e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

MAP REDUCE IN HADOOP

Mercy Ruth Mahesh Daniel*1

*1SIWS College, Mumbai, Maharashtra, India.

DOI : https://www.doi.org/10.56726/IRJMETS32932

ABSTRACT

Vast Volumes of data are generated in today's data-driven market due to algorithms and applications constantly gathering information about individuals, businesses, systems and organizations. The tricky part is figuring out how to quickly and effectively digest this vast volume of data without losing insightful conclusions. A software framework and programming model called Mapreduce is used to process enormous volumes of data. To speed up processing , MapReduce executes logic on the server where the data already sits, rather than transferring the data to the location of the application or logic. In the end, it collects all the information from several servers and gives the application a consolidated output.

I. INTRODUCTION

MapReduce is a framework used by Google for processing huge amounts of data in a distributed environment and Hadoop is Apache's open-source implementation of the MapReduce framework. MapReduce is widely used for not only commercial applications but also scientific computations. Facebook uses a Hadoop cluster composed of hundreds of nodes to process terabytes of user data. Hadoop is a distribute computing platform written in Java. It incorporates features similar to those of the Google File System and of MapReduce. MapReduce is developed in a way that scales to large clusters of machines comprising thousands of machines. This creates efficient use of these machines resources and therefore is suitable for use on many of the large computational problems.

II. LITERATURE SURVEY

The performance of a parallel system like MapReduce system closely ties to its task scheduler. The current scheduler in Hadoop uses a single queue for scheduling jobs with a FCFS (First Come First Serve) method. Using this scheduler people can assign jobs to queues which could manually guarantee their specific resource share. Different kinds of jobs often simultaneously run in the data centre. These different jobs make different workloads on the cluster, including the I/O bound and CPU-bound workloads.

1. Definition of MapReduce

MapReduce is programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate keys.

2. Working of MapReduce

Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. A typical MapReduce computation processes many terabytes of data on thousands of machines. A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically, both the input and output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks. Typically, the compute nodes and the storage nodes are the same, that is, the MapReduce framework and the Hadoop Distributed File System are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks of the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master. Applications specify the input/output locations and supply map and reduce functions via implementations of appropriate interfaces and abstract classes. These, and other job parameters, comprise the job configuration. The Hadoop job client then submits the job (jar/executable etc) and configuration to the

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

3. Inputs and Outputs

The MapReduce framework operates exclusively on <key, values> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types. The key and value classes have to be serializable by the framework and hence need to implement the Writable interface. Additionally, the key classes have to implement the Writable Comparable interface to facilitate sorting by the framework.

4. MapReduce – User Interfaces

MapReduce User interface -provides a reasonable amount of detail on every user-facing aspect of the MapReduce framework. This is to help the users to implement, configure and tune their jobs in a fine-grained manner. The Javadoc for each class/interface remains the most comprehensive documentation available.

Mapper and Reducer interfaces – Applications typically implement them to provide the map and reduce methods. These forms the core of the job.

Mapper - Mapper maps input key/value pairs to a set of intermediate key/value pairs. Maps are the individual tasks that transform input records into intermediate records. All intermediate values associated with a given output key are subsequently grouped by the framework, and passed to the Reducer to determine the final output

Reducer-Reduces a set of intermediate values which share a key to a smaller set of values. Reducer has 3 primary phases: shuffle, sort and reduce.

Shuffle - Input to the Reducer is the sorted output of the mappers. In this phase the framework fetches the relevant partition of the output of all the mappers, via HTTP. Sort – The framework groups reducer inputs by keys. The shuffle and sort phases occur simultaneously.

Reduce – In this phase the reduce (Writable Comparable, Iterator, Output Collector, Reporter) method is called for each <key, (list of values)> pair in the grouped inputs. The output of the reducer is not sorted.

Partitioner-Partitions the key space. Partitioner controls the partitioning of the keys of the intermediate map-outputs. Hash Partitioner is the default Partitioner.

