100

DKV : A Distributed HashTable Key-Value Store

Jayanth Vangari, CWID: A20337867
Department of Computer Science, Illinois Institute of Technology, Chicago IL

ILLINOIS INSTITUTE OF TECHNOLOGY

ABSTRACT

Lately NoSQL databases have gained popularity as data storage paradigm, the motivation being their ability to easily scale to a large number of machines which is pretty difficult in the case of traditional relational databases and their ease of design. Scalability would also imply we obtain highly efficient systems with high performance. There are a variety of NoSQL databases such as Document Stores , Graph databases and Key-Value stores that have evolved over-time and are basically classified based on their data model. DKV is designed as a key-value store that uses "Distributed Hash Table" approach to store data in key-value pairs and allows operations such as insert , retrieve and delete operations on the available data. And also DKV is evaluated against other existing key-value stores like Cassandra (wide column store), Oracle NoSQL and ZHT (A Zero-Hop Distributed Hash Table) on the amazon cloud EC2 instances upto 16 nodes scale of virtual cluster and their latencies and throughputs are compared on 100k insert, retrieve and delete operations.

MOTIVATION

- The importance of data is obvious in the current era of information explosion.
- This data could be accessed, manipulated and various operations shall be performed.
- The data can be stored either by a centralized system or by distributed storage systems.
- Having a centralized system does provide low latency to access or manipulate the data, but could raise scalability issues at larger scales on a distributed platform.
- A variety of distributed storage systems like the NOSQL systems are scalable and each of these systems has to offer different usecases. But most of them tend to suffer high latencies at larger scales.

PROPOSED WORK

Distributed Hash Table

- A Distributed Hash table is a decentralized system that maintains mappings from keys to values using a variety of hashing methods across number of nodes and allows to perform operations on the key-value pairs.
- DHTs with their decentralized distributed architecture are pretty much scalable.
- Since latency appears to be an issue in most of the distributed storage systems with most of the good ones tend to operate with a routing time of logN. DKV uses Distributed Hash Table and tries to minimize the latency to perform the operations.

EXPERIMENT SETUP

- Experiments carried out on Amazon's public cloud AWS EC2 instances
- m3.medium (1 vCPU, 3.75 GiB Mem, 1* 4 GB SSD)
- Nodes are scaled from 1 to 16.
- On each instance/node a client-server pair is deployed.
- N clients to N servers : All-to-All communication pattern
- A set of key-value pairs are randomly generated where key is 10 bytes and value is 90 bytes.

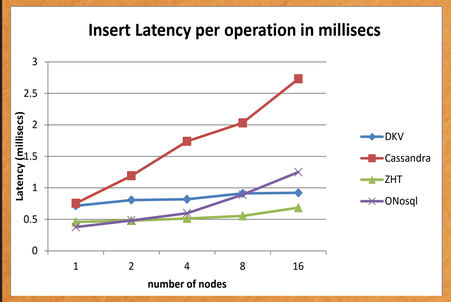
EXPERIMENT

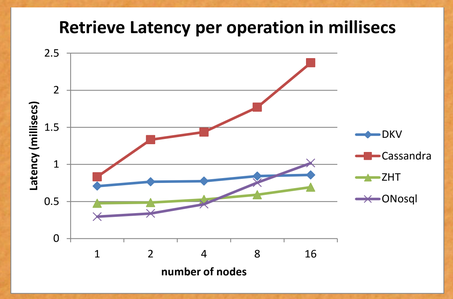
Calculated the cost of insert,retireve and delete operations across all systems . Latency (millisecs) and throughput (Kilo Ops/sec) are calculated for these operations.

