Distributed Key-Value Store on HPC and Cloud Systems

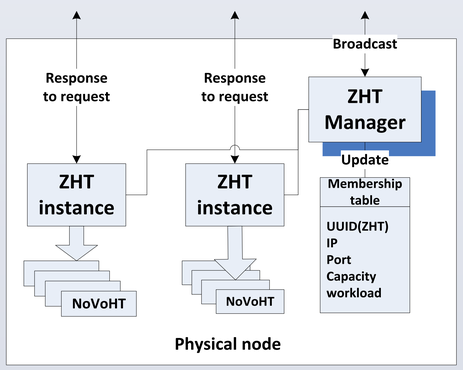
Tonglin Li[1], Xiaobing Zhou[1], Kevin Brandstatter[1], Ioan Raicu[1,2]
[1]Department of Computer Science, Illinois Institute of Technology
[2]Mathematics and Computer Science Division, Argonne National Laboratory

Abstract

ZHT is a zero-hop distributed hash table, which has been tuned for the requirements of high-end computing systems. ZHT aims to be a building block for future distributed systems. The goals of ZHT are delivering high availability, good fault tolerance, high throughput, and low latencies, at extreme scales of millions of nodes. ZHT has some important properties, such as being light-weight, dynamically allowing nodes to join and leave, fault tolerant through replications, persistent, scalable, and supporting unconventional operations such as append. ZHT scaled up to 32K-cores with latencies of 1.1ms and 18M operations/sec throughput on IBM Blue Gene/P supercomputer, and 96 nodes on Amazon EC2 cloud with 800ns latency and 1.2M ops/s throughput.

In previous work we proved ZHT's excellent performance and scalability on supercomputers, and in this work we show that it also works great on cloud environment from both performance and cost perspective.

Architecture and Design



Features

- Zero-hop: constant routing time
- Dynamic membership: support nodes leaves and joins
- Persistent: NoVoHT work as backend, data flush to disk
- Fault tolerance: replication support
- Append operation: allowing data to be incrementally added to an existing value

Name	Impl.	Routing Time	Persistence	Dynamic membership	Append
Cassandra	Java	log(N)	Yes	Yes	No
Memcached	C	2	No	No	No
C-MPI	C/MPI	log(N)	No	No	No
Dynamo	Java	0 to log(N)	Yes	Yes	No
ZHT	C++	0 to 2	Yes	Yes	Yes

Experiment Setup

Supercomputer

- IBM Blue Gene/P supercomputer
- Up to 8192 nodes
- 32768 instance deployed

Commercial Cloud

- Amazon EC2
- M1.medium (2 Compute Unit)
- Cc2.8xlarge (88 Compute Units)
- 96 VMs, 768 ZHT instances

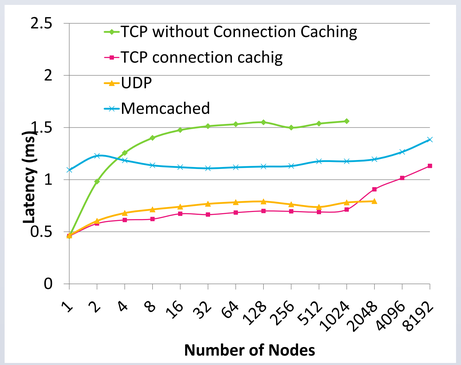
Commodity Cluster

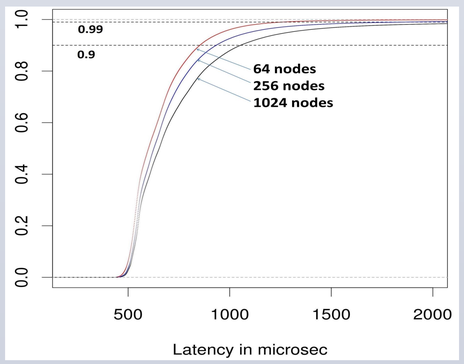
- Up to 64 nodes

Micro benchmark settings

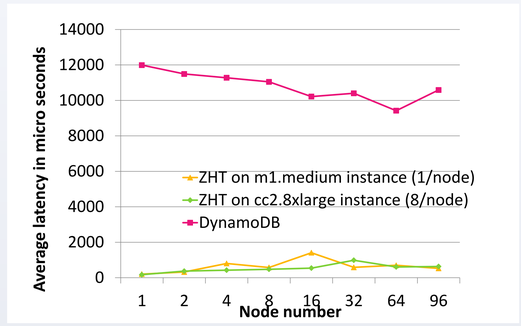
- One or more ZHT client-server pairs per node
- N clients to N servers: All-to-All communication pattern
- Random generated key–value pairs: 15 bytes key, 132 bytes value

