MONGODB
VS
COUCHBASE SERVER

Sreeja Matturu
Jithin Jacob Benjamin Jacob

MongoDB and Couchbase

- NoSQL Distributed database
- Document oriented
- Share-Nothing architecture
- Stores JSON type data

Introduction

- Taken two huge JSON datasets called as “Business” and “Tip” from ”Yelp” dataset
- Yelp dataset contains the information about different businesses in different cities and states.
- Business dataset has 192K documents that in it, while the Tip dataset had 1.2M documents in it.
- Imported the datasets into both the systems and tested on 1 master with 3 slave, 1 master with 1 slave and a single 1 master setup.

Installation of MongoDB

- Implemented both the systems using Amazon Web Services (AWS)

- Installed MongoDB Atlas, which is the cloud version of MongoDB with m5.a large EC2 instances

- Primarily created four clusters determining them as one master and 3 slaves

- Connected the MongoDB Atlas to the terminal and performed different queries within the terminal.

- The performance was evaluated from the MongoDB Atlas interface.

Couchbase Installation

- Couchbase Server Enterprise Edition and Couchbase Sync Gateway was installed on AWS EC2 instances

- For Good Replication and High Availability, 1 Couchbase sync gateway instances is required for 2 Couchbase servers

- Created a new cluster With 6 instances where 1 is the master and 3 are slaves and 2 systems are Couchbase sync gateway servers

- Primary design for the Couchbase is the one master with 3 slaves

- Dataset import was made by using cbimport through the terminal of the local machine

Hardware Specification

- 4 instances for MongoDB and 6 instances for Couchbase server taken from AWS

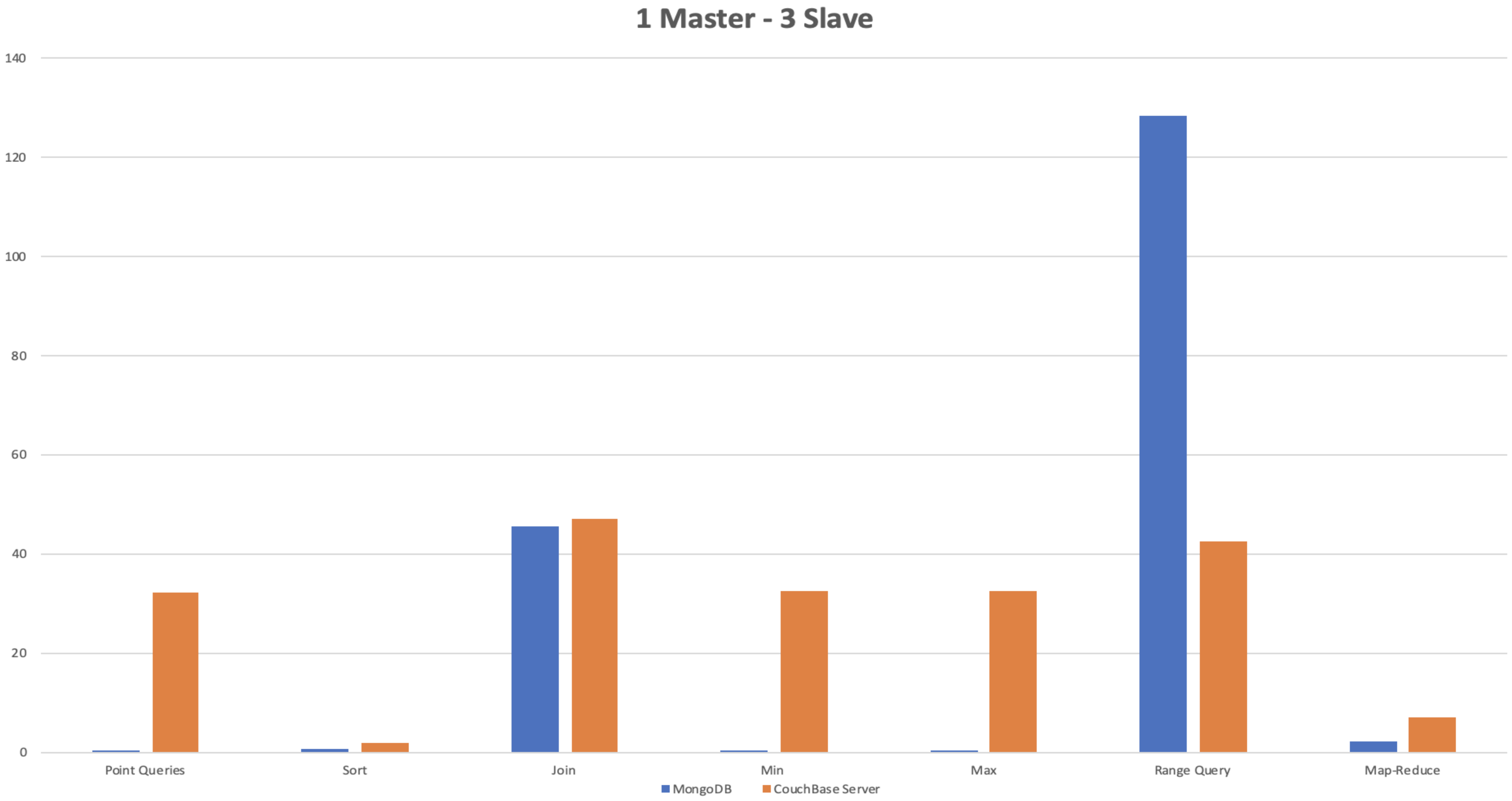
- Each instance has:
  - 8GB of RAM
  - 2-core vCPU – Intel Xeon Platinum 8000 series processor
  - Up to 10Gbps of Network Bandwidth
  - Elastic Block Storage bandwidth of 2120Mbps
  - All chosen from AWS us-east-1 (North Virginia)

