
BENCHMARKING DATABASES USING APACHE JMeter

Aditi Mittal(amittal@umass.edu) Christopher Gomez(cdgomez@umass.edu)
Mehul Ramaswami(mramaswami@umass.edu)

ABSTRACT

For each of the business application, an in-depth research is required to find a suitable data store to serve business' technology needs within the required cost. In this project, we aim to compare the latency and throughput of lookups, insert and update queries for both relational and non-relational databases. We will be utilizing the three major database systems MySQL, SQLite and MongoDB for bench-marking using the Apache JMeter tool.

1 INTRODUCTION

Every business must focus on different metrics to decide the database they should use to fulfil the technical goals within the available cost. For example, if the focus of the system is on the latency then it should choose the database that provides lowest latency for the amount of data it would be processing along with the computation power it can afford. They should also consider which operations will be more frequent. This makes it important to analyze the performance of both relational and non relational databases on the same type of queries and datasets so that the metrics can be appropriately compared.

In one of the existing works, the authors (Rautmare & Bhalerao, 2016) have compared two systems - MySQL and MongoDB - on the IoT sensor data using JMeter. In other work, the author (Keshavarz, 2021) compared the runtime of MySQL and MongoDB by performing set of queries containing read, write, update, and delete operations on the same data. They concluded that MongoDB performed better than MySQL for most operations.

JMeter is an Open Source tool, widely used to measure performance parameters of an application. Our work focuses on providing more detailed analysis on the specific metrics of latency and throughput on two different type of publicly available datasets for the three database systems of MySQL, SQLite and MongoDB using the same JMeter tool. We will be using lookup, insert and update queries to compare their performances.

Our hypothesis is that MongoDB being a non-relational database, will perform better on both the datasets for most of the queries. We also expect MySQL and SQLite to perform somewhat similarly.

2 METHODOLOGY

Our first step would be to clean and pre-process both the datasets in order to get them into the ideal format for effi-

cient bench-marking with Apache JMeter. Next, we would integrate JMeter with each of our 3 database systems using the JDBC. We will be using Java language for this project. Then, we would run different queries such as insert, update and lookup on the the databases, and subsequently record the latency and throughput measures for each. Finally we would use these metrics along with the internal design to determine the strengths and weaknesses of MySQL, SQLite and MongoDB, and decide which performs the best.

3 EVALUATION

We will be using both latency and throughput as the metrics for the comparison of the three database systems (MySQL, SQLite, and MongoDB), using the Airbnb and Walmart datasets. The datasets will be discussed in details in the next section. This evaluation will primarily be performed using the Apache JMeter tool. We will be modifying the number of records to insert/lookup in bulk along with the number of threads that should be used by the system.

4 DATA

We will be using two publicly available datasets, namely the "Walmart Recruiting - Store Sales Forecasting" dataset available on Kaggle and the Airbnb dataset available on Airbnb's website(<http://insideairbnb.com/get-the-data/>). The Walmart Recruiting dataset doesn't have textual data, so to measure the metrics on more complex data(textual) as well, we will be using Airbnb dataset for further evaluation. Walmart dataset has 8191 rows with 12 columns. Airbnb dataset has 6894 rows and 75 columns.

We will be writing a script to initially add the complete data to the databases after proper pre-processing.

5 TASK-LIST

1. Clean and pre-process both datasets (All)

2. Integrate JMeter with MySQL (Mehul)
3. Integrate JMeter with MongoDB (Christopher)
4. Integrate JMeter with SQLite (Aditi)
5. Decide what queries should be run (All)
6. Record latency and throughput metrics for each database and query with varying number of records (1 person per database system)
7. Analyze pros and cons of three systems based on system design (1 person per database system)

6 TIMELINE & DELIVERABLES

At milestone, we plan to show the results and analyses of latency and throughput for look-up queries within the three databases for the Walmart Recruiting dataset. The dataset would need to first added in the databases and JMeter should be integrated for all the databases. The progress will be pushed on Github.

By the project presentation day, we plan to complete the detailed analysis on all three query types like multiple joins using both the Walmart and Airbnb dataset. It will involve analysis of the system design of the three systems and discussing their pros and cons, and how those factors potentially leverage into and explain the observed results. The final clean code will be pushed on Github.

REFERENCES

- Keshavarz, S. Analyzing performance differences between mysql and mongodb. 2021.
- Rautmare, S. and Bhalerao, D. M. Mysql and nosql database comparison for iot application. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pp. 235–238, 2016. doi: 10.1109/ICACA.2016.7887957.