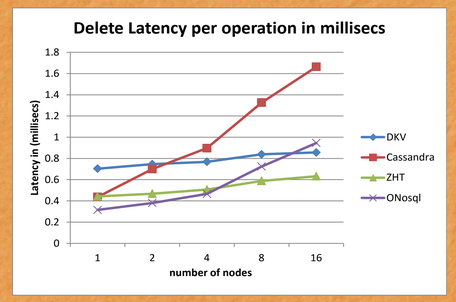
PERFORMANCE EVALUATION

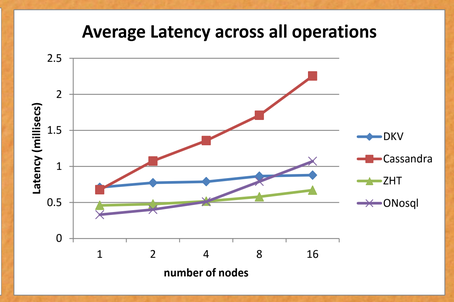
EVALUATION METHOD

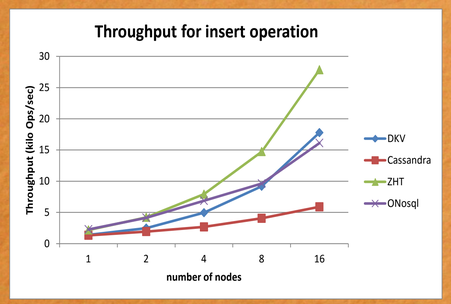
Initially a virtual cluster of 1 node is created and one server and one client is deployed and experiment for all the systems is run. Then the same is repeated for 2 nodes, 4 nodes, 8 nodes and 16 nodes. 1:1 mapping between servers and clients is maintained for complete Evaluation.

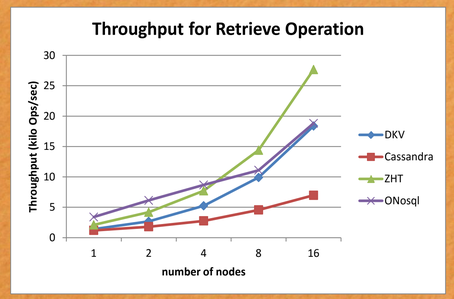


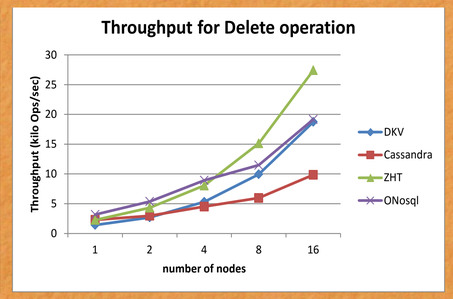


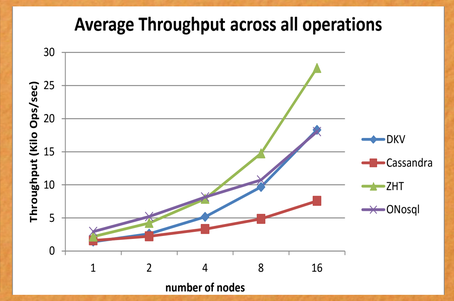












- For all the operations latency of Cassandra increases at higher rate and throughput at lower rate as nodes are scaled from 1 to 16.
- Cassandra takes logN routing time.The lower performance of Cassandra as compared to other key value stores is due to the moderate network of m3.medium aws instances.
- Oracle NoSQL although seems to perform better than DKV up to 4 nodes, the latency of is seen to be increased gradually as Oracle NOSQL uses range search for data access.
- ZHT which also uses distributed hash table for storing key-value pairs shows very good performance upto 16 nodes with very little increase in latency and almost linear increase in throughput. All the operations have similar latencies, this should be communication cost.
- DKV is observed to show higher performance than Oracle and Cassandra with little increase in latency and higher throughput as scaled to 16 nodes.

CONCLUSION

- Both Oracle NOSQL and Cassandra have increasing latencies when nodes in virtual cluster is scaled from 1 to 16.
- As desired DKV has lower latencies and increased Throughput for 100k operations of insert,retrieve and delete compared to other key-value stores.
- ZHT also uses distributed hash table to store data in key-value pairs, shows almost linear increase in throughput when nodes are scaled larger. Thus, DHT can be used to provide faster data access across nodes.

REFERENCES

- http://datasys.cs.iit.edu/publications/2015_CCPE-zht.pdf
- https://en.wikipedia.org/wiki/NoSQL
- https://blogs.oracle.com/NoSQL/entry/multigetiterator_behavior_examples
- http://datasys.cs.iit.edu/reports/2013_GCASR13_paper_NoVoHT.pdf
- http://www.datastax.com/wp-content/uploads/2013/11/WP-DataStax-Oracle.pdf
- http://www.oracle.com/technetwork/database/database-technologies/nosqldb/documentation/nosql-vs-cassandra-1961717.pdf