International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

K-Means Clustering Algorithm on Different Platforms for Big Data Analytics

Snehal V. Pansare[1], Suvarna A. Borhade[2], Sudhakar S. Jadhav[3]

[1, 2]Master of Computer Applications (M.C.A), Lokmanya Tilak College of Enginnering Sector 4, Koparkhairane, Navi Mumbai, India (MH) -400709.

[3]Professor, Master of Computer Applications (M.C.A), Lokmanya Tilak College of Enginnering Sector 4, Koparkhairane, Navi Mumbai, India (MH) -400709.

Abstract: ***Big Data is a term which is used when traditional database and data handling techniques were not able to handle the unstructured and very large data, for analyzing these types of data big data analytics strategy is used. Big data analytics allows data scientists and various users to assist in evaluating large volumes of data that traditional systems would not be able to handle properly. This paper describes implementing K-means clustering algorithm on different platforms available for performing big data analytics.***

Keywords: Big data, Big data analytics, MapReduce, Graphic Processing Unit (GPU), Message Passing Interface (MPI)

1. Introduction

Nowadays, we tend to produce 2.5 quintillion (the number that is represented as a one followed by 18 zeros) bytes of data, which is so large that 90% of the data about more in the world today has been produced in the last two years alone [3].

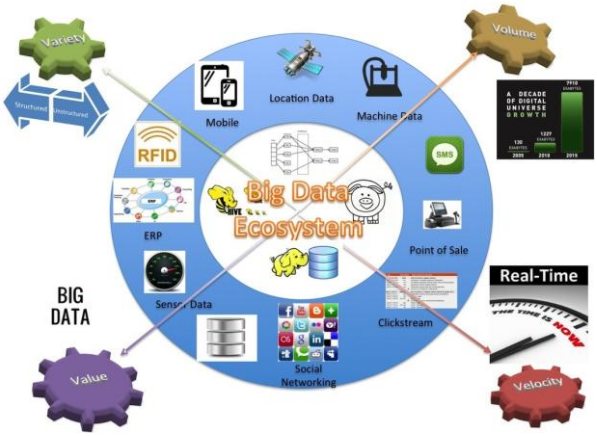

Figure 1: Sources of big data

This data comes from many different sources: The smart phones, the data they generate and consume; sensors embedded into everyday objects, which resulted in billions of new and constantly updating data feed containing location, climate and other information; posts to social media sites, digital photos and videos and purchase transaction records. This data is called big data. The first organizations to grab it were online and startup firms. Firms such as Facebook, Google and LinkedIn are built around big data from the beginning [9].

"Big Data" refers to data sets too large and complicated containing structured, semi-structured and unstructured data, which is very difficult to handle with traditional software tools. In many organizations, the volume of data is bigger or it moves faster or it exceeds current processing capacity [16]. An example of big data might be Petabytes (1,024 terabytes) or Exabyte's (1,024 petabytes) of data containing billions to trillions of records of millions of various users—all from different sources such as social media, banking, web, mobile, employees and customer's data etc. These types of data are typically loosely structured data that is often incomplete and inaccessible [4].

Table 1: Unit Names and Values

Unit Name	Symbol	Value
Kilobyte	KB	10^3
Megabyte	MB	10^6
Gigabyte	GB	10^9
Terabyte	TB	10^12
Petabyte	PB	10^15
Exabyte	EB	10^18
Zettabyte	ZB	10^21
Yottabyte	YB	10^24

2. Why Big Data?

2.1 Difficult to Process by a Traditional System

As given in below diagram, it is depended on capabilities of the system and at an organization level, it is depended on capabilities within the organization. Here big data is resulting into the massive amount of data and growing files, at high speed, in various formats. So it's become very difficult for traditional database system to handle these types of data, which is increasing at every fraction of seconds.

