# Analysing Distributed Big Data through Hadoop Map Reduce

Arpit Gupta
Scholar
Amity University
Uttar Pradesh,
Lucknow-226028

Rajiv Pandey,PhD
Member IEEE
Amity University
Uttar Pradesh,
Lucknow-226028

Komal Verma
Scholar
Amity University
Uttar Pradesh,
Lucknow-226028

## ABSTRACT
This term paper focuses on how the big data is analysed in a distributed environment through Hadoop Map Reduce. Big Data is same as "small data" but bigger in size. Thus, it is approached in different ways. Storage of Big Data requires analysing the characteristics of data. It can be processed by the employment of Hadoop Map Reduce. Map Reduce is a programming model working parallel for large clusters. There are some principles that are followed by Hadoop Map Reduce. It also solves the challenges of cluster computing as it hides complexity and minimizes the movement of data.

## Keywords
Map Reduce, Hadoop

## 1. INTRODUCTION
Big Data is same as "small data" but much bigger in size. There are different approaches for it like techniques, tools, architecture. It generates very large quantities value that can't be analysed through normal computing techniques. For Example: From the users, Facebook handles approx. 40 billion photos each day. There are V3s that describe the basic characteristics of big data:

1. Volume: Data Quantity
2. Variety: Data types
3. Velocity: Data Speed

By analysing the data characteristics, the problem of storage of big data can easily be rectified.

Processing of big data cannot be done with normal computing techniques. Hadoop Map Reduce is employed when the extracted data from storage is transformed and sub-divided. Map Reduce is a programming model in which user implements Map() and Reduce(). It is very useful model for large data. Map() extracts the valuable information from each data. Then the extracted information is shuffled and sorted. Reduce() aggregates or filters the information. The analysed data is the output. Hadoop Map Reduce is an effective method for analysing massive datasets. There are some principles of Map Reduce that hides the complexity of processing data sets.

## 2. BIG DATA
Big data, as the name suggests, is a large and complicated data that can't be processed in the way relational data is processed. Big data can be in a structured or unstructured form and size of data may vast. Internet plays a major role in generating this type of data. ***"Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization."*****[1]** Devices from small PDAs, cyber physical systems to big super computers all contribute for the size. Volumes of the data are generated through various social networks like Facebook, WhatsApp, and Twitter. There are many challenges like searching, sharing, storing, transferring, securing, analysing etc. due to which this data can't undergo normal data processing. Advance Methods are used to gather valuable information from the data which reduces the cost and risk, increases the operational efficiency by assured decision making.

Big data comprises of the data with large size which is impossible to be managed, captured, and processed by normal means or common softwares within small amount of time. It's (Big Data) size is constantly increasing, like it moved up from range of terabytes to petabytes around the year 2012. In other words, "Big data can be considered as techniques and technology set which can gather and uncover logical information from vast and complex data which will require new forms of integration.**"**

Big data can be understood more clearly through the under mentioned features:

- **Volume** – It emphasises on the quantity of data *i.e.* volume of data being generated. It is the size that helps determine the value of data and whether or not it is to be considered as Big Data. Hence, the name which also contains a word "Big".
- **Variety** – In order to process and analyse this type of data there is a need to know the category it belongs to. With the information of the category of the data, people analysing or associated with it can use it effectively for their advantage and confirm or support the importance of Big Data.
- **Variability–**It refers that the data is variable or inconsistent, which can cause hindrance in the process of handling data and managing it effectively. It can sometimes create problems for the people who analyse the data.
- **Veracity** – This quality of big data can vary. Veracity refers to the accuracy of the analysis performed on the source data.
- **Velocity** – As the name suggests, velocity refers to the speed at which the data is generating or being processed so as the demands are met and so are the challenges for the development and growth.
- **Complexity** – Management of this type of data is very complicated. As already mentioned above,

when we talk about large amount of data it can't be managed in normal ways. Valuable information is extracted from this complex data which was to be conveyed. Thus, complexity of the data is a characteristic.

