**Mongo DB Benchmarking with YCSB by RITIK HARIANI**

**What is YCSB?**

* The **Yahoo! Cloud Serving Benchmark** (YCSB) is an open-source specification and program suite for evaluating retrieval and maintenance capabilities of computer programs. It is often used to compare relative performance of NoSQL database management systems.
  + - * + **What is MongoDB?**
* MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema.

Installation:

1. Install Ubuntu or any other OS as VM
2. Install Maven in the terminal
   1. sudo apt install maven
3. Install Java Client
   1. sudo apt install default-jdk
4. Install Python
   1. sudo apt-get install python
5. Install MongoDB
6. Install Curl
   1. sudo apt install curl
7. Install YCSB from the github repo : <https://github.com/brianfrankcooper/YCSB/>
8. Install YCSB using following commands

curl -O --location https://github.com/brianfrankcooper/YCSB/releases/download/0.17.0/ycsb-0.17.0.tar.gz

tar xfvz ycsb-0.17.0.tar.gz

cd ycsb-0.17.0

Workloads

* Workload A: Update heavy workload: 50/50% Mix of Reads/Writes
* Workload B: Read mostly workload: 95/5% Mix of Reads/Writes
* Workload C: Read-only: 100% reads
* Workload D: Read the latest workload: More traffic on recent inserts
* Workload E: Short ranges: Short range based queries
* Workload F: Read-modify-write: Read, modify and update existing records

Commands to load and run the workloads:

Method 1:

1. Load

./bin/ycsb load mongodb -s -P workloads/workloada -p recordcount=100000 -threads 16 > outputLoad.txt

1. Run

./bin/ycsb run mongodb -s -P workloads/workloada -p operationcount=100000 -p recordcount=100000 -threads 16 > outputRun.txt

This can be similarly be done by running for all workloads from A to F

Method 2:

1. Load the database, using workload A’s parameter file (workloads/workloada) and the “-load” switch to the client.
2. Run workload A (using workloads/workloada and “-t”) for a variety of throughputs.
3. Run workload B (using workloads/workloadb and “-t”) for a variety of throughputs.
4. Run workload C (using workloads/workloadc and “-t”) for a variety of throughputs.
5. Run workload F (using workloads/workloadf and “-t”) for a variety of throughputs.
6. Run workload D (using workloads/workloadd and “-t”) for a variety of throughputs. This workload inserts records, increasing the size of the database.
7. Delete the data in the database.
8. Reload the database, using workload E’s parameter file (workloads/workloade) and the "-load switch to the client.
9. Run workload E (using workloads/workloade and “-t”) for a variety of throughputs. This workload inserts records, increasing the size of the database.

YCSB Properties:

* TIER 1 – PERFORMANCE
  + There is an inherent trade-oﬀ between latency and throughput: on a given hardware setup, as the amount of load increases, the latency of individual requests increases as well since there is more contention for disk, CPU, network and so on
  + Threads increase the workload into the system and hence, throughput also increases. But the throughput saturates at a specific number of threads and will stay constant for further increase in the number of threads. But too much increase in the number of threads causes overload and the throughput starts to decrease.
* TIER 2 – SCALING
  + A key aspect of cloud systems is their ability to scale elastically, so that they can handle more load as applications add features and grow in popularity. The Scaling tier of the database examines the impact on performance as more machines are added to the system.

Factors : Scale-up and Elastic speed up

* + Scale up: Adding new machines and increasing loads
    - A good database’s performance should remain constant and the throughput should increase proportionally
  + Elastic Speed up: New machines are adding during an ongoing workload
    - A good datbase’s performance should increase with the slightest of disruption

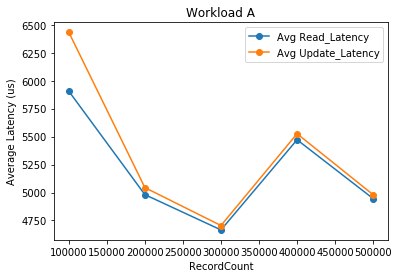
Advantages of NoSQL over SQL databases

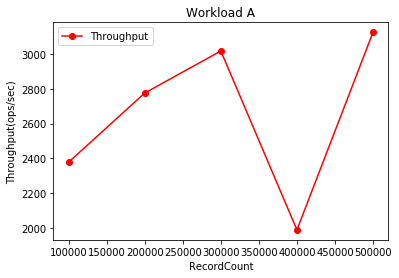
* Non-Relational means table-less: NoSQL databases are non-relational, hence, very different from SQL databases. This means they are easier to manage and they provide a higher level of flexibility with newer data models.
* Mostly Open Source and Low-Cost: The open source nature of NoSQL databases makes them an appealing solution for smaller organizations with limited budgets. The top NoSQL databases on the market today (MongoDB, MarkLogic, Couchbase, CloudDB, and Amazon’s Dynamo DB) allow for rapid processing of real-time Big Data applications in ways that are affordable.
* Easier scalability through support for Map Reduce: NoSQL database experts often use elastic scalability as a major selling point of NoSQL. NoSQL databases are designed to function on full throttle even with low-cost hardware.
* No need to develop a detailed database model: The non-relational nature of a NoSQL database allows database architects to quickly create a database without needing to develop a detailed (fine-grained) database model. This saves a lot of development time.