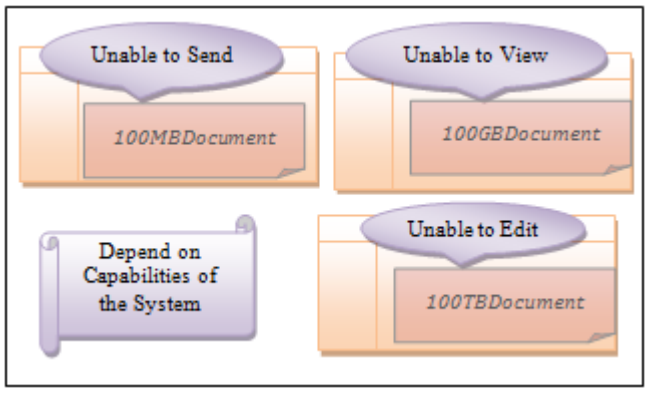

Figure 2: Difficulties in Traditional Data

Big data does not contain only numbers or strings. It contains various data formats like geospatial data, 3D data, digital photos, audio and video, unstructured text, including log files and social media. Traditional database systems were designed to handle an only smaller amount of structured data with very few changes or a predictable, consistent data structure. Traditional database systems are also designed to operate on a single server, creating increased capacity expensive and finite. As today's applications have developed to serve large volumes of data and as application development has had become agile, the use of the traditional relational database has become a drawback for many organizations rather than an empowering factor in their business [15].

3. Challenges of Big Data

As "data" is the key word in big data, one must know the different challenges involved in the data itself in detail, which is given below [6]:

1. Volume (scale of data)
Volume means large amounts of data that is generated every second. The larger the volume of data, it becomes difficult and risky to manage it. The data can be of any size, i.e. it can be of zettabytes or brontobytes. Advanced big data tools use distributed systems so that we can store to analyze and handle data across databases that are cherished around anywhere in the world [5]. A typical PC might have had 10 gigabytes of storage in 2000[9]. Facebook currently holds more than 45 billion photos in its user database [14], a number that is growing daily; a Boeing 737 will generate 240 terabytes of flight data during a single flight across the US. These all shows how volume of data is growing rapidly everyday [9].

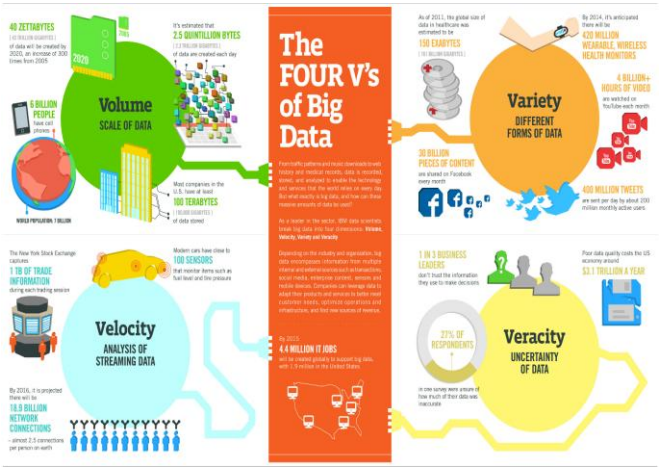

Figure 3: 4V's of Big Data

2. Volume (scale of data)
Volume means large amounts of data that is generated every second. The larger the volume of data, it becomes difficult and risky to manage it. The data can be of any size, i.e. it can be of zettabytes or brontobytes. Advanced big data tools use distributed systems so that we can store to analyze and handle data across databases that are cherished around anywhere in the world [5]. A typical PC might have had 10 gigabytes of storage in 2000[9]. Facebook currently holds more than 45 billion photos in its user database [14], a number that is growing daily; a Boeing 737 will generate 240 terabytes of flight data during a single flight across the US. These all shows how volume of data is growing rapidly everyday [9].

3. Velocity (analysis of streaming data)
Velocity refers to the speed at which new big data is generated and the speed at which data moves around. Just think of the New York Stock Exchange alone generates one terabyte of trade information during each session [5]. Clickstreams and ad impressions capture user behavior at millions of events per second, high-frequency stock trading algorithms affected market changes within a fraction of seconds; data exchange between various devices; infrastructure and sensors that created a large amount of log data in real-time; on-line gaming systems support millions of concurrent users, each producing multiple inputs per second [8].

4. Variety (different forms of data)
Variety means the different formats of data. Big data is data of different forms. In the past, we focused only on structured data, which is recorded correctly into tables or relational databases, such as customer's data. But 80% of world's data that is generated everyday is unstructured (text, pictures, audio, video, etc.). With big data technology, we have to analyze and integrate data from different types such as messages, social media conversations, images, video or audio [5]. Therefore, Handling and managing different types of data, their formats and source is a big challenge.

5. Veracity (uncertainty of data)
Veracity means accuracy or correctness of data. Because of various forms of big data quality and precisions are less controllable. (Just think of twitter posts with hash tags, abbreviations, typos and colloquial speech as well as the reliability and accuracy of content)[5]. But with new technology, one should be able to work with imprecision, uncertainty, missing values, misstatements or untruthful data [6].