## 3. BIG DATA STORAGE

Since the data is growing rapidly in current world, it has requirements for its management and storage. ***"Big data storage refers to the storage and management of large-scale datasets while achieving reliability and availability of data accessing."*****[2].**
There should be reliable storage service that provides information storage (provided by storage infrastructure); also ***"it must provide a powerful access interface for query and analysis of a large amount of data."*****[3]** Traditionally, devices that stores data are used to store, manage, update, alter and analyze the data as with typical and structured Relational Database Management Systems. Devices storing data are becoming more important as with the growing amount of data**.** ***"Many Internet companies pursue big capacity of storage to be competitive. Therefore, there is a compelling need for research on data storage."*****[4]**

There are many different storage systems that have been developed to serve the demands of the huge datasets. The current technologies for storing huge datasets are broadly classified into two storage systems:

1. Direct Attached Storage (DAS)
2. Network Storage
    i. Network Attached Storage (NAS)
    ii. Storage Area Network (SAN)

By analyzing the data characteristics big data stores can be identified:

1. Selecting sources of data for analysis
2. Eliminating the redundant data

Big Data stores include Data Models, Hadoop Distributed File System (HDFS), Hive and HBase.

## 4. BRIEF ABOUT PROCESSING

Processing of Big data includes two main steps:
**Integrating different data stores:**

1. Mapping of data to programmable framework.
2. Extracting the data from stores by establishing a connection.
3. Transformation of data for processing.
4. Subdividing the data for Hadoop map reduce.

**Employing Hadoop Map Reduce:**

1. Preparing Map Reduce jobs.
2. Distributing data for processing across servers.
3. Executing the jobs.
4. Monitoring the progress

## 5. MAP REDUCE

***"MapReduce is a programming model and an associated implementation for processing and generating large data sets."*****[5]** Map Reduce the distributed data processing feature by Apache has been used for mining really massive datasets. Map Reduce as a model of programming can be understood as a method which is implemented to process big data, by the use of a distributed and parallel running algorithm in a group or groups.
We can consider MapReduce as the heart of the Hadoop. It is the programming model that allows thousands of servers in cluster of Hadoop capable to cope and perform under an increased or massive workload. MapReduce is easily understandable to those who have knowledge of cluster computing and data processing in cluster.

We know that big data can't be processed in a normal fashion.

In a map reduce program, filtering and sorting are performed through a procedure called Map(). A Map Reduce program is contains a Map() procedure which performs two things, sorting and filtering, and a procedure called Reduce() which performs summary operation.

Basically, MapReduce refers two tasks that are distinct and separate that is performed by the Hadoop programs. Firstly, there is a task called map which takes and converts a set of data into another set of data, where the elements are individually broken into key/value pairs or tuples. Secondly, there is a reduce job that takes the input from the output of the map task and form combination of the data key/value pairs into a set of smaller key/value pairs.
As the name 'MapReduce' depicts, the map task is performed first and then the reduce task takes place (after the map task).

A MapReduce splits the data-set (which are input) into many independent units which are then processed in complete parallel manner by the map tasks. The output of the maps are then sorted by the framework, which are then again fed into reduce tasks as input (see Figure 1). Scheduling, monitoring the tasks and re-executing (tasks fails) the tasks are handled by the framework. File-system stores both the inputs as well as output of the job.

