LogStore: A Cloud-Native and Multi-Tenant Log Database

Wei Cao, Xiaojie Feng, Boyuan Liang, Tianyu Zhang,
Yusong Gao, Yunyang Zhang, Feifei Li
Alibaba Cloud Database Department

2021.06
Database services at Alibaba Cloud

Cloud Database Management Systems (DBMS) Leader

Database Products and Services:
26 Products or Services

Enterprise Users:
100 thousands

Databases Migrated:
400 thousands

Expressway for running fully managed databases on Alibaba Cloud.
https://www.youtube.com/watch?v=5VkLDC_ulxM
The scenarios of using log data

SQL Explorer

Database Autonomy Service (DAS)

https://www.alibabacloud.com/help/doc-detail/96123.html?spm=a2c5t.11065259.1996646101.searchclickresult.44307061pHfASV

https://www.alibabacloud.com/help/doc-detail/64851.htm?spm=a2c63.p38356.b99.2.61d09bb0Xe1M PU
The log solution in Alibaba cloud
Problems & Challenges

- **Extremely High Write Throughput**
  - More than 50 millions logs per second.

- **Huge Storage Volume**
  - More than 10 PB.
  - Periodic retention and archive? Troublesome.

Figure 1: The total write throughput of Alibaba Cloud DBaaS audit logs in a day.
Problems & Challenges

- **Large Number of Tenants and Highly Skew Workload**
  - More than 100,000 tenants, different life cycle.
  - Workloads close to Zipfian distribution.
  - One tenant one store? Inefficient for most tiny tenants.

- **Log Retrieval on Massive Data**
  - Petabyte-sized historical logs.

*Figure 2: A statistics of tenants’ daily data size in the LogStore production environment, which is highly skewed and close to the Zipfian Distribution.*
Designs & Contributions

✓ Cloud native architecture
  - Combine shared-nothing and shared-data designs.
  - Best practices to leverage object storage in database.

✓ Low-latency Writes
  - Multi-replicas, WAL synchronization by Raft.
  - Real-time data visibility.

✓ Query optimization for cloud storage
  - LogBlock, column-oriented, full-column indexed, self-contained.
  - Multi-level cache.
  - Data skip and parallel pre-fetch.

✓ Dynamic Flow Scheduling on Heterogeneous Resources
  - Global traffic control algorithm.
  - Backpressure mechanism.
Architecture: Shared-Nothing VS. Shared-Data

- The most popular distributed architecture
- Data partitioned and stored on local disks
- Difficult to horizontal scaling, data repartition

Decouple computing and storage
- Leverage cloud storage, low costs
- Higher latency, depend on network

Figure 1: Multi-Cluster, Shared Data Architecture

Figure 3: Amazon Redshift system architecture
Architecture

• Controller
  - 3-nodes by ZK, one node is active
  - Metadata management
  - Cluster monitoring
  - Task scheduling, ex. checkpoints, archive, retention etc.

• Query Layer
  - Peer brokers, dispatched by SLB
  - Parsing, optimization
  - Parallel DAG execution

• Execution Layer
  - Work groups, synchronized by Raft
  - Real-time store, write-optimized
  - Data builder, transfer to read-optimized
  - File and Object Caches
Architecture – Storage Layer

• Alibaba Cloud OSS
  - A reliable and cost-effective object storage.
  - 99.9999999999% durability and 99.995% availability
  - Support HTTP(s) RESTful APIs or SDKs.

• Best practices
  - row-column hybrid storage
  - two-phase writing process
  - multi-Tenant storage
  - read-optimized LogBlock
Architecture – Log Block

- Self-contained
  - can rename or move

- Compressed
  - support Snappy, LZ4, ZSTD

- Columnar-oriented

- Full-column indexed and Skippable
  - SMA
  - Inverted index
  - BKD tree index
Load balancing

**Why Load imbalance?**
- High Skewed Workload
  - close to Zipfian Distribution
- Variations of Traffic
  - online promotions
  - business upgradation
- Heterogeneity of ECS nodes
  - Various ECS node configuration

**State of Art**
- Dynamic partition splitting
  - HBase
- Rule-based/heuristic algorithms
  - Yak
- Greedy algorithm
  - EStore
Global Traffic Control – modules

- **Monitor**
  - Collect tenant traffic, shard load and worker node load
  - Detect hot spots

- **Balancer**
  - Handle hot spots and imbalance
  - Scale out

- **Router**
  - Maintain routing tables on each broker
Global Traffic Control - modeling

**Constraints**

\[ \forall P_j \in P, f(P_j) \leq c(P_j) \]

\[ \forall D_k \in D, f(D_k) \leq \alpha \cdot c(D_k) \]

**Goals**

Maximum the traffic from \( S \) to \( T \)

\[ \sum_{i=0}^{m} f(K_i) \]

**Algorithm**

- Greedy Algorithm
- Max-Flow Algorithm
Global Traffic Control – backpressure

• Why?
  - Extreme cases which rebalancing cannot respond in time
  - Inspired by streaming computing, Heron, Flink

• Strategy
  - Monitor the log number of queue
  - Monitor the total log size of queue
  - Threshold-based trigger
  - Reverse transfer to reject writing

• BP based Raft implementation
  - Synchronizing queue
  - Apply queue
Query Optimization – data skipping

- **Less is More?**

- Skip on log block map
  - ‘tenant_id’, ‘ts’

- Skip on column
  - ‘fail’ column

- Column with index
  - Scan index directly, ‘ip’

- Column without index
  - Skip on column block, <min, max>
  - Scan related block

![Diagram of query optimization](image-url)
Query Optimization – multi-level cache

• **How to bridge the gap between cloud storage and local storage?**

• Pass through whole query process
  - meta cache
  - indexes cache
  - data cache

• Multi-level
  - Memory block cache (8GB)
  - SSD block cache (200GB)
  - Memory object cache
Query Optimization – parallel prefetching

- Single thread per query?

- Multi-threads? Future direction

- Bottlenecks on query execution
  - Waiting IOs from cloud storage
  - Data computing
  - Vectorized execution

- Tradeoff
  - Parallel prefetching, then single thread execution
  - Avoid IO blocking with cloud
Benchmark – write throughput and traffic control

• Greedy vs. Max Flow
  - write throughput
  - routes of tenants

• Before vs. After Balancing
  - Worker accesses distribution
  - CPU utilization
Benchmark – query optimization

• Overall Performance
  - P99 within 2 sec, P90 within 1 sec.
  - About 10 times improvement.

• Data Skipping
  - Average query latency improved 1.7 times
  - More obvious to large tenant.
  - About 2.7 times improvement for large tenants.

• Parallel Prefetch from Oss
  - Without
    - 18.5 times slower than local
  - With
    - 6 times slower than local
Conclusion and Future

• LogStore has been deployed in Alibaba Cloud,
  - More than 500 machines.
  - Process more than 100GB logs per second.
  - Run stably for more than two years.

• Future works
  - Read/Write Splitting
  - Parallel query based on cloud storage
  - Add light-weight index structures on real-time store
  - Vectorized execution and JIT compilation
Thanks