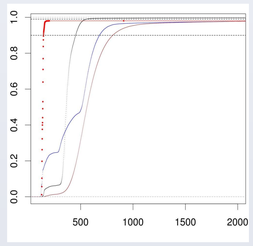
ZHT on BlueGene/P



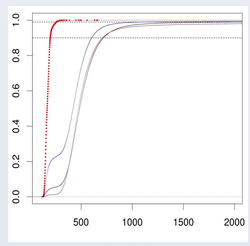


ZHT on Cloud



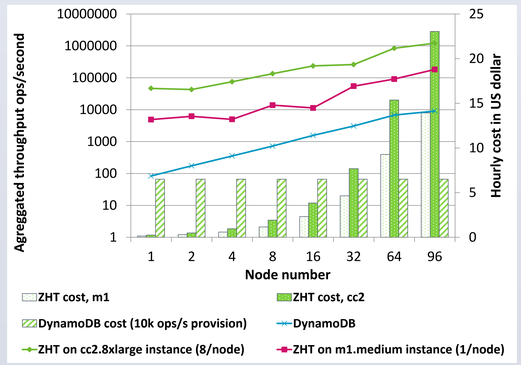


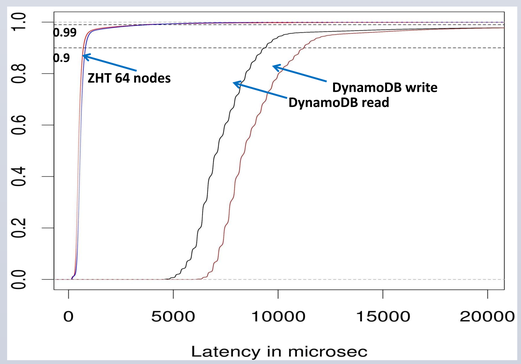
ZHT on m1.medium instance



ZHT on cc2.8xlarge instance

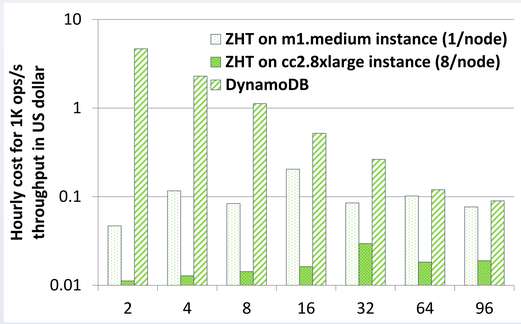
ZHT V.S. Amazon DynamoDB





Cost Comparison

Instance type	m1.medium	cc2.8xlarge	DynamoDB
Performance	2 CU	8CU	Max 10K/s
Cost	$0.112/hour	$2.4/hour	$0.65/hour for 1k/s write



Conclusion

ZHT has shown excellent performance and scalability. It's been used as building blocks of several distributed systems. Beside being highly effective on HPC environment, it also shows versatility on commercial cloud.

ZHT is more than 20 times faster than Amazon DynamoDB while costing less than 1/10 of the premium (spent on running VMs), which make it a great candidate for both a building block of distributed HPC systems and a general-purpose key-value store on cloud.

Related Work and References

- Tonglin Li, Xiaobing Zhou, Kevin Brandstatter, et al. ZHT: A Light-weight Reliable Persistent Dynamic Scalable Zero-hop Distributed Hash Table, IPDPS, 2013
- B. Fitzpatrick. "Distributed caching with Memcached." Linux Journal, 2004(124):5, 2004
- Cassandra http://cassandra.apache.org, 2012

Acknowledgment

This work was supported in part by the National Science Foundation grant NSF-1054974. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

Data-intensive Distributed Systems Laboratory
Computer Science Department
Illinois Institute of Technology, Chicago