Reporter-Reporter is a facility for MapReduce applications to report progress, set application-level status messages and update Counters. Mapper and Reducer implementations can use the Reporter to report progress or just indicate that they are alive.

5. Architecture of MapReduce

MapReduce architecture consists of various components.

Job: This is the actual work that needs to be executed or processed.

Task: this is a piece of the actual work that needs to be executed or processed. A MapReduce job comprises many small tasks that need to be executed.

Job Tracker: This tracker plays the role of scheduling jobs and tracking all jobs assigned to the task tracker.

Task Tracker: This tracker plays the role of tracking tasks and reporting the status of tasks to the job tracker.

Input data: This is the data used to process in the mapping phase.

Output data: This I the result of mapping and reducing

Client: This is a program or Application Programming Interface (API) that submits jobs to the MapReduce.

Hadoop MapReduce Master: This plays the role of dividing jobs into job-parts.

Job-parts: These are sub-jobs that result from the division of the main job.

6. Phrases of MapReduce

The MapReduce program is executed in three main phases: mapping, shuffling, and reducing. There is also an optional phase known as the combiner phase.

Mapping phase-This is the first phase of the program. There are two steps in this phase: splitting and mapping. A dataset is split into equal units called chunks (input split) in the splitting step. Hadoop consists of a Record

Reader that uses TextInputFormat to transform input splits into key-value pairs. The key-value pairs are then used as inputs in the mapping step. This is the only data format that a mapper can read or understand. The mapping step contains a coding logic that is applied to these data blocks. In this step, the mapper processes the key-value pairs and produces an output of the any form(key-value pairs).

Shuffling phase-This is the second phase that takes place after the completion of the Mapping phase. It consists of two main steps: sorting and merging. In the sorting step, the key-value pairs, are sorts using the keys. Merging ensures that key-value pairs are combined.

The shuffling phase facilitates the removal of duplicate values and the grouping of values. Different values with similar keys are grouped. The output of this phase will be keys and values, just like in the mapping phase.

Reducer phase-In the reducer phase, the output of the shuffling phase is used as the input. The reducer processes this input further to reduce the intermediate values into smaller values, it provides a summary of the entire dataset. The output from this phase is stored in the HDFS.

Combiner Phase- This is an optional phase that's used for optimizing the MapReduce process. It's used for reducing the pap outputs at the node level. In this phase, duplicate outputs from the amp outputs can be combined into a single output. The combiner phase increases speed in the Shuffling phase by improving the performance of Jobs.

III. CONCLUSION

As the Internet scale keeps growing up, enormous data needs to be processed by many Internet Service Providers. MapReduce is now becoming a leading example solution for this. MapReduce is designed for building large commodity cluster, which consists of thousands of nodes by using commodity hardware. In this Environment, many people share the same cluster for different purpose. This situation has led to different kinds of workloads running on the same datacentre. The MapReduce programming model has been successfully used at Google, Yahoo and several other companies for different purposes. MapReduce is easy to use, even for programmers without experience with parallel and distributed systems, since it hides the details parallelization, fault-tolerance, locality optimization and load balancing. A large variety of problems are easily expressible as MapReduce computations. For example, MapReduce is used for the generation of data for Google's production web search service, for sorting, for data mining, for machine learning and many other systems.

IV. REFERENCE

[1] HTTPS://WWW.CSE.SCU.EDU/~TWANG1/STUDENTPROJECTS/MAPREDUCE_CLUSTER_13S.PDF
[2] HTTPS://GRUT-COMPUTING.COM/HADOOPBOOK.PDF
[3] HTTPS://WWW.SECTION.IO/ENGINEERING-EDUCATION/UNDERSTANDING-MAP-REDUCE-IN-HADOOP/
[4] HTTPS://HADOOP.APACHE.ORG/DOCS/R1.2.1/MAPRED_TUTORIAL.HTML
[5] HTTPS://WWW.SPICEWORKS.COM/TECH/BIG-DATA/ARTICLES/WHAT-IS-MAP-REDUCE/