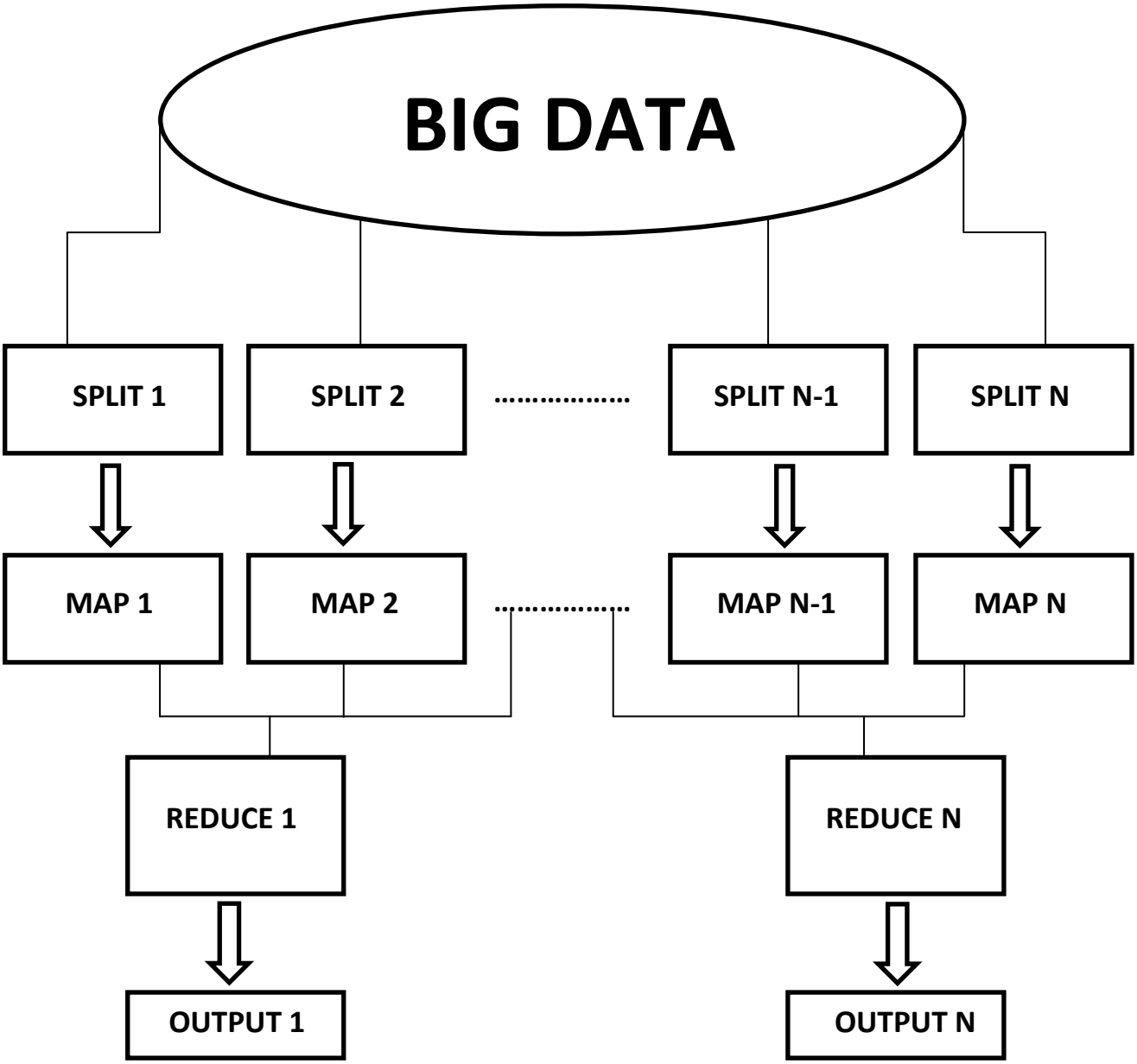


**Figure 1: Basic map reduce flow diagram for big data**

## 6. EXAMPLE

Let's take an example to understand the concept behind the MapReduce (ignoring the algorithms and complex computations).

Assume that anyone has five files, and there are two columns in each file, one of a key and other of a value. The two columns (see Table 1 & Table 2) represent the city and its temperature as recorded for different weeks. Similarly, there are 3 more tables (not shown) that represent city and temperature for different weeks. Consider that the city is the key and temperature is the value for week 1 and week 2.

