Group members:
ANOUAR ZOUAD MCS211049,
CHONG XIAN JUN MCS211047

Question 1 (10 marks)

Read articles discusses on CAP, ACID and BASE theorems provide in the e-learning. Discusses the how component consistency is handled in these theorems. Discusses ***the consistency issue relate the distributed and centralised architecture of data management****. Discuss* ***two applications that suits to ACID and BASE properties***

In the context of CAP theorem raised by Eric Brewer in 2000, consistency is a standard that can be measured by whether all client nodes in a distributed system see the same, latest data.

In a **centralized database architecture**, all clients are directly accessible to the database in the central server. As all read and write functions are executed at a consistent, known order on the same cdatabase, it thus allows for a very high consistency.

Meanwhile in a **distributed database architecture**, distributed systems replicate data across a large number of storage devices to protect the system against storage failure and allow for acceptable data availability level. This results in an issue in defining consistency of a distributed system, which could be answered through quorum protocol [1].

In a distributed system, each replica stores the state for each item, and assigns a version number on it, i.e.. v1,value or v2,value. The larger the version number, the more recent is the state associated. This allows a read function to recognize the latest update [2].

Within the distributed system,
N = total replicated storage units in the system.
W = number of replicas that write commands act on.
R = number of replicas that read commands act on

- **(W + R) > N: the system is strongly consistent**. As (W + R) > N, it assured that when a write sends a new value to a group of nodes, a read can detect the updated value as the group of nodes that the read sends to intersects with that of the write command.

- **(W+R) <= N isweakly consistent**. A write could send a new value to one group of nodes that is completely missed out by its subsequent read as they might not intersect. [1]

Getting back to distributed database system, we are all well-aware that it is limited by CAP theorem that allows only two out of its three properties [3], which implies that a distributed database can only be
- consistent and partition-tolerant (CP), attributed to ACID

- available and partition-tolerant (AP). attributed to BASE

atomicity, **consistency**, **isolation**, **and durability** (**ACID**) is usually executed in relational database management systems like MySQL, PostgreSQL, Oracle, SQLite, and Microsoft SQL Server. Along with development of big data, more and more NoSQL DBMSs, such as Apache's CouchDB or IBM's Db2, have developed to possess a certain degree of ACID compliance too.

In a very large scale distributed database system, ACID compliance becomes very costly and impractical. Thus, in the space of big data management, the commonplace practice is **basically available**, **soft state**, **eventually consistent** (**BASE**). BASE principles are usually implemented in NoSQL databases like MongoDB, Cassandra and Redis are among the most popular NoSQL solutions, together with Amazon DynamoDB and Couchbase [4].

References:

1. Thain, D. (2016, March). *Cloud Computing - Notes on CAP Theorem*.
2. A Note on Quorum Consensus (n.d.) Retrieved from: A note on Quorum Consensus (mit.edu)
3. Gilbert, S., & Lynch, N. (2012). Perspectives on the CAP Theorem. *Computer*, *45*(2), 30-36.
4. *ACID Model vs BASE Model For Database*. (2022, February 10). GeeksforGeeks. Retrieved from: https://www.geeksforgeeks.org/acid-model-vs-base-model-for-database/

Question 2 (10 marks)

*Explain how Mapreduce works **in a query for a particular application**. Describe the application, include **data sample, output/input on each map/reduce phase** (i.e key/value) in your explanation.*

Example
1. Application: Web Analytic Find the number of URL in web logs Find where a page link
2. Application: e-commerce Find the number of order on certain item Find average daily sale obtain by each company from start date – to end date

UTM library makes use of MapReduce Framework to manage its library collection. In an attempt to optimize its database collection, it conducts an analysis on the count of times that students access each database.

UTM library thus makes use of web logs to access the number of times each subscribed database is accessed by students. In the MapReduce Framework, the query is divided into Map phase and Reduce Phase.

