Theme: Database

Topic: POLARDB multiple-master writability technology base on distributed shared memory

Background

The current cloud-native database adopts the shared storage architecture with Single-Writer-Multiple-Readers. Through separating the computing and storage, the storage and read computing nodes can be elastically scaled, which provides ultimate elasticity and scalability for the cloud-native databases. However, there are still two obvious bottlenecks in the shared storage architecture, limited memory elasticity and single-master cluster problems.

The memory is critical to the performance of the database. However, it is hard to achieve the independent elasticity of memory in the current cloud-native architecture, because the memory is tightly coupled with CPU in the computing node. Losing memory elasticity means losing real second-level elasticity and serverless. Therefore, it is significantly necessary to make CPU and memory separated in a cloud-native database, and realize the independent elasticity of memory. In the meantime, the separation of CPU and memory allows multiple CPUs to share the same memory, which improves memory resource utilization effectively.

The current cloud-native database cluster is always a single-master cluster. Only a single DB instance supports writing operations and any other instances are read-only. In other words, the overall cluster's writing performance is limited by a single DB instance and lacks of write scaling capabilities. To solve this problem, we aim to design and implement a new database system based on distributed shared memory, which realizes multiple-master writability and external consistency, as well as RTO = 0 to support core business such as bank transactions. To achieve this goal, we need to break through the existing bottlenecks, the cross-node distributed physical consistency (B + Tree concurrency control and Crash Safe) and logic consistency (distributed transaction and external consistency).

Traditional distributed database systems have mostly exploited the same storage and network infrastructure since the 1970s, such as disks/flash for persistent storage, and ethernet for network transmission technique. However, the excessive access latency between physical nodes (induced by terrible I/O and network performance) made it difficult and inefficient to achieve physical consistency and logic consistency in distributed systems. The continuous emergence of hardware such as RDMA and NVM in recent years has provided the possibility of solving the above problems.

Target

Take advantages of new hardware to design and implement

- High-performance distributed memory pool supporting cache coherence
- Distributed B+TREE concurrency control and crash-safe algorithm
- High-performance distributed transaction consistency algorithm

On the basis of the above, implement a whole database cluster that supports multi-master writability and external consistency

Related Research Topics

- Distributed cache coherency algorithm based on RDMA
- B+TREE crash-safe algorithm based on NVM
- Distributed timestamp and transaction consistency
- Distributed Concurrency control algorithm of B+TREE
- Implementation of general memory pool based on RDMA

Suggested Collaboration Method

AIR (Alibaba Innovative Research), one-year collaboration project.