**CP2285 Lab #2 Part 1**

**Adam Barry**

**January 25th, 2023**

9) What are the key assumptions made by the Hadoop Distributed File System approach?

The key assumptions that are made by the Hadoop Distributed File System approach are the following

1) High Volume – The volume of data in big data applications expected to be the terabytes, petabyte or larger. Hadoop assumes that files in HDFS will be extremely larger. Data in HDFS is organized in physical blocks just as in any other type of file storage. As a result of the number of blocks profile is greatly reduced simplified method of overhead trafficking one file.

2) Write-once, read-many - using a read-once read-many models simplifies concurrency issues and improves overall data throughput. Using this model of file is created, written to the file system and has been closed. Once the files are closed, changes cannot be made to its contents. This improves overall system performance and works well for types of tasks that are performed by big data applications.

3) Streaming Access – different transaction processing systems where queries after receiving small pieces of data from several different tables, big data applications typically process entire files. Instead of optimizing the file system to randomly access individual data elements a dupe is optimized for batch processing of entire files as a continuous out stream of data. This is called Streaming Access.

4) Fault Tolerance - a dupe is designed to be distributed across thousands of low-cost computers. Many devices at any point in time can experience some hardware errors. This is where HDFS is designed to replicate data across many different devices so that if one device fails the data is still available and backed up on another device.

10) What is the difference between a name node and a data node in HDFS

The difference between a name node and a data node in HDFS is the Data nodes store the actual data within the HDFS which is broken up into blocks and are duplicated to ensure fault tolerance. Whereas Name Nodes contain the meta data for the file system there is only one name node in a HDFS cluster. The metadata is made to be smaller and simpler to be recoverable if needed. Keeping the metadata smaller allows the name node to hold all of the important metadata in the memory.

11) Explain the basic steps in MapReduce processing.

The basic steps in MapReduce processing are the following…

* Take a complex task and break it down into a collection of smaller subtasks.
* Once all the subtasks are preformed at the same time combine the results of each subtask to produce a final result.
* A map function takes a collection of data and sorts the data by a key-value pairs the map is preformed by the mapper.
* The reduce function takes the collection of key-value pairs and summarizes them into single results. The reduce function is preformed by the reducer.

12) Briefly explain how HDFS and MapReduce are complementary to each other.

HDFS and MapReduce are complementary to each other because the HDFS structure is composed of a name node as several data nodes. the map reduce uses a job tracker several other task chasers. the job tracer acts as a central hooked role for the mat produced processing and normally exists unsafe server acting like a name node. Task tracker program reside on the data nodes however now tracker will take care of locating data terminating which node to use dividing the job task for nodes and managing failures of all nodes.