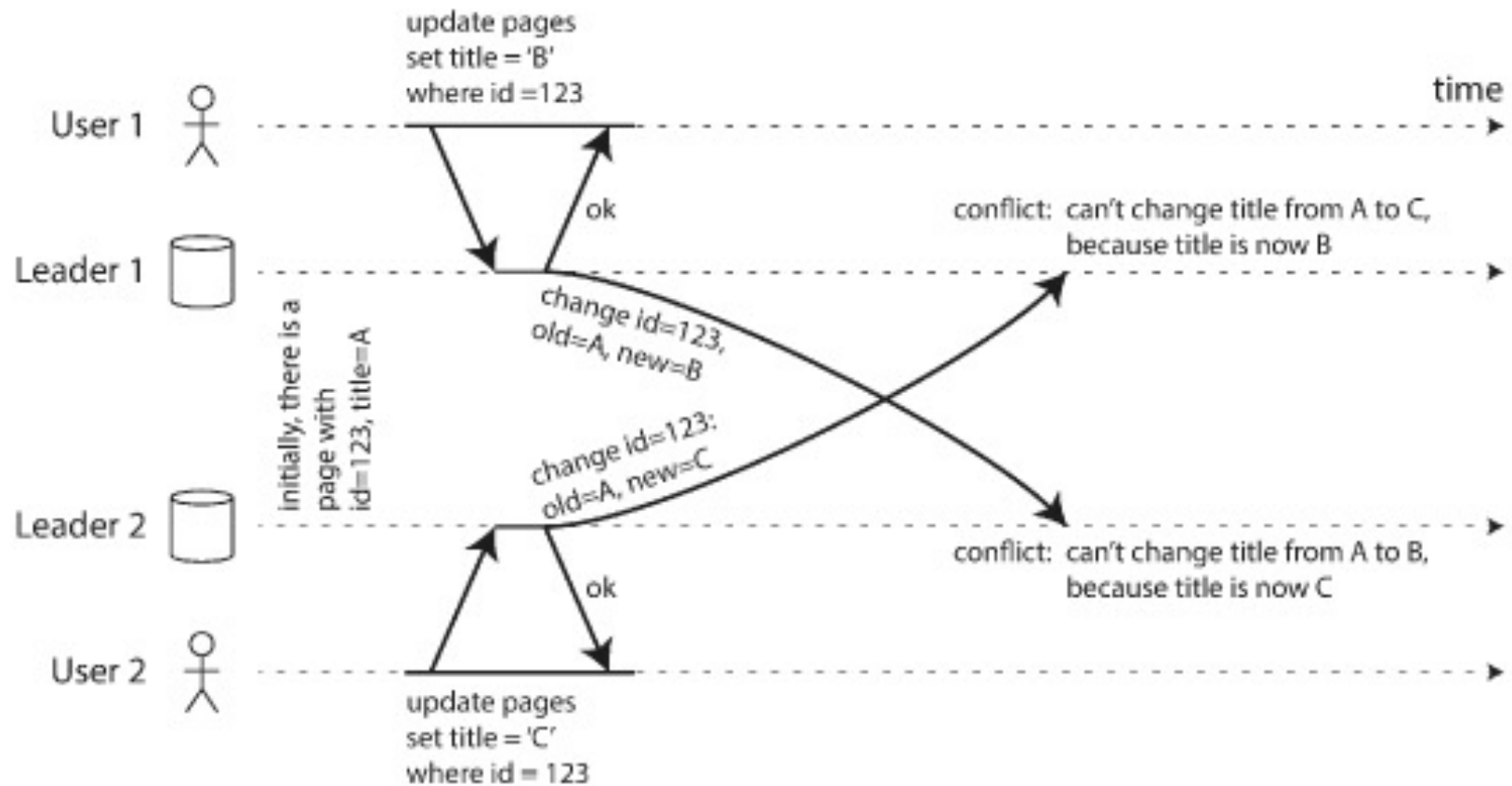
- Performance
 - If multiple data centers and single leader – every write goes to that data center with leader- adds latency.
 - In multileader and multi data center config – write processed in local data center and replicated async in other data centers.
- Tolerance of datacenter outages
 - In single leader config – if data center with leader fails, follower in another data center is promoted as leader.
 - In multi leader config , each data center has a leader.
- Tolerance of network problems
 - Traffic between data centers – over internet – less reliable.
 - Single leader config – sensitive to inter datacenter link
 - Multi leader config with async replication can tolerate network problems.

