CS 839: Topics in Database Management Systems

Lecture 8: Transaction Processing-2

Xiangyao Yu

10/2/2023

Updates

Mon 10/9. Project meetings (No lecture)

- Meeting (optional) with the instructor to discuss the course project
- Location: CS4361. Signup sheet

Wed 10/11. No Lecture

- Attend Wisconsin DB Affiliates Workshop
- **10/12 Thu (optional) @Union South Northwoods (3rd floor)**. Whole-day workshop
- **10/13 Fri 9:00-10:30 (required) @CS 1240**. Sponsor talks from Microsoft, Google, and Snowflake.
- Attend and **submit a review for the talks**

More papers uploaded to course website

Group Discussion from Lecture #7

Socrates and Aurora use a storage disaggregation architecture but supports only a single write node. Spanner supports multiple write nodes, but largely follows a shared-nothing architecture. Is it possible to support multiple write nodes in a storage disaggregation architecture? How would you design such a system? What are the advantage and disadvantages of your design compared to Spanner?

Besides logging, storage, and computation, what other functions are good candidates for disaggregation in a transactional database?

Group Discussion from Lecture #7

Multi-master design

- Client request go to any read/write node
- A lock in the landing zone for multiple writers. Need to read from the landing zone.
- Need a service like Chubby to coordinate transactions with conflicts
- Data inconsistency with multiple writers
- Single server processing all write requests.

Partitioned/sharded design

- Partition the database across compute nodes
- Use 2PC to manage distributed transactions
- Disadvantages: complexity, latency overhead, operational challenges

2PC and Paxos solve very different problems

Group Discussion from Lecture #7

Other candidates for disaggregation in a transactional database

- Transaction coordination, centralized locking service
- Concurrency control
- Indexing
- Data caching and coherence management
- Data filtering and pruning
- Hardware acceleration
- Failure recovery

Control plane

- Query optimization
- Load balancer
- Monitoring tools, auto-configuration and scheduled upgrades.
- Security, authentication authorization

Transaction Processing-2 – Q/A

Advantages of having a NoSQL database? (most DBs adopt SQL)

Atomic writes in Cornus?

Cornus vs. Paxos Commit?

Cornus still blocks when storage fails; how to handle such failures?

How well could Percolator integrate with existing data systems?

Discussion Questions

The architecture of Cornus provides one answer to the question we discussed in last lecture—it supports multiple writers in a disaggregation architecture, by sharding data across compute nodes. Do you see any limitation of this architecture? Can you think of any optimization to mitigate such limitations?

Can you think of one specific design aspect to improve in the architecture of FoundationDB?