# Improving Cloud Database Performance With Unified BUS

Chong Chen, Jack Ng, Shu Lin, Jason Lam, Danny Chen

Huawei Research Canada
chongchen@huawei.com, jack.ng@huawei.com, shu.lin@huawei.com
jason.lam@huawei.com, danny.chen@huawei.com

## 1 Introduction

Cloud data centers are undergoing a fundamental architectural shift driven by resource disaggregation and high speed networking. Compute, memory, and storage are increasingly decoupled into independently scalable pools, enabled by technologies such as RDMA and CXL, which support low latency, high bandwidth interconnects. HUAWEI Unified BUS (UB) is a similar next generation high speed interconnect technology. UB provides a single interconnection fabric capable of both scaling out (connecting tens of thousands of compute components) and scaling up with bandwidth approaching that of on board memory Buses while maintaining ultra-low latency close to the physical limit set by the speed of light. Its generic memory semantic interface allows any component on the fabric to access another component's resources as if they were local memory.

This unified remote memory access capability opens new opportunities for cloud native database design. In this talk, we first present a concise overview of HUAWEI Unified BUS architecture and programming interface. We then show how UB technology is applied in two production cloud database systems to deliver significant performance improvements. The first case study is TaurusDB, a cloud-native OLTP service, where UB helps overcome global consistency performance bottlenecks. The second is GaussDB, a cloud-native HTAP service, where UB enables a redesigned cross node shuffle service for accelerated hybrid transactional/analytical processing.

## 2 TaurusDB Global Consistency

One of the fundamental requirements for consistency in distributed systems is Global Read-After-Write (GRAW), also known as global consistency. The expectation is that any read, which is known by the user to occur after a write, must return the value produced by that write. In a cloud native database with a single master and multiple read replicas, this requirement creates significant challenges when reads and writes are split and the read request is routed to a replica. To guarantee global consistency, the read replica must first synchronize its committed Log Sequence Number (LSN) with the master and replay the committed log stream to reach the target LSN before processing the query. This synchronization introduces substantial query latency and can lead to dramatic overall performance degradation. Consequently, most production cloud native database systems either do not support GRAW or limit it to a small subset of queries, forcing users to redirect most global-consistency reads to the master node which in turn increases load on the master and limits scalability. With Unified BUS and its remote-memory access capability, we have re-designed the system synchronization protocol covering both LSN synchronization and log shipping to minimize and hide latency while reducing CPU consumption on database nodes. The result is a significant performance and scalability improvement for both global-consistency and regular workloads

## 3 GaussDB UB-based Shuffle

In a distributed HTAP (Hybrid Transactional/Analytical Processing) database, complex queries frequently require multi-table joins across nodes. Whenever the join keys of two tables differ from their partition keys, a data shuffle across nodes is unavoidable. The shuffle phase typically involves heavy network data transfer and is therefore bandwidth sensitive. Before shuffling, a Bloom filter broadcasting phase is required to send to all nodes to reduce data volume. Although the Bloom filter itself is small, its distribution latency is critical, because the table scan must wait until the filter is in place. Both operations directly impact end-to-end query performance, particularly in large clusters, where profiling shows that network communication alone can account for over 40% of total query time for some complex workloads. By exploiting Unified BUS (UB) and its remote memory access feature, we have re-designed the communication layer to fully leverage UB's high bandwidth for the data shuffle and ultra-low latency for Bloom filter broadcast. To address scalability across large core counts on each node, we also introduced an innovative, scalable "logical mailbox" mechanism that efficiently orchestrates data exchange and further improves performance.