4. Big Data Analytics

Big data analytics is the process of examining big data to discover hidden patterns, unknown correlations and other useful information that can be used to make better decisions [7]. To perform any kind of analysis on such large and complicated data, scaling up the hardware platforms become necessary and choosing the right platforms becomes a crucial decision to satisfy the user's requirement in fewer amounts of time. There are various big data platforms available with different characteristics. To choose a right platform for specific application one should have knowledge of the advantages and limitations of all these platforms. The platform you choose must be able to cater to increased data processing demands if it is appropriate to build the analytics based solutions on a particular platform.

4.1 Scaling

Scaling refers to the system's ability to cater to rising demands in terms of data processing. To process big data different platforms used different forms of scaling. Big data

analytic platforms can be classified into the two types of scaling:

1) **Horizontal Scaling:**
Horizontal scaling means distribution of the workload across multiple servers who can be even commodity server machines. It is also called as "scale out" as to improve performance of data processing numerous separate machines are combined together. It usually involves many instances of the operating system that are running on independent machines. Example of the Horizontal scaling platform is Peer to peer network and Apache Hadoop under which Message Passing Interface (MPI) and MapReduce come respectively.

2) **Vertical Scaling**
Vertical Scaling means installation of more processors, more memory and faster hardware within a single server. It is also called as "scale up" and typically it involves a single instance of an operating system. Examples of the vertical scaling platform are Graphics Processing Unit (GPU), High-Performance Computing Clusters (HPC), Multicore processors, etc. [1]. Some of these platforms are discussed below:

a) **MapReduce**:
MapReduce is a programming model introduced by Google and an associated implementation for processing and generating large data sets on clusters of computers. The MapReduce framework has two parts:
1. A function called "Map," that allows different points of the distributed cluster to distribute their work.
2. A function called "Reduce," which is designed to reduce the final form to the cluster's results into single output.

The main advantage within the MapReduce framework is its fault tolerance, wherever periodic reports from every node within the cluster are expected once work is completed. A task is transferred from one node to another. If the master node identifies that a node has been idle for a long interval than expected, the master node assigns the process to the frozen or delayed task [2].

4.2 Graphics processing unit (GPU)

A graphics processor unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display [12]. GPUs have also served as the processing units in gaming cards. GPUs cannot perform all of the functions of a CPU. However, GPUs are hyper-efficient at performing mathematical computations. GPUs can perform hundreds of thousands (and potentially trillions) of mathematical operations in parallel on hundreds (or thousands) of cores with speed that a quad-code or 8-core which CPUs simply can't achieve. Modern GPUs are very efficient at manipulating computer graphics and image processing, and their highly parallel structure makes them more effective than general-purpose CPUs for algorithms where processing of large blocks of data is done in parallel [13].

4.3 Message Passing Interface (MPI):

MPI is a standardized and portable message-passing system which is designed by a group of researchers to function on a wide variety of parallel computers [9]. MPI was designed to handle distributed-memory systems, i.e. clusters. MPI as a distributed-memory system implies that multiple processes are started from the beginning and run, usually on different CPUs for completion. These processes do not have anything in common, and each has its own memory space. The MPI interface is used to provide essential virtual topology, Asynchronization and Communication functionality. One of the main features of MPI includes the state preserving process, i.e. processes can live as long as the system runs and there is no need to read the same data again and again [10].

5. How to choose a platform for big data analytics?

The decision to choose a particular platform for a specific application usually depends on the following important factors: data size, speed or throughput optimization and model development.

1) **Data Size:**
The size of data which need to be processed is one of the major factors for choosing a platform. If the data can fit into the system memory, then there is no need of clusters and the entire data can be processed efficiently on a single machine. If the data does not fit into the system memory, then one has to look for other cluster options.

2) **Speed or throughput optimization:**
Speed is the ability of the particular platform to process data in real-time whereas throughput means the amount of data that system can handle and process simultaneously. The users should be clear about whether the goal is to improve the system for speed or throughput. If one needs to process large amount of data and do not have a time limit, then one can choose the systems which can scale out to process huge amounts of data. Otherwise if one needs to optimize the system for speed rather than data sizes, then they need to consider systems, which are more capable of real-time processing.

3) **Training/Applying a model:**
In big data analytics, training on the model is done offline, which take a lot of time. When a model is applied in an online environment, user wants the results within a short period of time. Therefore, it's become necessary to investigate different platforms for training and applying a model depending upon the end-user application. In training process, the user has to deal with a large amount of training data. As training is done offline the processing time is not a critical, therefore, horizontal scale out platforms suitable for training a model. But when a model is applied, vertical scale up platforms are usually used as results are expected in real-time [1].