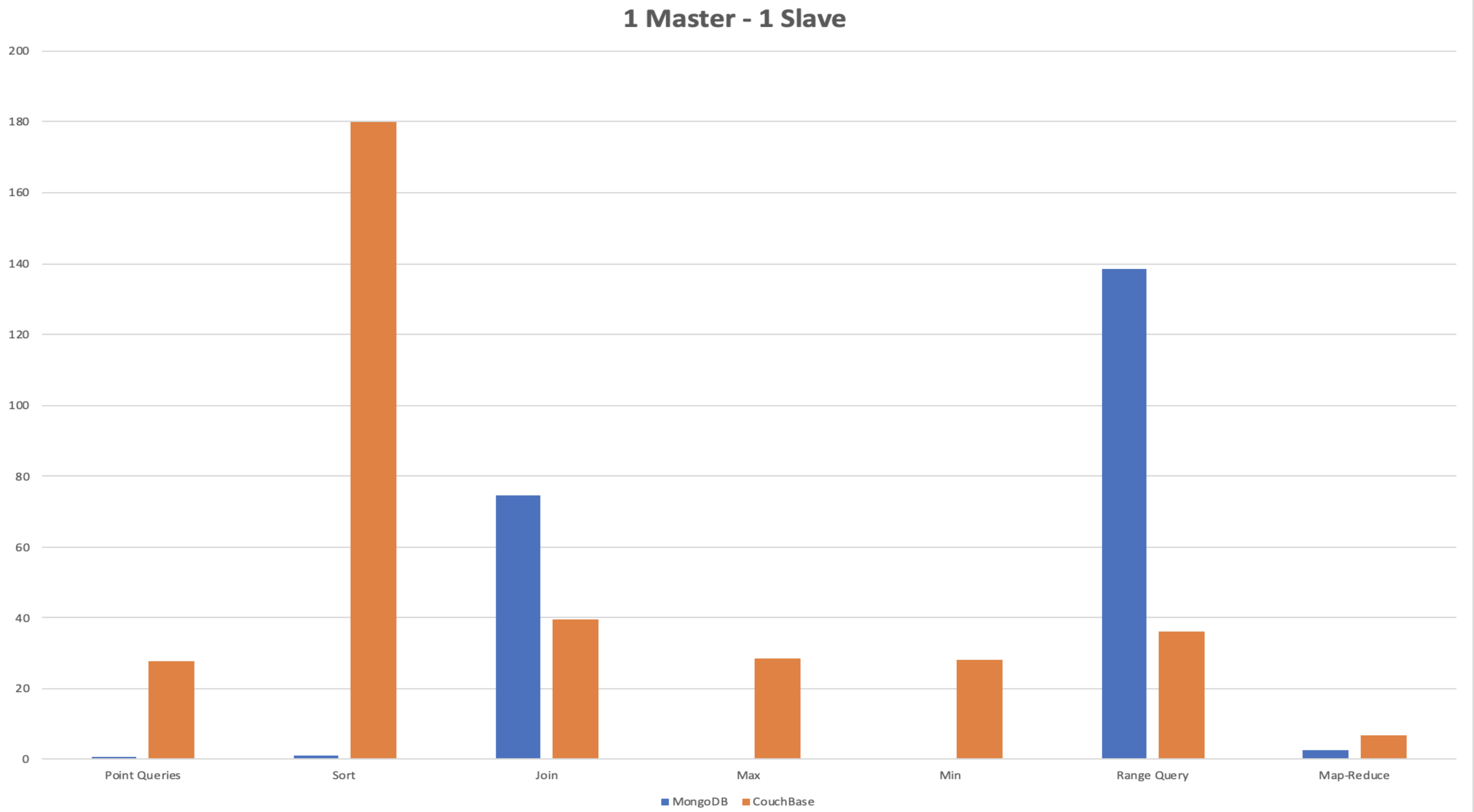
Query Scenario

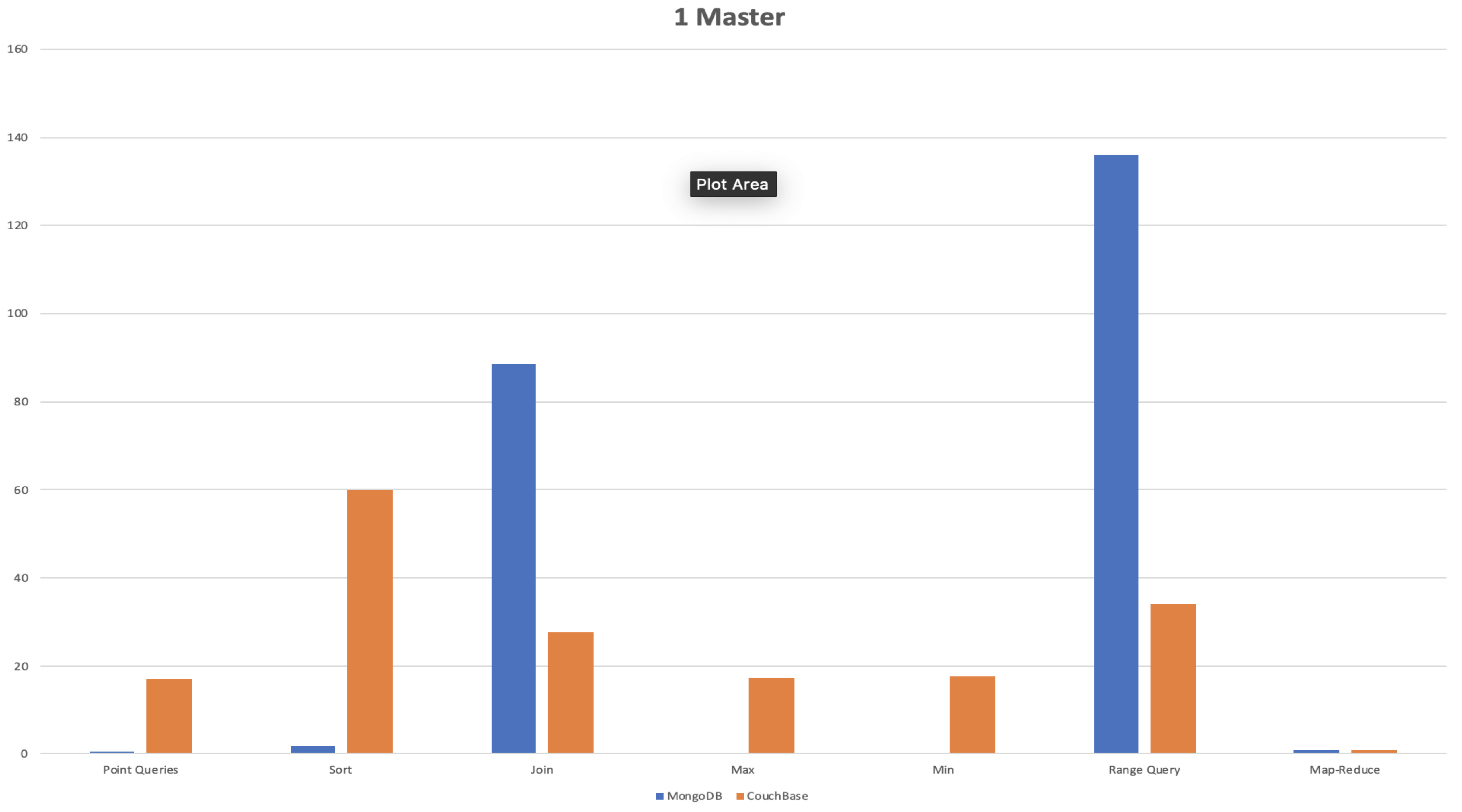
- Point Query to show the result of the name of businesses from the Business dataset that has the review_count>1000
- Count aggregation with sort to count each compliment_count value in the Tip dataset and sort them in ascending order.
- Join Business and Tip dataset using a unique ID called business_id.
- Min and Max aggregation operation to show the highest and lowest review_count from Business dataset.
- To calculate the review_count between 0 and 50 in the Business dataset.
- Map Reduce to calculate the review_count range between 0 and 50.

Query Performance Analysis

- The performance analysis for both the systems was made on:
  - One Master + Three Slaves
  - One Master + One Slave
  - Only One Master


1 Master - 3 Slave
140
120
100
80
60
40
20
0
Point Queries
Sort
Join
Min
Max
Range Query
Map-Reduce
■ MongoDB
■ CouchBase Server


1 Master - 1 Slave
200
180
160
140
120
100
80
60
40
20
0
Point Queries
Sort
Join
Max
Min
Range Query
Map-Reduce
■ MongoDB
■ CouchBase



1 Master

160
140
120
100
80
60
40
20
0
Plot Area
Point Queries
Sort
Join
Max
Min
Range Query
Map-Reduce
■ MongoDB
■ CouchBase

- How are multiple queries executed parallelly?
    - Allows multiple users to read and write the same data.
    - Uses Lock
- How are single queries parallelized?
    - Intra-operation Parallelism is followed by MongoDB
- What is the sort algorithm?
    - Uses In-Memory Sort algorithm
    - Uses Top k-sort algorithm
- How are Joins parallelized?
    - Intra Query parallelism
- Are query plans evaluated parallelly?
    - Multiple indexes are evaluated in parallel

Conclusion

- The cost for the Couchbase is higher when compared to MongoDB in AWS.
- From the query execution analysis, MongoDB is faster in point queries and aggregate functions but not the join operations.
- The performance for range queries can be increased by creating more indexes manually in MongoDB.
- In MongoDB, one master with three slave has higher performance than one with single master.

Thank you!!