**Table 1: Week 1**

| **City name (Key)** | **Temperature (Value)** |
|---|---|
| Darjeeling | 20 |
| Shillong | 25 |
| Srinagar | 22 |
| New Delhi | 32 |
| Darjeeling | 4 |
| New Delhi | 33 |
| Srinagar | 18 |

**Table 2: Week 2**

| **City name (Key)** | **Temperature (Value)** |
|---|---|
| Darjeeling | 18 |
| Shillong | 27 |

| | |
|---|---|
| Srinagar | 22 |
| New Delhi | 37 |
| Darjeeling | 12 |
| New Delhi | 33 |
| Srinagar | 32 |

The data, users are concerned to find is the maximum temperature for each of the city through all the data files (each file can have similar city multiple times). In total there are five files, by the use of MapReduce framework, five map tasks can be used to work on one of the five files. The map task analyses the data and then gives maximum temperature for the given city as an output. For example, suppose one map task produces the following result for the given data:

WEEK 1-(Darjeeling, 20) (Shillong, 25) (Srinagar, 22) (New Delhi, 33)

Suppose the other four map tasks produce the result as shown:

WEEK 2- (Darjeeling, 18) (Shillong, 27) (Srinagar, 32) (New Delhi, 37)

WEEK 3- (Darjeeling, 32) (Shillong, 20) (Srinagar, 33) (New Delhi, 38)

WEEK 4- (Darjeeling, 22) (Shillong, 19) (Srinagar, 20) (New Delhi, 31)

WEEK 5- (Darjeeling, 31) (Shillong, 22) (Srinagar, 19) (New Delhi, 30)

Now the reduce tasks would take the output of all the five map tasks as input, would combine the results and will give a single value for each of the given cities as output as shown below:

(Darjeeling, 32) (Shillong, 27) (Srinagar, 33) (New Delhi, 38)

Thus, the output is the maximum temperatures of the given cities.
Although the real time application is not very simple as this was but the principle that was used to analyze the amount of data was same as that for big data. In real time application, there are millions of rows. Now, one can consider how big the data would be.

## 7. MAP REDUCE PRINCIPLES

- **Simple in development**

As we know that MapReduce is a simple model, development as done by the developers is not quite complex. In fact, they do not have to a ton of hard work in their jobs. Simply, there are just programs that process the input files and returns the output. Developers are saved from the complex web of the parallel programming.

- **Scaling**

The map tasks and the reduce tasks doesn't share anything. All the map tasks are not concerned with the other map tasks. They are independent of the other map tasks and their work. Similarly, all the reduce tasks are independent of other reduce tasks. So the map tasks and reduce tasks can work parallel in a massive scale. For example: If a person has 10 nodes and each of them can run 10 map tasks, then a job can easily be split into 100 tasks. Now, if person has block size of 128MB, then a job that takes input file as large as 128MBx100 = 12.8GB can run. If the processing speed is 100MB/sec, then MapReduce would take 2 seconds to process 12.8GB file.

- **Distribution of work automatically**

In the MapReduce model, the map task usually processes one record at one time. Thus, the file can be divided into *n* number of pieces so that it can be executed in parallel. The work *i.e.*, the files is automatically divided into many records by the frameworks that are to be processed by the map tasks.

- **Fault tolerance**

***"Since the MapReduce library is designed to help process very large amounts of data using hundreds or thousands of machines, the library must tolerate machine failures gracefully."*****[6]** Map Reduce model is built immune to the failure. Now, it uses commodity Linux nodes which indicate that failures can occur anytime. But the robustness of the system automatically handles if there is any failure and the users need not to do anything. For example, if a node currently running map tasks fails, the map task will be automatically scheduled on another node.

- **Commodity hardware**

Not only processing is that matters. A person should also be concerned about the economy which is an important aspect of MapReduce. High class machines are not to be expected but the machines with performance that is achieved by data locality and parallel processing.

- **Locality of Data**

This refers to bringing of processing to data rather than bringing data to processing. Anyone can get data from disk to memory in a much fast manner than the data on network. MapReduce framework reduces the data to be transferred over network just by running tasks on the nodes where the data exists.

Overall processing time is significantly reduced by the tasks of scheduling which handles a data block on node.

