

Assignment: 01 (Key-Value Store)

By

Turzo Ahsan Sami

ID: 20166012

MSc. In CSE, BRAC University

Date: 22 January 2020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted in partial fulfillment of the module

CSE-707: Distributed Computing System

To

Mr. Md. Safiuzzaman Rajib

CSE Department, BRAC University

**Key-Value Store / NoSQL**

Distributed Database Systems: A distributed database is a collection of multiple interconnected databases dispersed at different autonomous computer systems connected by a network. Databases in the collection are logically interrelated and represent a single logical database. Data is physically stored across multiple independent systems. Distributed DBMS (DDBMS) is a centralized software system that manages a distributed database as if it were a single database. DDBMS is designed for heterogeneous database platform.

NoSQL (Not Only SQL) Database: NoSQL databases are non-relational databases. Concept of *Table* is absent in NoSQL. NoSQL is a flexible database system used for big data and real-time web apps. Conventional Relational Database Management Systems have a rigid structure which can not be altered at different specifications hence can not be used in distributed database systems. NoSQL handles big data, predefined schema and unstructured data. Horizontal scaling is easier with NoSQL.

NoSQL encompasses a wide variety of different database technologies most prominent of which are – Document oriented, Graph database, Column oriented and Key value store.

Key-value store: These databases are organized as key value pairs, where each key appears exactly once.

Key value stores are designed to store, retrieve and manage dictionary or hash tables. Dictionaries contain a collection of objects or records which have multiple fields within them containing data. The records are stored and retrieved using a unique key. Key-value systems treat data as a single opaque collection, which may have different fields for each record as opposed to relational database systems where data is stored in predefined tables. Hence, key value stores provide flexibility and closely follows modern concepts like object-oriented programming. Since optional values are not represented by placeholders or input parameters, key value stores require less memory to store the same dataset which leads to performance gains in certain workloads. Key value databases provide high performance, scalability, and flexibility.

Example of key-value databases are Couchbase, Dynamo, Apache Ignite, Arango DB, Redis, Oracle NoSQL etc.

REDIS (Remote Dictionary Server): Redis is a key value store. Redis does not have any conventional tables, rows, columns functions, procedures etc. Redis does not use query language such as SELECT, INSERT, UPDATE, DELETE etc. Redis uses data structures such as String, Lists, Sets, Hashes, Bitmaps, Hyperloglogs and Geopatial inexes to store data. Redis utilizes commands such as GET, SET to interact with data. Redis is an in-memory database hence it is superfast. Redis can replicate data ins any number of slaves. An efficient configuration of Redis master-slave model is where the master is configured as ‘write-only’, one of the slaves configured as ‘read-only’ and the other slave rites data to disc asynchronously. If master goes down, request will be served from slave therefore providing zero-downtime. Redis is single threaded and utilizes pipelining.

Example of a Redis record is as follows.

{

Name: “John”,

Age: 26

}

Here (Name, Age) are keys values of which are (“John”, 26).