TAO: Facebook’s Distributed Data Store for the Social Graph

Authors: Nathan Bronson, et. al

Presentors: Zhichao Cao, Fenggang Wu
10/01/2018
The Social Graph

At the summit — at Charlotte Dome.

Source: https://www.usenix.org/node/174510
**Objects = Nodes**
- Identified by unique 64-bit IDs
- Typed, with a schema for fields

**Associations = Edges**
- Identified by <id1, type, id2>
- Bidirectional associations are two edges, same or different type

id: 1807 =
type: POST
str: “At the summ...

<1807,COMMENT,2003>
time: 1,371,704,655

id: 2003 =
type: COMMENT
str: “how was it...

id: 308 =
type: USER
name: “Alice”

<308,AUTHORED,2003>
time: 1,371,707,355

<2003,AUTHOR,308>
time: 1,371,707,355

Center for Research in Intelligent Storage

Source: https://www.usenix.org/node/174510
Objects and Associations API

Reads – 99.8%
- Point queries
  - obj_get 28.9%
  - assoc_get 15.7%
- Range queries
  - assoc_range 40.9%
  - assoc_time_range 2.8%
- Count queries
  - assoc_count 11.7%

Writes – 0.2%
- Create, update, delete for objects
  - obj_add 16.5%
  - obj_update 20.7%
  - obj_del 2.0%
- Set and delete for associations
  - assoc_add 52.5%
  - assoc_del 8.3%
Stage 1: Relational Database Supported

1. Low write latency
2. High read latency, difficult to achieve load balancing
3. Low scalability
Stage 2: Cache Supported (memcache)
Stage 2: Cache Supported (cache in between)

1. Higher write latency (clean the cache)
2. Lower read latency
3. Higher scalability

WebServers

Cache Layer

MySQL
Stage 3: TAO (The Associations and Objects)

Independent Scaling by Separating Roles

Web servers
- Stateless

Cache
- Objects
- Assoc lists
- Assoc counts

Database
- Sharded by id
- Servers → read qps
- Sharded by id
- Servers → bytes

Source: https://www.usenix.org/node/174510
TAO Design 1: One Tier with Sharding

Web servers

- Inefficient failure detection
- Many switch traversals

Cache

- Many open sockets
- Lots of hot spots

Database

Source: https://www.usenix.org/node/174510
TAO Design 2: Multiple Caching Tiers

Source: https://www.usenix.org/node/174510
1. Complex distributed write logic to handle to ensure data correctness, consistency

2. A lot of concurrent read or write go to the MySQL server, for the same data

3. Make tradeoffs between limiting the maximum tier size and scaling the cache

Source: https://www.usenix.org/node/174510
Follower and Leader Caches

Web servers

Follower cache

Leader cache

Database

Source: https://www.usenix.org/node/174510
Write-through Caching – Association Lists

Web servers

Follower cache

Leader cache

Database

Source: https://www.usenix.org/node/174510
Write-through Caching – Association Lists

Web servers

Follower cache

Leader cache

Database

Source: https://www.usenix.org/node/174510
Write-through Caching – Association Lists

Web servers

Follower cache
Y,A,B,C
refill X

Leader cache
range get

Database
Y,...

Source: https://www.usenix.org/node/174510
How to Scale out

Master region

Slave region 1

Slave region 2

Master DB

Slave DB

Slave DB

Center for Research in Intelligent Storage
Asynchronous DB Replication

Master data center

Replica data center

Web servers

Follower cache

Leader cache

Database

Source: https://www.usenix.org/node/174510
Improving Availability: Read Failover

Master data center

Web servers

Follower cache

Leader cache

Database

Replica data center

Source: https://www.usenix.org/node/174510
Improving Availability: Read Failover

Master data center

Web servers

Follower cache

Leader cache

Database

Replica data center

Source: https://www.usenix.org/node/174510
Improving Availability: Read Failover

Master data center

Web servers
Follower cache
Leader cache
Database

Replica data center

Source: https://www.usenix.org/node/174510
MySQL Architecture

Server's functionality like connection management/authentication... is done in this layer

Connection Management/security

SQL Parsing, execution and caching...

MyISAM
Innodb
Heap (In-Memory)
NDB Network DB

Responsible for storage and retrieval of all available information

Source: https://techsoftcompute.wordpress.com/2013/08/19/mysql-internal-architecture/
InnoDB Architecture

Service

Application

Handler API

Embedded API

Transaction

Cursor/Row

B-Tree

Page

Lock

Mini Transaction

Buffer

File Space Manager

IO

MySQL Architecture in FaceBook

Server's functionality like connection management/authentication... is done in this layer

Connection Management/security

SQL Parsing, execution and caching...

MyISAM  RocksDB  Heap (In-Memory)  NDB Network DB

Responsible for storage and retrieval of all available information

Source: https://techsoftcompute.wordpress.com/2013/08/19/mysql-internal-architecture/
RocksDB Architecture

Write

Immutable Memtable

Active Memtable (4MB)

Memory

Disk

Level 0 (4 SSTfile)

Compaction

Level 1 (10MB)

SSTfile (2MB)

Level 2 (100MB)

Log

Info Log

MANIFEST

CURRENT

Source: https://www.slideshare.net/meeeejin/rocksdb-detail
**TAO Summary**

**Efficiency at scale**
- Separate cache and DB
- Graph-specific caching
- Subdivide data centers

**Read latency**

**Write timeliness**
- Write-through cache
- Asynchronous replication

**Read availability**
- Alternate data sources
Thanks

Q/A