Data Replication

Refers to the “synchronization” of data in a distributed database. Increased data redundancy and availability is also maintained across the multiple databases since data is supported through each of the database. It is also considered “Fault tolerant” since a loss of a single server would not result into suspensions or disruptions of the distributed database.

Distributed Database

A distributed database is a type of database that does not have a single storage in a specific device. It is a database that stores its data on multiple devices (or even in a network). Communication from a single database to others is enabled by the “computer network”. This type of database is managed by a software called distributed database management system (DDBMS). One of the features of a DDBMS is having the ease of doing CRUD (create, retrieve, update, delete) in the distributed database. Also, updates (from the changes made in the database) are “synchronized” by the DDBMS to ensure data integrity for the data inside (Retrieved on October 25, 2016/ https://www.tutorialspoint.com/distributed\_dbms/distributed\_dbms\_databases.htm). Below is/are example/s of a DDBMS:

Apache Cassandra

From The Apache Software Foundation, Cassandra is an open source DDBMS that was made to manage “very large” data among a distributed database with “…a highly available service with no single point of failure…” (Lakshman & Malik, 2008). This DDBMS uses NoSQL. It is a non-relational database which “scales out” relational databases (data is found on multiple device/ nodes contrary to the other one that stores it in a single device/node) and is recommended for databases managing large amounts of data (Retrieved on October 25, 2016/ https://www.youtube.com/watch?v=qUV2j3XBRHc). One of Cassandra’s features is that it is “Decentralized”: Each node in a network has the same function. Also, Cassandra “Supports replication and multi data center replication” since it is a DDBMS, “Fault tolerant” since the data is “…automatically replicated to multiple nodes…”, It also has its own Query Language named Cassandra Query Language (CQL) (Retrieved on October 26, 2016/ http://cassandra.apache.org/ and http://immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120911A.pdf).

CockroachDB

From Cockroach Labs. This DDBMS aims to be fault-tolerant by having “minimal latency disruption” and having no “manual intervention” during some failures (from nodes, datacenters, etc.). Unlike Cassandra, CockroachDB still supports the ACID semantics, has its transactions distributed over the whole database. Also, it uses SQL rather than the NoSQL (which specializes in distributed databases) (Retrieved on October 26, 2016/ https://github.com/cockroachdb/cockroach/blob/master/docs/design.md).