Problems with multiple leader

- Same data may be modified concurrently in two different datacenters and that write conflict must be resolved.
- Autoincrementing keys, triggers and integrity constraints can be a problem.
- Use cases of multi leader
- Calendar apps in your devices
- Collaborative editing tools



Handling write conflicts




Synchronous vs Asynchronous conflict detection

- In single leader – second writer will wait for the first one or block the database or abort the transaction and ask user to retry.
- In multi leader– both writes are successful, and conflict is detected async at later stage when user cannot be asked to resolve the conflict.
- Should we do Conflict detection synchronous?
 - Wait for writes to be replicated to all replicas.
- Conflict avoidance
 - All writes of a particular record go through the same leader
- Converging towards a consistent state
 - Single leader – writes are sequential, multi leader – no order



Convergent Conflict Resolution

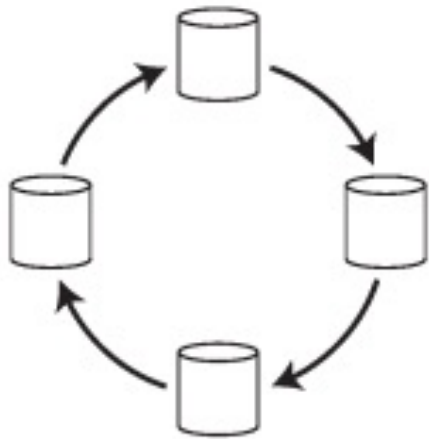
- Each write a unique ID(timestamp, UUID or hash), pick the highest ID as the winner – Last Write Wins(LWW) – prone to data loss.
 - Each replica a unique ID and let writes that originate at a higher numbered replica takes precedence over lower numbered replica – again data loss.
 - Merge values together and concatenate.
 - Record conflict in some data structure and apply logic at application code.
- 

Custom conflict resolution logic

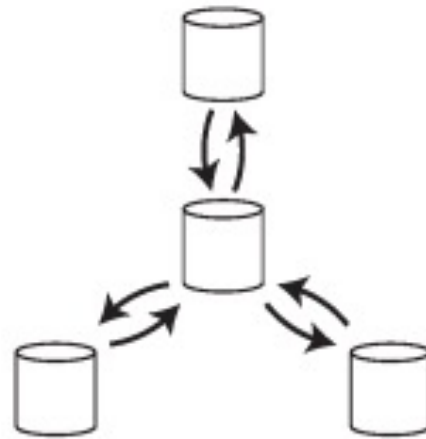
- On write – as DB detects a conflict in the log of replicated changes – calls conflict handler.
- On read – when a conflict is detected – all conflicting writes are stored – when next time data is read – multiple versions are returned to the application and application resolves conflicts and writes back to DB.
- Conflict free replicated datatypes.
- Mergeable persistent data structures – similar to Git – uses three way merge function.
- Operation transformation – Google Docs



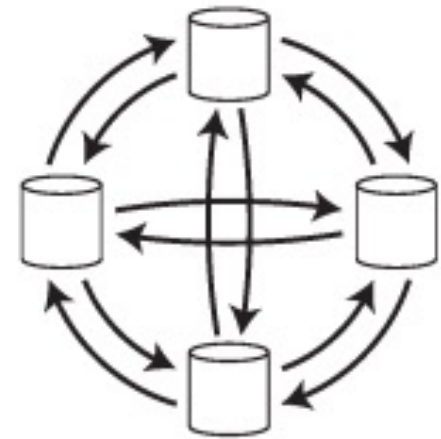
Topologies in Multi Leader



(a) Circular topology



(b) Star topology



(c) All-to-all topology

Summary



What is Multi Leader Replication?



Problems with multi leader approach



Handling write conflicts



Conflict resolution



Topologies in Multi Leader



Thank You!