## 8. WHY MAP REDUCE?

We all are familiar of normal processing of data (see Figure 2)

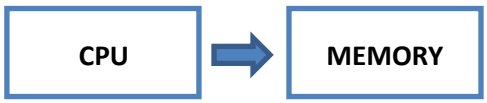


**Figure 2: Normal Processing**

CPU runs algorithm which access data from the memory and processes it. Before the processing, memory reads the data from disk. Once the data is fully in the memory such that there is no need to access the disk again, CPU runs an algorithm to access the data from memory and process it accordingly.

What if the data is so big that it can't even fit in the memory at once? This is when Data Mining comes into play. In classical data mining, there are algorithms that look into CPU and memory as well as the disk. Small portion of data is brought into memory from disk and then the processing can be done with the use of batch processing. Sometimes this type

of data processing model is also not sufficient. It can considered using an example:

Let's suppose that Google has a total of 10 billion webpages and the average size of each webpage is 20 KB.

Total size of webpages= 10 billion*20KB=200TB

Now, if anyone uses classical data mining model to read the data from the webpages, assuming that bandwidth to read data from disk is 50MB/sec.

Total time to read data = 4 million seconds= 46+days. We should consider that this is only the time to read the data. If we have to do something else with the data, something useful, then we need more time. Obviously this would not be accepted by anyone.

Now, there is a much more logical solution for this problem. We can split the data into multiple disks and then can read as well as process the data in multiple CPUs that will surely shorten the time.

Consider that anyone has 1000 CPUs and disks which are working parallel, thus

Total time to read data = 4 million seconds/1000 = 4000 seconds = approx. 1 hour

This is lot less time require to read the data than the previous model and is quite acceptable. Basically, this is the idea of cluster computing. There is basic architecture of cluster computing. There are racks that contain 16-64 commodity Linux nodes (because they are cheap and 1000s of them can be bought easily). These nodes are connected by a switch which is a Gigabit switch so there is 1Gbps bandwidth between the nodes in a rack. There are multiple racks as 64 nodes are not sufficient, and the racks are connected by a switch called backbone switch (which are of higher bandwidth approx. 2-10 Gbps).

This is the standard architecture of cluster which is used for storing as well as mining datasets that are large (see Figure 3)

Now this cluster computing too doesn't solve the problem completely as it has its own disadvantages.

1. Node failure: It means that the nodes on which the data is stored fails.

   *(i)* Then, How to keep the data available (can read it again) if the node fails?

   *(ii)* What if some of nodes that had data necessary for computation, fail while running a long computation?

2. Network Bandwidth: There are several complex computations which require moving a lot of data which can slow the computation down.

3. Distributed Programming: Distributed programming is very hard and complex. There is a need of simple model that can hide the complexity.

"Map Reduce" solves the challenges of the cluster computing.

1. It stores the data on multiple nodes redundantly so that the data is available and guaranteed that it can be read again.
2. It moves the computation close to the data which minimizes the movement of data.
3. There is a simple programming model which hides the complexity of the processing.