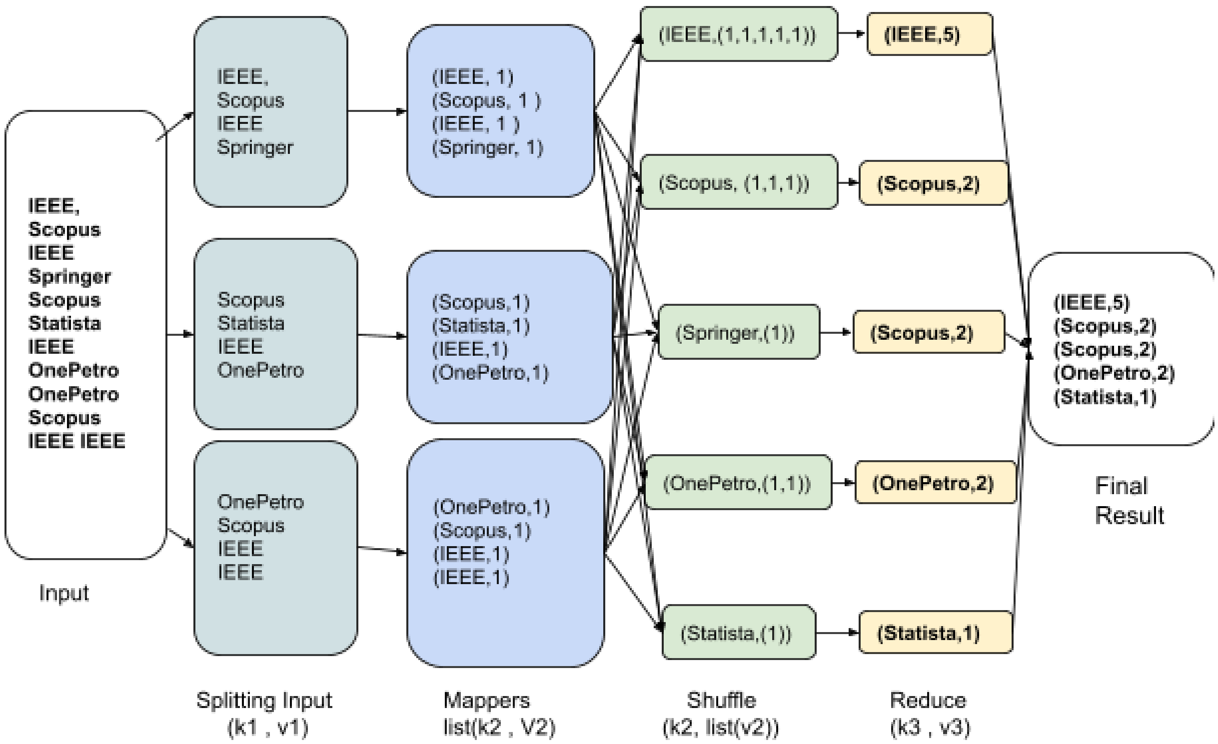
In order to simplify the illustration, **data sample** used in this example will only include parts of the UTM library database subscription, i.e. (IEEE, Scopus, IEEE, Springer, Scopus, Statista, IEEE, OnePetro, OnePetro, Scopus, IEEE, IEEE)

MapReduce consists of two main phases the Map phase and Reduce phase:

At the start, input web logs URLs are split into chunks that each consists of a list of values (web logs). At the same time, a cluster of machines consisting of a master program and many worker programs are activated. The master picks idle worker programs to assign a Map task or a Reduce task to each of them.

Map phase: The worker programs that are assigned with the Map tasks process their corresponding input split data. It then assigns the value of 1 to each data and parses key/value pairs (k2, v2), e.g. (IEEE, 1) out of these web logs. The intermediate key/value pairs produced by the Map function are then buffered in memory.

Reduce phase: The master then notifies the Reduce workers about the locations where these intermediate key/value pairs (k2, v2) are deposited. The Reduce workers then remotely call to read this buffered data from the local disks of the map workers. When a reduce worker has read all intermediate data, it executes a shuffle process to sort and aggregate them by the intermediate keys. Consequently, it generates a new output in the form of (k2, list(v2)), e.g. (IEEE, (1,1)). This data is then consolidated into the final output of the Reduce phase in the form of (k3, v3), e.g. (IEEE, 2) that consists of the queried info: The number of times that students access each database. In the illustration displayed below, the query gets the output of (IEEE,5) (Scopus,2) (Scopus,2) (OnePetro,2) (Statista,1)



References:

1. Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified data processing on large clusters.
2. Sinha, S. (2016, Autumn 11). *Fundamentals of MapReduce with MapReduce Example*. Medium.Com. Retrieved from: [Fundamentals of MapReduce with MapReduce Example | by Shubham Sinha | Edureka | Medium]