Big Data Analytics: Map Reduce Function

S.Swarnalatha[#1], K.Vidya[*2]
[#]*Assistant Professor & Department of CSE & JNTUH*
Hyderabad, India

Abstract *- Big data often refers simply to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. Big data analytics is the process of examining large and varied data sets i.e., big data -- to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful information that can help organizations make more-informed business decisions. The utilization of Big Data Analytics after integrating it with digital capabilities to secure business growth and its visualization to make it comprehensible to the technically apprenticed business analyzers. Analyzing big data is a very challenging problem today, for such applications; the Map Reduce framework has recently attracted a lot of attention. Google's Map Reduce or its open-source equivalent Hadoop is a powerful tool for building such applications. In this paper, we explained Map Reduce function with sample data.*

Key words: Map *Reduce, Big Data, Data Set*

1. INTRODUCTION

Now a day's web users are creating some quintillion bytes of data. This data comes from social media sites, digital pictures, videos, purchase records and few more. Such huge amount of data is being produced continuously is can be called as Big Data. Nevertheless, as the amounts of data increases exponential, the current techniques are becoming superseded. Dealing with Big Data requires comprehensive coding skills, domain knowledge and statistics.

Big Data applications are almost omnipresent-from marketing to scientific research to customer interests and so on. We can witness Big Data in action almost everywhere today. From facebook which handles over 50 billion photos from its user base to CERN's Large Hydron Collider (LHC) which generates 15PB a year to Walmart which handles more than 1 billion customer transactions in an hour. Over a year ago, the World Bank organized the first WBG Big Data Innovation Challenge which brought forward several unique ideas applying Big Data such as big data to predict poverty and for climate smart agriculture and fore user-focused Identification of Road Infrastructure Condition and safety and so on. The ten V's sum it up pretty well – Viscosity,Virality, Vision, Volume, Velocity, Variety, Variability, Veracity , Visualization, and Value.

Viscosity – Viscosity measures the resistance to flow in the volume of data. This resistance can come from different data sources, friction from integration flow rates, and processing required turning the data into insight. Technologies to deal with viscosity include improved streaming, agile integration bus', and complex event processing.

Virality – Virality describes how quickly information gets dispersed across people to people (P2P) networks. Virality measures how quickly data is spread and shared to each unique node. Time is a determinant factor along with rate of spread.

Vision – every company, that starts with Big Data should have a vision, what to do with them. Downloading Hadoop, installing it and feeding with some data will not help. The company needs to be ready for digital transformation. Sometimes we hear, that technology is ahead of business by 5 years, but that also may be the trap. If management will not understand, what can Big Data offer, there will be no success at all. Vision should be also followed by internal programs a and change of current/old processes.

Volume- Volume is how much data we have – what used to be measured in Gigabytes is now measured in Zettabytes (ZB) or even Yottabytes (YB). The IoT (Internet of Things) is creating exponential growth in data. This infographic from CSC does a great job showing how much the volume of data is projected to change in the coming years.

Velocity - Velocity is the speed in which data is accessible.

Variety - Variety describes one of the biggest challenges of big data. It can be unstructured and it can include so many different types of data from XML to video to SMS. Organizing the data in a meaningful way is no simple task, especially when the data itself changes rapidly.

Variability - Variability is different from variety. A coffee shop may offer 6 different blends of coffee, but if you get the same blend every day and it tastes different every day, that is variability. The same is true of data, if the meaning is constantly changing it can have a huge impact on your data homogenization.

Veracity - Veracity is all about making sure the data is accurate, which requires processes to keep the bad

data from accumulating in your systems. The simplest example is contacts that enter your marketing automation system with false names and inaccurate contact information. How many times have you seen Mickey Mouse in your database? It's the classic "garbage in, garbage out" challenge.

Visualization - Visualization is critical in today's world. Using charts and graphs to visualize large amounts of complex data is much more effective in conveying meaning than spreadsheets and reports chock-full of numbers and formulas.

Value - Value is the end game. After addressing volume, velocity, variety, variability, veracity, and visualization – which takes a lot of time, effort and resources .

Fig 1:Big Data with its characteristics

2. FIVE CASES WHERE BIG DATA WAS A BIG FLOP

Big data may be the technology that everyone's talking about, but that doesn't mean it is flawless. Big data has created havoc in some cases—and the reasons can be anything, such as detection of false positives, lack of tools, technical glitches, low quality data, wrong data, or unnecessary data, Not starting with clear business objectives, Not making a good business case, Management Failure, Poor communication, Not having the right skills for the job. With such errors, it may be possible that the results may be completely different from what you expected. Moreover, the results are sometimes not analyzed, which can lead to unpleasant results.