6. K-Means Clustering

K-Means Clustering is the process of partitioning a group of data points into a small number of clusters [11]. This is one of the most commonly and effective methods to classify data because of its simplicity and ability to handle large data sets. The K-Means algorithm was chosen because of the following characteristics:

1) Iterative nature of the algorithm wherein the results of current iteration are needed before starting the next iteration.
2) Compute-intensive task of calculating the centroid from a set of data points.
3) Aggregation of the local results to obtain a global solution when the algorithm is parallel.

The main purpose of clustering techniques is to partition a set of entities into different groups, called clusters. These groups may be consistent in terms among similarity of its members. Different steps involved in a basic K-means clustering algorithms are given below:

Input: Data points D, number of clusters k
Step 1: Initialize K centroids randomly
Step 2: Calculate the distance associated with each data point D with the nearest centroid. This will divide the data point into K clusters.
Step 3: Recalculate the position of centroids.
Step 4: Repeat steps 2 and 3 until there are no more changes in the membership of the data points.
Output: Data points with cluster memberships.

The algorithm starts with initializing the cluster centroid. After initialization, distance between each data point and cluster centres is calculated and in the third step, the centroids are recalculated for all the associated data instances for a given cluster. The second and third steps are repeated until the centroids converge or after a predefined number of iterations.

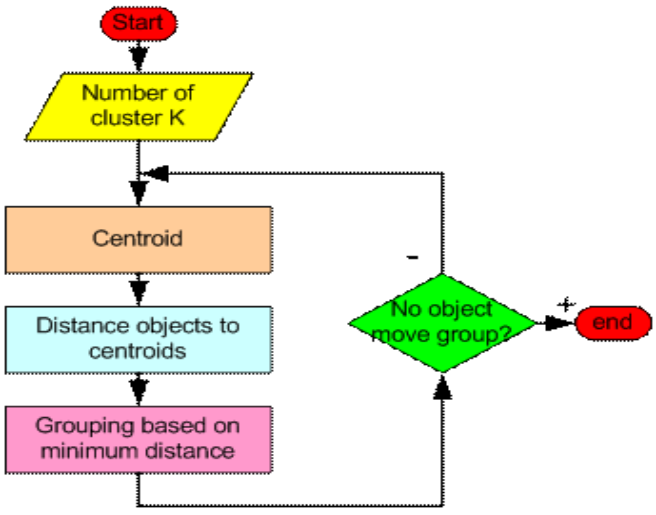

Figure 3: Flowchart of K-means clustering algorithm.

The pseudocode of the K-means clustering algorithm on different Platform:

- Implementation details of an algorithm on different platforms are discussed below to get a deeper understanding of how such iterative algorithms are modified to fit different communication schemes.
- In order to provide more details about an analytics algorithm on different platforms, we will demonstrate the implementation of the K-Means clustering algorithm on different platforms.

6.1 K-means on MapReduce

MapReduce is not an ideal choice for iterative algorithms such as K-Means clustering. The pseudocode for mapper and reducer functions for k-means clustering algorithm is given below:

K-means: Map
Input: Data points D, number of clusters k and centroids
Step 1: For each data point d ∈ D do
Step 2: Assign d to the closest centroid
Output: Centroids with associated data points
K-means: Reduce
Input: Centroids with associated data points
Step 1: Compute the new centroids by calculating the average of data points in cluster.
Step 2: Write the global centroids to the disk
Output: New Centroids

Here, mappers read the data points and the centroids from the disk. These mappers then assign data points to clusters. Once every mapper has completed their operation, reducers compute the new centroids by calculating the average of data points present in each cluster. Now, these new centroids are written to the disk. These centroids are then read by the mappers for the next iteration, and the entire process is repeated until the centroid converges. This shows the disk access bottleneck of MapReduce for iterative tasks as the data has to be written to the disk after every iteration.

6.2 K-means on MPI:

The pseudo code for k-means clustering algorithm on MPI is given below:

Input: Data points D, numbers of clusters k
Step 1: Slaves read their part of data
Step 2: do until global centroids to the slaves
Step 3: Master broadcasts the centroids to the slaves
Step 4: Slaves assign data instances to the closest centroids
Step 5: Slaves compute the new local centroids and local cluster sizes
Step 6: Slaves send local centroids and cluster sizes to the master
Step 7: Master aggregates local centroids weighted by local cluster sizes into global centroids.
Output: Data points with cluster memberships.

