Report on MapReduce - Simplified Data Processing on Large Clusters

MapReduce is a programming model developed by Google, it is used to generate and process large data sets. The MapReduce is designed in such a manner that the system automatically parallelizes tasks and seamlessly integrate and utilize distributed system resources. The MapReduce sum up the parallelization, data distribution, load balancing and fault-tolerance in its libraries. The "map" operation processes a key-value pair to generate intermediate key-value pairs while the reduce operation combines all the values with the same key. MapReduce is used in many scenarios such as to count the URL access frequency, term-vector per host, reverse-vector per host, distributed sort, Inverted Index, and distributed grep. The MapReduce is used for clustering, large-scale ML problems, extraction of data for processing queries, extraction of features and properties of web pages for a new experiments product and large-scale graph computations. One of the major conclusions of MapReduce is that Large-Scale Indexing, where the system takes huge sets of documents as an input and then it is stored in the files of Google File System. In their experience, the authors have expressed that parallelization and distributed computing are made easier by restricting the programming models and makes it fault tolerant. In order to handle fault tolerance, the master node pings every slave node periodically in case when no response is received from a slave node over a certain period of time, then the master node flags that particular slave node as a failure node. Furthermore, the system is optimized to avoid reduced traffic across the networks. Some of the major drawbacks in context for this paper is that MapReduce does not works efficiently on smaller datasets. The impact of slow performing machines was reduced, and the system was able to handle data loss and machine failures.

Report on The Google File System

Google has developed the Google File System as a distributed file storage, so that it allows the read and write operations on the files which are distributed across multiple servers. For the Google File System, Google used commodity hardware instead of expensive servers. The Google File System (GFS) is further optimized for large files to store and read them whenever it is needed. One of the remarkable features of the GFS is that the write operation on file is generally works as an append operation, whereas there is no random write operation on file. While the read operation on file are mostly works as sequential reads. Hence, a crawler only appends the data when it's combined to a single file, further the batch processing system creates a search index for this file. Each file can be in Gigabytes, but they are split into multiple chunks, where each chunk size is 64MB to 100MB and then these chunks are stored on chunk server. The GFS ensures that multiple copies are created in different servers in case one chunk is corrupted or the whole chunk server fails or gets crashed, these data can be retrieved from the other chunk servers. There is a high chance of failure in hard disk and chunk servers as they are based on commodity hardware, therefore tends to be less reliable but they are effective in the aspect of cost factor. GFS master server is a single server which stores all the metadata of all the clusters such as filenames, chunk ids, chunk location and access their control details. In order for master server to make sure that the chunk servers are active, the chunk servers sends heartbeat messages regularly. The earlier phase of GFS has a workload bottleneck issues which has been resolved in the current GFS by changing master data structures in order to allow binary searches more efficiently. All the file operation are recorded in an append only logs as a checkpoint when servers fail. In a scenario when the master server fails too, through the DNS server, the shadow master server with all the replicated data and the log files as the master server will take over the operations. Therefore the GFS is reliable and supports largescale data for processing on commodity hardware. Also, GFS storage are reliable because whenever the data gets corrupted it can detect and recovers very efficiently.

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

Signature: Janmejay Mohanty Date: 14th February 2022