Another BIG BIG big data failure is not doing data analytic, No matter data size one/company needs to do data analysis but how far and how much is the question to be asked and answered intelligently. A bank doing sentiment analysis on twitter and not analyzing the customer complaints and feedbacks is doing silly. A bigger question is what organization is currently doing with data available at fingertips before investing huge in big data; though investing huge in Big data is another big data failure. Another Big data failure is training organizations making big with big data and data science training's and individuals dreaming to make big money with short course on big data. In the End it is Exciting and it has lot more success case studies than failures.

There are various approaches for preprocessing Big Data, one of that is Map Reducing Function used in Hadoop

3. MAP REDUCING FUNCTION

It is Distributed Data Processing Algorithm, is mainly useful to process huge amount of data in parallel, reliable and efficient way in cluster environments. It uses Divide and Conquer technique to process large amount of data. It divides input task into smaller and manageable sub-tasks to execute them in-parallel.

Map Reduce Algorithm uses the following two main steps:

- Map Function
- Reduce Function

3.1 Map Function

Map Function is the first step in Map Reduce Algorithm. It takes input tasks and divides them into smaller sub-tasks. Then perform required computation on each sub-task in parallel. This step performs the following two sub-steps:

- Splitting- Splitting step takes input DataSet from Source and divide into smaller Sub-DataSets
- Mapping - Mapping step takes those smaller Sub-DataSets and perform required action or computation on each Sub

DataSet The output of this Map Function is a set of key and value pairs as <Key, Value> as shown in the below diagram

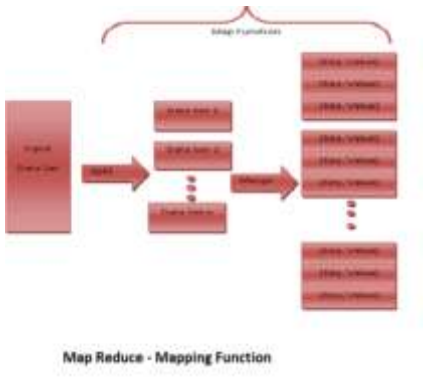

Fig 2: Map Reduce- Mapping Function

3.1.1 Map Reduce First Step Output:

Shuffle Function Output= List of <Key, List<Value>> Pairs

3.2 Reduce Function

It is the second step in Map Reduce Algorithm. It performs only one step: It takes list of <Key, List<Value>> sorted pairs from Shuffle Function and perform reduce operation as shown below.

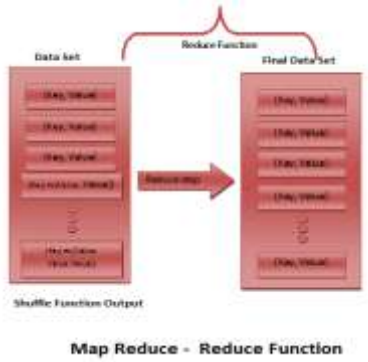

Fig 3: Reduce function

3.2.1 Map Reduce Final Step Output:

Reduce Function Output = List of <key, Value> Pairs

Final step output looks like first step output. However final step <Key, Value> pairs are different than first step <Key Value> pairs. Final step <Key, Value> pairs are computed and sorted pairs. We can observe the difference between first step output and final step output with some simple example. We will discuss same steps with one simple example in next section.

4. MAPREDUCE ALGORITHM WORDCOUNT EXAMPLE

In this section, we are going to discuss about "How Map Reduce Algorithm solves Word Count Problem" theoretically.

4.1 Problem Statement:

Count the number of occurrences of each word available in a DataSet.

4.2 Input DataSet

Please find our example Input DataSet file in below diagram. Just for simplicity, we are going to use simple small DataSet. However, Real-time applications use very huge amount of Data.

1	Strawberry Blueberry Strawberry Blueberry Greenapple Strawberry Blueberry Greenapple
2	Whitegrapes Blackgrapes
3	Strawberry Whitegrapes Blackgrapes
4	Orange Greenapple
5	Strawberry Blueberry Strawberry
6	Blueberry Greenapple Strawberry Blueberry
7	Greenapple Whitegrapes Blackgrapes