Analysis:

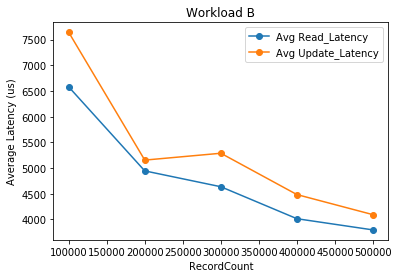
* The analysis of the performance of the Database on these workloads will provide us with a clear idea on how effective the Database is with respective to other NoSQL databases and Traditional Databases
* We shall analyse the Average Latency and the Throughput achieved for MongoDB against various workloads (A-F) loaded with record from 100k to 500k.
* The number of threads used to run these are 16.
* Threads are just used to increase/speed up the number of insertions per sec.

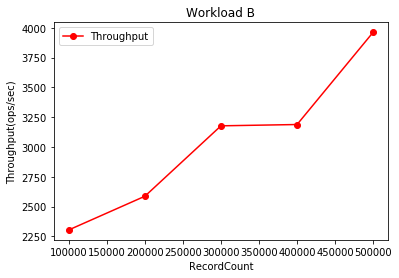
WORKLOAD A



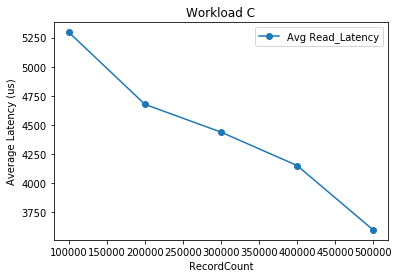


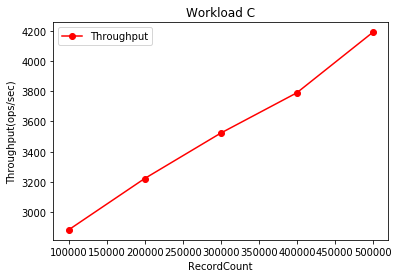
WORKLOAD B



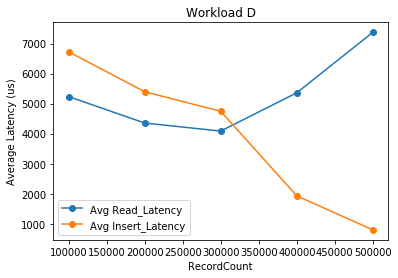


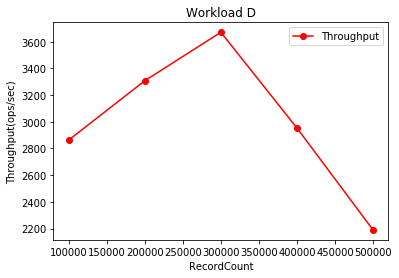
WORKLOAD C



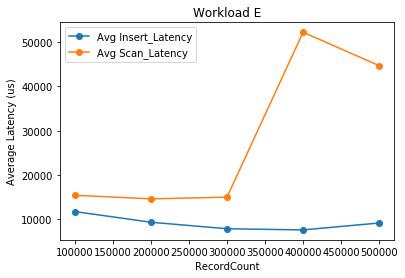


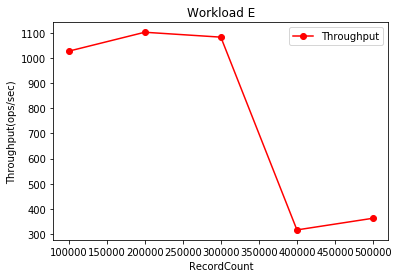
WORKLOAD D



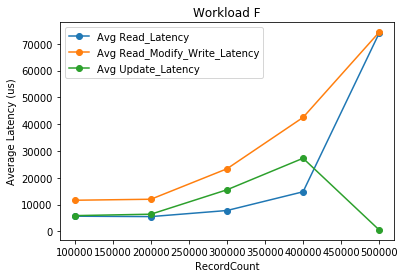


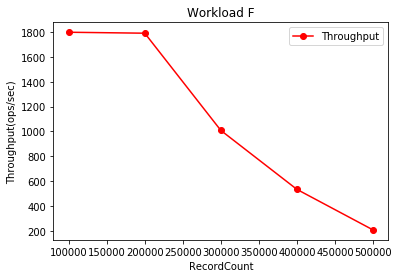
WORKLOAD E





WORKLOAD F





Conclusion:

* YCSB provides a standard to compare different databases on various factors such as total time taken, total ops, average throughput, min/max/average/(95/99)percentile latencies, etc
* Hence YCSB is a very effective benchmark framework that must be used by every developer when one decides to choose which DB to be used for what purpose in his application.
* Based on the above graphs obtained MongoDB should be used for:
  + E-commerce product catalog.
  + Blogs and content management.
  + Real-time analytics and high-speed logging, caching, and high scalability.
  + Configuration management.
  + Maintaining location-based data — Geospatial data.
  + Mobile and social networking sites.
  + Evolving data requirements.
  + Loosely coupled objectives — the design may change by over time.
* It should not be used for:
  + Highly transactional systems or where the data model is designed up front.
  + Tightly coupled systems