Abstract

On both HPC systems and clouds the continuously widening performance gap between storage and computing resources prevents us from building scalable data-intensive systems. Distributed NoSQL storage systems are known for their ease of use and attractive performance and are increasingly used as building blocks of large scale applications on cloud or data centers. However there has been many works on bridging the performance gap on supercomputers with NoSQL data stores.

This work presents a convergence of distributed NoSQL storage systems in clouds and supercomputers. It firstly presents ZHT, a dynamic scalable zero-hop distributed key-value store, that aims to be a building block of large scale systems on clouds and supercomputers. This work also presents several real systems that have adopted ZHT as well as other NoSQL systems, namely ZHTQ (a Flexible QoS Fortified Distributed Key-Value Storage System for the Cloud), FRIEDA-State (state management for scientific applications on cloud), WaggleDB (a Cloud-based interactive data infrastructure for sensor network applications), and Graph/z (a key-value store based scalable graph processing system); all of these systems have been significantly simplified due to NoSQL storage systems, and have been shown scalable performance.

Contribution

- ZHT: A lightweight reliable persistent dynamic scalable zero-hop distributed hash table
- Design and implementation of ZHT and optimized for high-end computing
- Verified scalability on 32k-core scale
- Achieving latencies of 0.1 ms and throughput of 18M ops/sec on a supercomputer and 8M ops/sec on a cloud
- Simulated ZHT on 1 million-node scale for the potential use in extreme scale systems
- ZHTQ: A Flexible QoS Fortified Distributed Key-Value Storage System for the Cloud
  - Supports different QoS latency on a single deployment for multiple concurrent applications
  - Both guaranteed and best-effort services are provided
  - Benchmarks on real system (16 nodes) and simulations (32 nodes)
- FRIEDA-State: Scalable state management for scientific applications on cloud
  - Design and implementation of FRIEDA-State
  - Lightweight capturing, storage and vector chunk-based event ordering
  - Evaluation on multiple platforms at scales of up to 64 VMs
- WaggleDB: A Dynamically Scalable Cloud Data Infrastructure for Sensor Networks
  - Design and implementation of WaggleDB
  - Supporting write concurrency, interactive command execution and inter-independent dynamic scalability
  - Evaluated with up to 128 concurrent clients
- GRAPH/z: A Key-Value Store Based Scalable Graph Processing System
  - Design and implementation of GRAPH/z, an BSP model graph processing system on top of ZHT.
  - The system utilizes data-locality and minimizes data movement between nodes.
  - Benchmarks up to 16-nodes scales.

ZHT: A Lightweight Reliable Dynamic Scalable Zero-hop Distributed Hash Table

**Motivation**
- Performance gap between storage and computing resource
- Large storage systems suffering bottle neck of metadata
- No suitable key-value store solution on HPC platforms

**Design and Implementation**
- Written in C++, few dependency
- Persistent backend: NoVoHT

**Primitives**
- Insert, lookup, remove
- append, swap, callback

**Highlighted features**
- Persistence
- Dynamic membership
- Fault tolerance by replication

**Performance**
- Distributed storage systems: ZHTQ, FusionFS, IStore
- Job scheduling/tauching system: MATRIX, Slurm++
- Other systems: Graph/z, Fabric

ZHTQ: A Flexible QoS Fortified Distributed Key-Value Storage System for the Cloud

**Motivation**
- Needs of running multiple applications on single data store
- Optimizing single deployment for many different requirements

**Design and Implementation**
- Request batching proxy
- Dynamic batching strategy

**Highlighted features**
- Adaptive request batching
- QoS support
- Traffic-aware automatic performance tuning

**Performance**
- Graph storage comparison
- Load balancing and data locality

FRIEDA-State: Scalable State Management for Scientific Applications on Cloud

**Motivation**
- Cloud for scientific applications
- Need application reproducibility and persistence of state
- Clock drifting issue in dynamic environments

**Design and Implementation**
- Use local files to store captured states
- Merge and reorder with vector clock
- Key-value store for storage and query support

**Performance**
- Transactional interface via database
- Column-family with semi-structured data for various data types

Graph/z: A Key-Value Store Based Scalable Graph Processing System

**Motivation**
- Cloud for scientific applications
- Need application reproducibility and persistence of state
- Clock drifting issue in dynamic environments

**Design and Implementation**
- Use local files to store captured states
- Merge and reorder with vector clock
- Key-value store for storage and query support

**Performance**
- Transactional interface via database
- Column-family with semi-structured data for various data types

Acknowledgement

Distributed NoSQL Storage for Extreme-Scale System Services

Tonglin Li¹, Ioan Raicu¹,²
¹Illinois Institute of Technology, ²Argonne National Laboratory

Selected Publications

- WaggleDB: A Dynamically Scalable Cloud Data Infrastructure for Sensor Networks
  - Design and Implementation of WaggleDB
  - Supporting write concurrency, interactive command execution and inter-independent dynamic scalability
  - Evaluated with up to 128 concurrent clients
- GRAPH/z: A Key-Value Store Based Scalable Graph Processing System
  - Design and implementation of GRAPH/z, an BSP model graph processing system on top of ZHT.
  - The system utilizes data-locality and minimizes data movement between nodes.
  - Benchmarks up to 16-nodes scales.