Fig 4: sample data

4.1 Map Reduce – Map Function (Split Step)

Fig 5: Map Reduce- split step

4.2 Map Reduce – Shuffle Function (Merge Step)

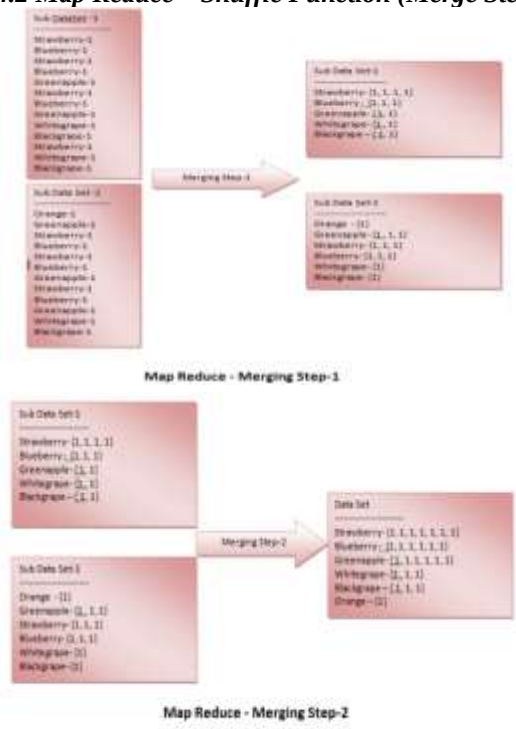

Fig 6: Map Reduce- merge step

Map Reduce – Shuffle Function (Sorting Step)

Fig 7: Map Reduce- sorting step

MapReduce – Reduce Function (Reduce Step)

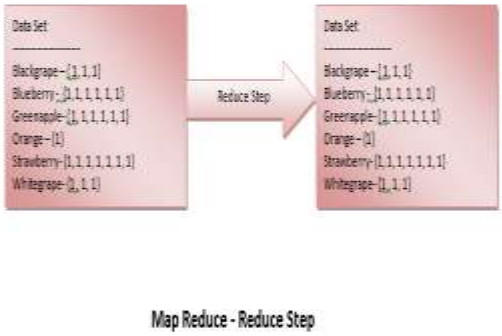

Fig 8: Map Reduce- reduce step

5. LIMITATIONS

. Here are some use cases where Map Reduce does not work very well.

1. Cascading tasks one after the other - using Hive, Pig might help, but lot of overhead rereading and parsing data.
2. Since Map Reduce is suitable only for batch processing jobs, implementing interactive jobs and models becomes impossible.
3. Applications that involve precomputation on the dataset brings down the advantages of Map Reduce.
4. Implementing iterative map reduce jobs is expensive due to the huge space consumption by each job.
5. A problem that cannot be trivially partitionable or recombinable becomes a candid limitation of Map Reduce problem solving. For instance, Travelling Salesman problem.
6. Due to the fixed cost incurred by each Map Reduce job submitted, application that requires low latency time or random access to a large set of data is infeasible.
7. Also, tasks that has a dependency on each other cannot be parallelized, which is not possible through Map Reduce.
8. Cascading tasks one after the other - using Hive, Pig might help, but lot of overhead rereading and parsing data.
9. Since Map Reduce is suitable only for batch processing jobs, implementing interactive jobs and models becomes impossible.
10. Applications that involve precomputation on the dataset brings down the advantages of MapReduce.
11. Implementing iterative map reduce jobs is expensive due to the huge space consumption by each job.
12. Problems that cannot be trivially partitionable or recombinable becomes a candid limitation of MapReduce problem solving. For instance, Travelling Salesman problem.
13. Due to the fixed cost incurred by each MapReduce job submitted, application that requires low latency time or random access to a large set of data is infeasible.
14. Also, tasks that has a dependency on each other cannot be parallelized, which is not possible through MapReduce.

CONCLUSION

As big data processes large volumes of data, with better analysis tools like Map Reduce over Hadoop, that guarantees faster advance in many scientific disciplines and improving profitability and success in many enterprises.

This paper exploits Map Reduce function for efficient analysis of big data for solving challenging data processing problem in large scale applications. It smoothly scale from a single machine to thousands, providing Fault tolerant & high performance.

REFERENCES

[1] R. Taylor. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics BMC bioinformatics,11(Suppl 12):S1, 2010.

[2] A. Pavlo et al . A comparison of approaches to large-scale data analysis. In Proceedings of the ACM SIGMOD, pages 165178, 2009.

[3] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems 25 (2009) 599616.

[4] Hadoop Distributed File Systemhttp://hadoop.apache.org/hdfs[3] Borthakur, D. (2007) The Hadoop Distributed File System: Architecture and Design.http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.p df

[5] W. Jiang et al . A Map-Reduce System with an Alternate API for Multi-core Environments. In Proceedings of the 10th IEEE/ACM CCGrid, pages 8493, 2010.

[6] Map-Reduce: Simplied Data Processing on LargeClusters, by Jerey Dean and SanjayGhemawat; fromGoogle Research.

[7] Arash Baratloo, Mehmet Karaul, Zvi Kedem, and Peter Wyckoff. Charlotte: Metacomputing on the web. In Proceedings of the 9th International Conference on Parallel and Distributed Computing Systems, 1996.

[8] Luiz A. Barroso, Jeffrey Dean, and Urs Holzle. ¨ Web search for a planet: The Google cluster architecture. IEEE Micro, 23(2):22–28, April 2003.