Assignment 1 Reflection Report

Lukasz Filanowski 18414616

The clean.sh contains commands to run every separate script for Task 1. Most parts of Task 1 have been split up into their own script.

# SQL Queries

3.

1. SELECT author FROM user;
2. SELECT title, subreddit

FROM post

SELECT author

FROM user

WHERE post.author\_id = user.author\_id;

1. SELECT subreddit,

COUNT(created\_month)

FROM post

GROUP BY(subreddit);

# Table Creation commands in SQL

create table user(

author\_id smallint unsigned not null,

author varchar(60) not null,

author\_cakeday BIT,

primary key (author\_id)

);

create table subreddit(

subreddit varchar(60) not null,

primary key (subreddit)

);

create table post(

id smallint unsigned not null,

author\_id smallint unsigned not null,

subreddit varchar(60) not null,

created\_month date not null,

title varchar(60) not null,

primary key (id),

subreddit foreign key references subreddit(subreddit),

author\_id foreign key references user(author\_id)

);

# 1.

Bash has its strength in combining many powerful tools and utilities for handling big data. These tools include cut, sed, paste, grep and many more. Without the use of these, bash isn’t very applicable for handling large data sets as these commands are essential for handling big data.

# 2.

The NoSQL model is a far newer technology; therefore, it is more suited to more modern applications. The NoSQL system is also more suitable for hierarchal data storage as it follows the key-value pair method for data storage. It is better at storing and handling big data sets. However, the SQL model is more suited to high transaction-based applications. SQL also provides a lot of community and vendor support while NoSQL has a far more limited support. SQL also supports far more complex queries as it has a standard query interface. The NoSQL structure is much better for scalable databases as scaling in SQL is quite costly. New data insertion is also easier in NoSQL databases. In SQL databases, changes need to be made like altering schemas and backfilling data. Meanwhile in NoSQL, data insertion doesn’t require any prior steps. NoSQL offers only eventual consistency. SQL also offers high retrieval speeds when it comes to database records.

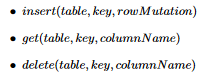
In conclusion, the NoSQL database system is far more effective at handling big data and offers great flexibility with data models, as it is table and schema free. Meanwhile, SQL is suitable for relational database models. It is code free and uses one standardised language across all RDBMs. It is also ACID compliant, offering good security and stability. SQL is suitable for database structures which don’t require big data handling.

# 3. Cassandra- A Decentralized Structured Storage System

The paper discusses a storage system solution called Cassandra. It is a distributed system which excels at managing large amounts of data that is decentralized and spread out across several servers. Cassandra aims to maintain its persistent state in the face of common failures, both large and small which at this scale are quite common. It provides a simple data model to the clients that focusses on dynamic control. The system is ideal for handling high data throughput at no expense to the read efficiency. The Cassandra system was designed and developed by Facebook in response to their specific needs and requirements to fulfil their service. Facebook is required to serve millions of users daily on thousands of servers in data centres all over the globe. The service must perform well and reliably but most of all must support continuous growth of the platform. Scalability was at the number one concern when designing Cassandra. With failures occurring regularly in such a vast and complex system, Cassandra had to treat failures as the norm rather than an exception.

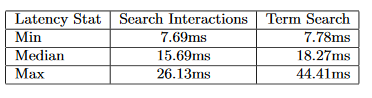
When designing Cassandra, the developers looked at many already existing systems to study their strength and limitations, especially under the criteria of availability, durability and distributing data for performance. Some systems like Ficus and Coda replicated files which led to high availability, but this was done at a cost to consistency. Farsite was a big influence on the development of Cassandra as it doesn’t rely on any centralized server. It also has very high availability and scalability potential. One limitation of this system was that it didn’t allow for disconnect operations and wasn’t resilient to outages or faults. Other systems like Google’s distributed file system GFS (Google File System) were looked at. GFS uses one master server where all the metadata is stored and where the data is split into chunks and stored on separate servers. However, this model required a separate fault toleration system called Chubby. Amazon’s Dynamo system as also examined. Dynamo is used by Amazon to store and process the contents of user’s shopping carts. The system had many desirable traits, such as allowing for read and write operations to continue during network partitions. It was described as having the most one-hop request routing. It handles update conflicts through a vector clock scheme but also facilitates client driven resolution. However, Dynamo was limited in providing high write throughput due to it requiring read to perform vector timestamps.

Hence, with all of this in mind, the development of Cassandra could begin. A table in the Cassandra system is a distributed multidimensional map which is indexed by a key. The key has no size restrictions and each value is an object. Cassandra has 2 types of column families, Simple and Super, the latter able to be displayed as a column family within a column family. The system allows for columns to be stored by both time and name; a feature required for implementation in the Inbox Search where results are displayed in time sorted order. The API consists of 3 methods:



The Cassandra system can scale incrementally by partitioning the data dynamically. This is done using hashing. Each partitioned node is given a random value that represents its position within the ring. It also uses replication for high availability and durability. The aforementioned coordinator that is assigned to a node is in charge of replication of data that are within its range. Cassandra offers various options for data replication such as “Rack Unaware”, “Rack Aware” and “Datacentre Aware”. Cluster membership in Cassandra is a Gossip based mechanism called Scuttlebug. It has efficient utilization of the CPU and gossip channel. Upon a new node joining the ring, it is assigned a random position. This information is then gossiped around the ring and this way, each node now knows its new respective positions within the ring. Failure detection in Cassandra is done in such a way that a node can determine locally if any other given node is up or down. The failure detection module produces a suspicion level for each suspected node instead of a Boolean to determine if the node is up or down. The data persistence is ensured by a local fie system. Data is represented on disk using a highly efficient data retrieval format.

When the system was being developed, implemented and maintained, some interesting results were found. The Map/Reduce jobs that were run on the MySQL data actually behaved as a client of Cassandra. The time to detect failures grew to an unacceptable number as the size of the cluster increased. In a cluster of 100 nodes for instance, it took over 2 minutes to detect a failed node. The Cassandra system was employed for the Inbox Search feature, which maintains a per user index of messages exchanged between the sender and recipient. When the user clicks into the searchbar, a message is sent to the Cassandra cluster to prepare the buffer cache with the index of the user. This way, the search time is reduced and likely, the search result will already be in memory.



In conclusion, the Cassandra system is a very efficient, scalable and applicable system for its designed purpose. It can very easily be used in many other applications due to its simplicity and versatility when it comes to big data management. The paper was laid out in a very structured and easy to follow way, providing ample detail on each topic discussed. The paper could have provided more statistical and mathematical evidence for their findings. Overall, the paper does an adequate job of describing the design process and the power of the Cassandra system.