Distributed NoSQL Storage for Extreme-Scale System Services

Tonglin Li[1], Ioan Raicu[1, 2]

[1]Illinois Institute of Technology, [2]Argonne National Laboratory

Abstract

On both HPC systems and clouds the continuously widening performance gap between storage and computing resource 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 are not 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 ZHT/Q (a Flexible QoS Fortified Distributed Key-Value Storage System for the Cloud), FREIDA-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 light-weight reliable persistent dynamic scalable zero-hop distributed hash table**
  - Design and implementation of ZHT and optimized for high-end computing
  - Verified scalability on 32K-cores scale
  - Achieving latencies of1.1ms and throughput of 18M ops/sec on a supercomputer and 0.8ms and 1.2M ops/sec on a cloud
  - Simulated ZHT on 1 million-node scale for the potential use in extreme scale systems.
- **ZHT/Q: 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 (512 nodes)
- **FRIEDA-State: Scalable state management for scientific applications on cloud**
  - Design and implementation of FRIEDA-State
  - lightweight capturing, storage and vector clock-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 high write concurrency, transactional command execution and tier-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, a BSP model graph processing system on top of ZHT.
  - The system utilizes data-locality and minimize data movement between nodes.
  - Benchmarks up to 16-nodes scales.

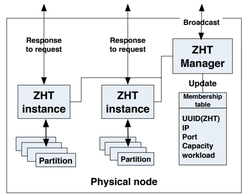
ZHT: A Light-weight Reliable Dynamic Scalable Zero-hop Distributed Hash Table

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
- Modified Consistent hashing
- Persistent backend: NoVoHT



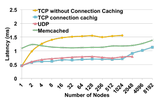
Primitives

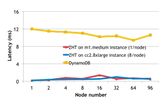
- `insert, lookup, remove`
- `append, cswap, callback`

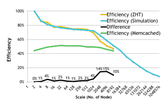
Highlighted features

- Persistence
- Dynamic membership
- Fault tolerance via replication

Performance







Applications

- **Distributed storage systems:** ZHT/Q, FusionFS, IStore
- **Job scheduling/launching system:** MATRIX, Slurm++
- **Other systems:** Graph/Z, Fabriq

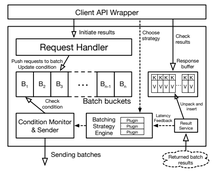
ZHT/Q: A Flexible QoS Fortified Distributed Key-Value Storage System for the Cloud

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



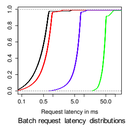
Highlighted features

- Adaptive request batching
- QoS support
- Traffic-aware automatic performance tuning

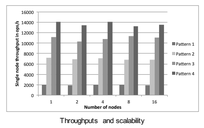
Performance

Workloads with multiple QoS

QoS latency time	1ms	10ms	100ms	1000ms
Pattern 1	25%	25%	25%	25%
Pattern 2	4%	32%	32%	32%
Pattern 3	0%	33%	33%	33%
Pattern 4	0%	0%	50%	50%



Batch request latency distributions



Throughputs and scalability

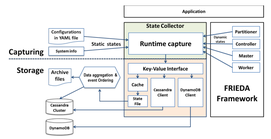
FRIEDA-State: Scalable State Management for Scientific Applications on Cloud

Motivation

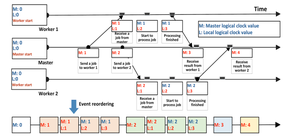
- Cloud for scientific applications
- Need application reproducibility and persistence of state
- Clock drifting issue in dynamic environments

Design and Implementation

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



System architecture



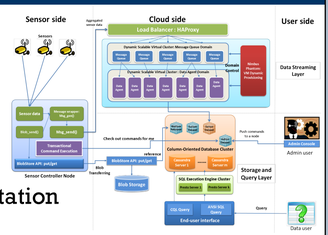
Distributed event ordering

Performance

WaggleDB: A Dynamically Scalable Cloud Data Infrastructure for Sensor Networks

Motivation

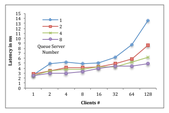
- Unreliable network
- Wide range of request rate
- Admins interaction to nodes
- High write concurrency
- Many data types for sensors
- Scalable architecture



