An Empirical Evaluation of Distributed Key/Value Storage Systems

Akash Rafaliya A-20320800

Department Of Computer Science

Illinois Institute Of Technology

Abstract

Selecting a good system from available systems for key-values storage is a major challenge for high performance computing. It's require to evaluate each systems and compare with each other. Evaluation is measured on Latency and Throughput of systems. We evaluated 4 existing systems with our system developed in assignment – PA2. PA2 is a distributed key-value store based on socket, and developed in java. It is implemented on hashing techniques to store key-value pair on distributed system. Comparision of PA2 with other systems is visualized on graps for both latency and throughput.

Experiment Setup

Commercial Cloud

- Amazon EC2
- M3.medium (1 compute unit)
- RAM 3.75GB
- SSD 4 GB
- 16 instances

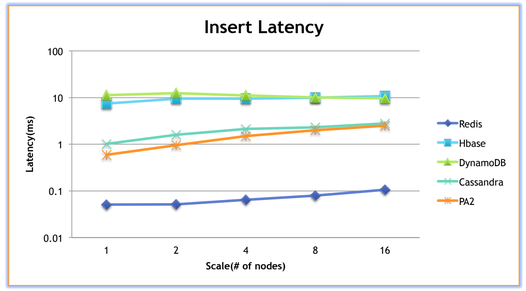
Micro benchmark settings

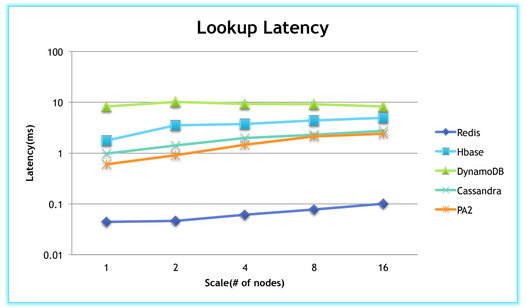
- Client-Server pair on each instance.
- N Client and N Servers with 1:1 mapping pattern
- Randomly generated key of 10 Bytes
- Randomly generated Value of 90 Bytes

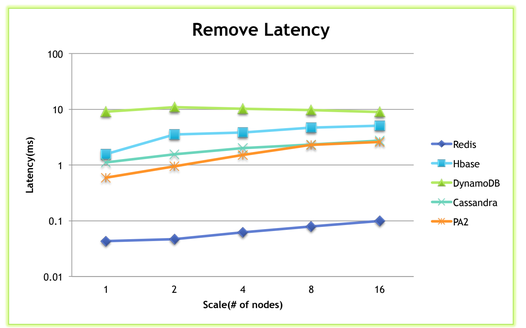
Evaluated Systems

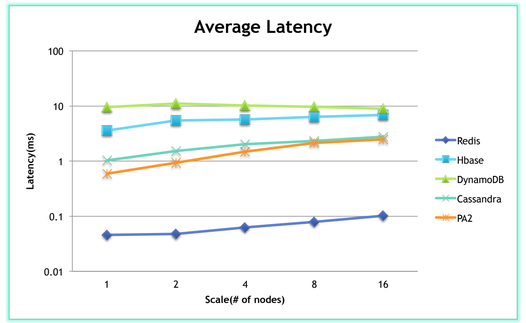
PA2
Redis
DynamoDB
Hbase
Cassandra

Latency

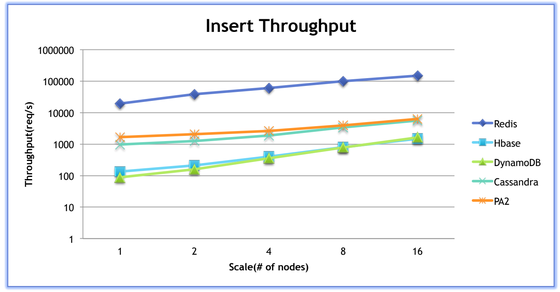


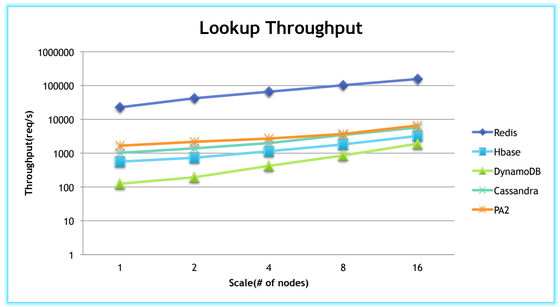


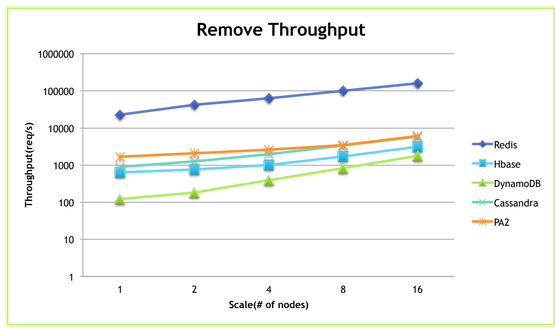


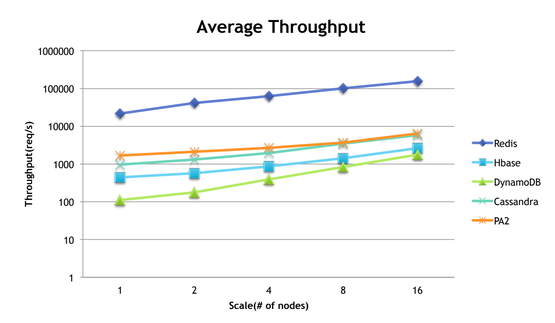


Throughput









Conclusion

The latency increases for every system with increase in the nodes except DynamoDB as the server increases dynamically with respect to the workload.

The throughput increases with increase in the nodes. From comparison in graph it concludes that Redis gives maximum throughput out of all 5 systems. Another system with higher throughput is PA2.

As scale increases all systems performance becomes very close to each other except Redis.

For higher scale throughput increases very faster then Latency for all systems.

Future Work

Include replication in PA2 system to make it's more resilience/fault tolerant. Making system dynamic in terms of scale to increase or decrease nodes without affecting network configuration.

Refrences

1. Redis http://redis.io/
2. DynamoDB https://aws.amazon.com/dynamodb/
3. Cassandra http://cassandra.apache.org/
4. Hbase https://hbase.apache.org/
5. http://www.tutorialspoint.com/redis/redis_java.htm

Acknowledgment

I would like to express sincere thanks to Dr. Ioan Raicu and his team for sharing the valuable knowledge and constant feedback during the coursework Advanced Operating System.