Abstract

MapReduce is a programming model which is used to process the large amount of data sets. MapReduce is a innovation and a good idea for general purpose computations. In this paper, we will discuss about the brute-force approach used by MapReduce instead of using Indexing like other RDBMS tools. This paper explores the advantages and the disadvantages of MapReduce and comparison with the relational database systems. From a book, Hadoop The Definitive Guide by Tom White defines MapReduce as, MapReduce has an ability to run an ad hoc query against the whole dataset and the result can get in a reasonable time is very Transformative.

Introduction

MapReduce is a batch query processor. It is programmed in such a way that it takes the set of records and split the job into two stages i.e. Map and Reduce. As name suggest, Maps takes the whole set of records and splits into the subparts and the output of the Map is the Input of the Reducer i.e. all the subparts of the records(Datasets) and all these subparts sends into different machines so that the time taken to complete the job will reduce because all the subparts will work parallelly and the last phase will be reducer will combine all the reduced data ad squish it into one combined result. In my words, this concept worked on Divide and rule concept. It is not a relational application, MapReduce just call all the set of records. In MapReduce, the run time system takes care of portioning the data into subparts and scheduling the program’s execution across different number of machines.

Model

MapReduce takes input as Key/value pair and gives result in the same format as key/Value pair. It works in two phases:

First Phase:

At first, it takes the input as Key/Value pair and produces the intermediate key/values pairs as subparts. In general, Multiple instances are running in Map function on different nodes of a compute cluster and it uses the same hash function and hence all the output files will be corresponding of these same hash function.

Second Phase:

In second phase, the output of the Map function is the input of the Reduce function. The output records of the map function used the same hash function to spilt the whole data and no matter which map instance produced them, it will be consumed the same reduce instance. M/R is great for large amount of data and it is feasible to run the hole dataset set one shot.

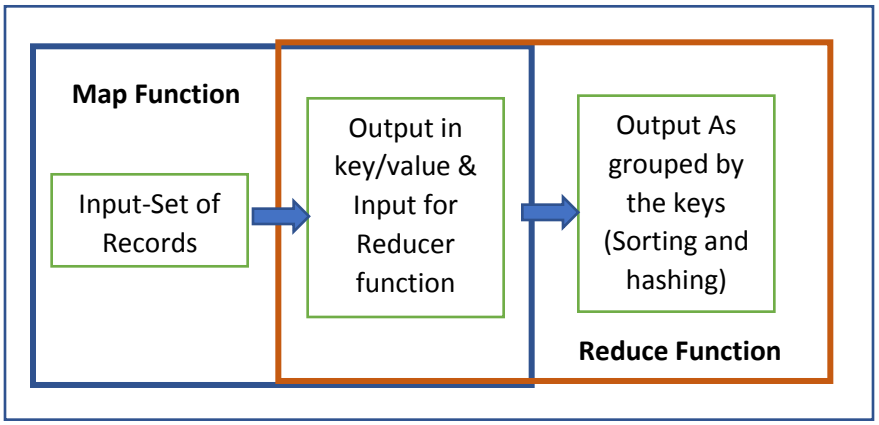

Figure 1 MapReduce Framework

Comparison with Relational Database tool

With the help Output of the MapReduce function, we cannot get any clue about the initial points or initial processed data because MapReduce squished all the Dataset at once. The biggest

innovation with MapReduce is that it works on large cluster data parallelly and takes lesser amount of data while Relational Database Management system cannot work parallelly and mainly Relational Databases does not depends on the type of the data while MapReduce is a Primitive type and suboptimal type, it depends on the type of the data. MapReduce is not an alternative of any relational Databases, it is own a kind which works for clustering of large amount of data and processed at one shot.

To avoid the operating system bugs, human errors and the failures of disks, memory, connectors, networking and power supplies Google using MapReduce but this is not an alternative of any Relational databases.

Challenges

In the paper, MapReduce- A major step backwards by David De Witt claims that MapReduce is a step backwards in databases access. Because RDBMS works using schemas while MapReduce used brute-force concept. Relational Databases works with the schema, it has a plan decided before starting any job and it is the best way to keep an application from adding garbage to dataset. But MapReduce has no such functionalities, it works on all the datasets at once.

Second point: Poor Implementation-

MapReduce is automatically providing parallel execution on grid of computer. I fell it is not having any poor implementation because it gives high performance, commercial grid oriented

engines. The problem occurs in map reduce when there is wide variance into the distribution of records with the same key. It used push method instead of using pull all the data sets.

MapReduce has missing features: MapReduce works with brute force instead of using indexing and because of that there is no clue about the initial points or history about processed data. And we must perform the whole method to update the same datasets which is time consuming if we must have to do same operations for the same data again and again. It is not a database application, it is used for massive amount of data to processed all at once.

It is incompatible with DBMS tools: Databases-oriented tools and processes don't work with/for MapReduce programming. These two are different things and work independently.

Conclusion

The MapReduce programming model has been effectively utilized at Google for various purposes. The following reason to accomplishment of large amount of data cluster: In the first place, the model is anything but difficult to utilize, notwithstanding for developers without experience with parallel and dispersed frameworks, since it conceals the subtle elements of parallelization, adaptation to non-critical failure, territory streamlining, what's more, load adjusting. Second, a vast assortment of issues is effectively expressible as MapReduce calculations.

For instance, MapReduce is utilized for the era of information for Google's creation web look benefit, for sorting, for information mining, for machine learning, what's more, numerous different frameworks. Third, we have built up an execution of MapReduce that scales to vast groups of machines involving many machines. The execution makes proficient utilization of these machine assets also, subsequently is reasonable for use on a significant number of the huge computational issues experienced at Google.

References

[1] MapReduce: Simplified Data Processing on Large Clusters by Jeffrey Dean and Sanjay Ghemawat.

[2] MapReduce: A major step backwards by David De Witt

[3] Databases are hammers; MapReduce is a screwdriver by Mark C. Chu-Carroll

[4] The Google File System by Sanjay Ghemawat, Howard Gobioff and Shun-Tak Leung