Practical experiences with a BFT middleware database

João Sousa, Alysson Bessani
Outline

- Protocol
- Implementation
- Evaluation (local and wide-area)
- Conclusion
Replication

• Sharing information to ensure consistency between redundant resources
• Data/computations are propagated to a set of hosts known as replicas
• Purposes:
  – Improve performance;
  – Enforce availability;
  – Provide fault-tolerance.
• Two replication approaches: passive and active
Byzantium (EuroSys'11)

- Middleware for transactional replicated databases
- Descentralized and concurrent transaction execution
- Tolerates Byzantine faults
- Supports off-the-shelf Database management systems (DBMS)
- Requires total order broadcast primitive
  - Original paper uses PBFT (OSDI'99)
- Provides snapshot isolation semantics
Byzantium (single master)

Comunication steps

Slaves only execute operations at commit time
Byzantium

The original evaluation was done in a local network. How does it fare on a geo-replicated environment?
SteelDB

- JDBC driver implemented by Navigators (Tclouds project, Master Thesis of Marcel Santos)
- Simplified version of Byzantium
- Tested with PostgreSQL
- Uses BFT-SMaRt as communication layer
- 2 variants (normal & optimized)
SteelDB

Communication steps (normal)

Clients only contact slaves at commit time
SteelDB

Communication steps (optimized)

Slaves execute operations speculatively (SOSP’07)
Architecture

[Diagram showing the architecture of a system with components labeled as Application, JDBC, SteelDB, and ServiceProxy, connected to a BFT State Machine Replication (BFT-SMaRt) on the client side and replica side.]
Total Order Broadcast

- BFT-SMaRt (DSN’14): Standard algorithm
- WHEAT (SRDS’15): BFT-SMaRt optimized for wide-area replication
BFT-SMaRt

Communication steps

Request  Propose  Write  Accept  Reply
TPC Benchmark C

- Standard specification to evaluate scalability of OLTP systems
- Simulates a wholesale supplier comprised by multiple warehouses and districts
- Terminals send 5 operation types: New Order, Payment, Delivery, Order Status e Stock Level
Personal Experience
Evaluation

• 4 (BFT-SMaRt) or 5 (WHEAT) replicas
  – Both tolerating a single failure
• 30 terminals
• Default TPC-C workload with 50 warehouses
  – 45% New Orders
  – 43% Payments
  – 4% Delivery, Order Status & Stock Level
  (92% of transactions are write-heavy)
• PostgresSQL 9.6, Ubuntu 14.04
Evaluation

• Local network: quinta.navigators.di.fc.ul.pt
  – 32 GB memory, 2 quadcores with hyperthreading, magnetic disks

• Geo-distributed network: Amazon EC2 (t2.medium)
  – 4 GB memory, 2 vCPU, SSD disks
Evaluation

BenchmarkSQL results

Throughput (transactions per minute)

- 41% less
- 27% less
- 32% less
- 17% less

LAN
WAN

PostgreSQL
Proxy
SteeDB (Std w/ BFT-SmarT)
SteeDB (Opt w/ BFT-SmarT)
Evaluation

BenchmarkSQL results

Average Commit Latency (milliseconds)

- PostgreSQL
- Proxy
- SteelDB (Std w/ BFT-SMaRt)
- SteelDB (Opt w/ BFT-SMaRt)
- SteelDB (Opt w/ WHEAT)

LAN
WAN
Evaluation

• Commit latency (normal SteelDB w/ BFT-SMaRt):
  \[ L_{\text{roundtrip}} + L_{\text{total\_order}} + L_{\text{contention}} + L_{\text{exec\_ops}} + L_{\text{disk\_write}} \]

• LAN (30 terminals):
  \[ 2 \times \sim 1\ ms + \sim 1000\ ms + \sim 13\ ms + \sim 33\ ms \]

• WAN (30 terminals):
  \[ \sim 170\ ms + \sim 410\ ms + \sim 35\ ms + \sim 13\ ms + \sim 1\ ms \]
  \[ = \sim 630\ ms \]

• By default, PostgreSQL flushes buffers to disk (fsync)

• Amazon EC2 provides storage on SSD
Evaluation

What if Fsync is turned off in the local network?
Is SteelDB worth it?

• A non-replicated database should have Fsync turned on to prevent data loss/corruption
  – PostgreSQL throughput was around 2300 transactions/minute

• SteelDB provides redundancy, so Fsync can be turned off
  – Normal SteelDB throughput was a around 2400 transactions/minute
  – Optimized SteelDB throughput was 8700 transactions/minute
Conclusions

• Byzantium: database middleware for Byzantine fault tolerance
• SteelDB: our implementation of (part of) Byzantium
• Sound solution both for local and wide-area networks
Questions?