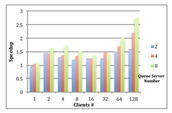
Design and Implementation

- Multi-tier architecture
- Independent components in each tier
- Organize each tier as a Phantom domain for dynamic scaling
- Message queues as write buffers
- Transactional interaction via database
- Column-family with semi-structured data for various data types

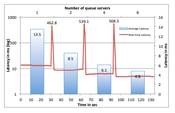
Performance



Scalable current write



Speedup of distributed servers



Dynamic tier scaling

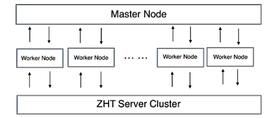
Graph/Z: A Key-Value Store Based Scalable Graph Processing System

Motivation

- Processing graph query
- Handle big data set
- Fault tolerance

Design and Implementation

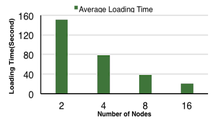
- Pregel-like processing model
- Using ZHT as backend
- Partitioning at master node

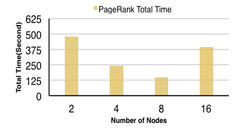


Highlighted features

- Data locality
- Load balance

Performance





Selected Publications

Journal papers

•**Tonglin Li**, Xiaobing Zhou, Ioan Raicu, etc., *A Convergence of Distributed Key-Value Storage in Cloud Computing and Supercomputing*, Journal of CCPE 2015.

•Iman Sadooghi, **Tonglin Li**, Kevin Brandstatter, Ioan Raicu, *etc. Understanding the Performance and Potential of Cloud Computing for Scientific Applications*, IEEE Transactions on Cloud Computing (TCC), 2015

•Ke Wang, Kan Qiao, **Tonglin Li**, Michael Lang, Ioan Raicu, etc. *Load-balanced and locality-aware scheduling for data-intensive workloads at extreme scales*, Journal of CCPE 2015.

Conference papers

•**Tonglin Li**, Ke Wang, Dongfang Zhao, Kan Qiao, Iman Sadooghi, Xiaobing Zhou, Ioan Raicu, *A Flexible QoS Fortified Distributed Key-Value Storage System for the Cloud*, IEEE International Conference on Big Data, 2015

•**Tonglin Li,** Kate Keahey, Ke Wang, Dongfang Zhao, Ioan Raicu, *A Dynamically Scalable Cloud Data Infrastructure for Sensor Networks*, *ScienceCloud 2015*

•**Tonglin Li,** Ioan Raicu, Lavanya Ramakrishnan, *Scalable State Management for Scientific Applications in the Cloud*, IEEE International Congress on Big Data 2014

•**Tonglin Li,** Xiaobing Zhou, Ioan Raicu, etc. ZHT: *A Light-weight Reliable Persistent Dynamic Scalable Zero-hop Distributed Hash Table,* IPDPS, 2013.

•Dongfang Zhao, Zhao Zhang, Xiaobing Zhou, **Tonglin Li**, Ke Wang, Dries Kimpe, Philip Carns, Robert Ross, and Ioan Raicu. *FusionFS: Towards Supporting Data-Intensive Scientific Applications on Extreme-Scale High-Performance Computing Systems*, IEEE International Conference on Big Data 2014

•Ke Wang, Xiaobing Zhou, **Tonglin Li**, Dongfang Zhao, Michael Lang, Ioan Raicu. *Optimizing Load Balancing and Data-Locality with Data-aware Scheduling*, IEEE International Conference on Big Data 2014

Posters and extended abstracts

•**Tonglin Li**, Chaoqi Ma, Jiabao Li, Xiaobing Zhou, Ioan Raicu, etc. , *GRAPH/Z: A Key-Value Store Based Scalable Graph Processing System*, IEEE Cluster 2015

•**Tonglin Li**, Kate Keahey, Rajesh Sankaran, Pete Beckman, Ioan Raicu, *A Cloud-based Interactive Data Infrastructure for Sensor Networks*, SC2014

•**Tonglin Li**, Raman Verma, Xi Duan, Hui Jin, Ioan Raicu. *Exploring Distributed Hash Tables in High-End Computing*, ACM SIGMETRICS Performance Evaluation Review (PER), 2011

Acknowledgement

NSF · U.S. Department of Energy · Argonne National Laboratory · Berkeley Lab · amazon web services · FutureGrid