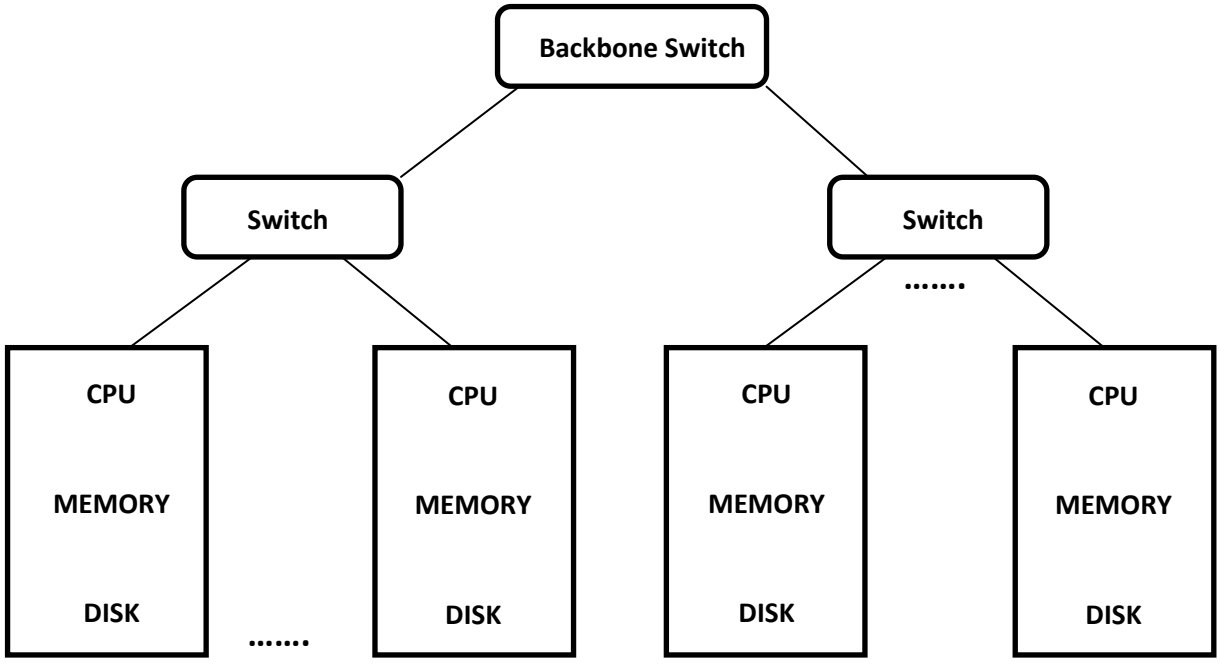


**Figure 3: Standard architecture of cluster computing**

## 9. DISCUSSION AND CONCLUSION

Specialization in processing large data, Hadoop is an open source technology on commodity hardware. Large volumes of data like analytics and commodities is processed on different computers by splitting into small size into size and then re-assembled.

As the size of data is increasing over time, use of Hadoop technologies including Map Reduce will increase. Now the data being generated is in TBs and PBs, although it started with GBs.

***"According to the 'Hadoop-MapReduce Market Forecast 2013-2018,' Hadoop MapReduce is expected to grow at a compound annual growth rate (CAGR) of 58 percent by 2018, which accounts for approximately $2.2 billion."*** **[7]**

***"It seems clear that Hadoop is well-positioned to become the industry standard technology for managing big data and business intelligence solutions, and it presents opportunities for business environments that rely heavily on big data."*** **[8]** There is rapidly growing need for accommodating increasing amounts of data that have to be processed, stored and analyzed. IT vendors have established more cost prohibitive pricing models.

## 10. REFERENCES

[1] Laney, Douglas. "The Importance of 'Big Data': A Definition". Gartner. Retrieved 21 June 2012.http://www.gartner.com/resId=2057415

[2] M.H. Padgavankar, Dr. S.R. Gupta. "Big Data Storage and Challenges" M.H. Padgavankar et al, / (IJCSIT) International Journal of Computer Science and InformationTechnologies,Vol.5(2)2014http://www.ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502284.pdf

[3] M.H. Padgavankar, Dr. S.R. Gupta. "Big Data Storage and Challenges" M.H. Padgavankar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol.5(2)2014http://www.ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502284.pdf

[4] M.H. Padgavankar, Dr. S.R. Gupta. "Big Data Storage and Challenges" M.H. Padgavankar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol.5(2)2014http://www.ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502284.pdf

[5] Jeffrey Dean and Sanjay Ghemawat. "MapReduce: Simplified Data Processing on Large" Google, Inc.http://static.googleusercontent.com/media/research.google.com/es/us/archive/mapreduce-osdi04.pdf

[6] Jeffrey Dean and Sanjay Ghemawat. "MapReduce: Simplified Data Processing on Large" Google, Inc.http://static.googleusercontent.com/media/research.google.com/es/us/archive/mapreduce-osdi04.pdf

**[7]** http://www.websitemagazine.com/content/blogs/posts/archive/2012/08/04/thefuturelooksbrightforhadoopmapreduce.aspx

[8] http://www.websitemagazine.com/content/blogs/posts/archive/2012/08/04/thefuturelooksbrightforhadoopmapreduce.aspx