MPI typically have a master–slave setting and the data has usually distributed, among the slaves. In the second step, the master broadcasts the centroids to the slaves. Next, the slaves assign data points to the clusters and compute new local centroids which are then sent back to the master. Master will then compute new global centroids by aggregating local

centroids weighted by local cluster sizes. These new global centroids are, then again, broadcasted back to the slaves for the next iteration of K-means. In this manner, the process continues until the centroids converge. In this implementation, the data is not written to the disk but the primary bottleneck lies in the communication when MPI is used with peer-to-peer networks since aggregation is costly and the network performance will be low.

6.3 K-means on GPU

The pseudo code for k-means clustering algorithm on MPI is given below:

K-means: GPU
Input: Data points D, number of clusters k
Step 1: Do until global centroids converge
Step 2: Upload data points to each multiprocessor and centroids to the shared memory
Step 3: Multiprocessor works with one data vector at a time and associate it with the closest centroid.
Step 4: Centroid recalculation is done on CPU.

In case of K-means, each processor is given a small task i.e. assigning a data point to a centroid. Centroid recalculation is done on the CPU as a single core of GPU is not powerful. Therefore, centroids are uploaded to the shared memory of the GPU, and the data points are classified and uploaded into each multiprocessor. These multiprocessors work on one data vector at a time and associate it with the closest centroid. Once all the points are assigned to the centroids, CPU recalculates the centroids and again will upload the new centroids to the multiprocessors. This process is repeated until the centroids converge. Another aspect to consider here is the density of the data. If the data is sparse, many multiprocessors will stop due to scarcity of data vectors to compute, which will eventually degrade the performance. In a nutshell, the performance of GPUs will be the best when the data is relatively denser and when the algorithm is carefully modified to take advantage of processing cores [1].

7. Comparison of Different Platforms

Features	*MapReduce*	*MPI*	*GPU*
Fault Tolerance	Have efficient in-built fault tolerant mechanisms	Don't have any fault tolerance mechanisms	Have fault tolerance mechanisms
Iterative Processing	Not suitable	Suitable	Highly suitable
Scalability	Highly scalable	Highly scalable	Less scalable
Data Size	Suitable for processing large data sets	Suitable for processing large data sets	Not suitable for processing large data sets
Limitation	Disk access is a major limitation which significantly degrades the performance.	Primary limitation lies in the communication when MPI is used with peer-to-peer networks since aggregation is costly.	The primary limitation is the limited memory because when the data size is more than the size of the GPU memory, the performance decreases.

8. Conclusion

This paper describes big data and 4v's (challenges of big data). Nowadays, the volume of information exchange involves a huge amount of data processing. So through this paper we try to focus on implementation of k-means algorithm on different big data analytics platforms for efficient data processing. K-means algorithm was chosen because of its iterative nature. The future of big data analytic involves implementing various algorithms like nearest neighbor, decision tree, page rank, etc. on different platforms. One can decide to choose the particular platform for specific application based upon its features and limitations. Combination of platforms can be used for better performance.

9. Acknowledgement

We would like to thank our guide, Prof. Sudhakar Jadhav for his guidance and support, which has helped us, complete this research paper successfully.

References

[1] Dilpreet Singh and Chandan K Reddy(A survey on platforms for big data analytics)
[2] http://www.techopedia.com/definition/13816/mapreduce
[3] http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html
[4] http://www.webopedia.com/TERM/B/big_data.html
[5] http://www.slideshare.net/nagaritwikindugu/big-data-analyticsintrohadoop-map-reducemahoutkmeans-clusteringhbase
[6] http://www.techopedia.com/2/30575/trends/big-data/big-datas-key-challenges
[7] http://www.sas.com/en_us/insights/analytics/big-data-analytics.html
[8] http://www.slideshare.net/nasrinhussain1/big-data-ppt-31616290
[9] https://en.wikipedia.org/wiki/Message_Passing_Interface
[10] https://www.hpcvl.org/faqs/programming/mpi-message-passing-interface
[11] http://www.onmyphd.com/?p=k-means.clustering
[12] https://en.wikipedia.org/wiki/Graphics_processing_unit
[13] http://www.fuzzyl.com/products/gpu-analytics/
[14] http://www.datacenterknowledge.com/archives/2012/09/12/a-look-into-the-big-data-battleground-analyzing-the-market/
[15] http://bigdata-tuts.blogspot.in/
[16] https://en.wikipedia.org/wiki/Big_data