ISSN 2395-1621

# Analysis of Big Medical Data using the k-mean Clustering on Hadoop Environment

[#1]Prajkta Diwan, [#2]Ajay Fargade, [#3]Sujata Karpe, [#4]Prof. Ganesh Bandhal

[1]diwanpraju18@gmail.com
[2]ajay25061991@gmail.com
[3]sujata.karpe1993@gmail.com
[4]ganeshbandhal@raisoni.net

[#1234]Department of Computer

G.H.R.C.E.M, Wagholi, Pune.

**ABSTRACT**

**There is a drastic growth of data's in the web applications such data's are said be as Big Data. The Hive queries with the integration of Hadoop are used to generate the report analysis for thousands of datasets. It requires huge amount of time consumption to retrieve those datasets. It lacks in performance analysis. To overcome this problem in medical analysis. A very popular Data Mining Algorithm is used in anywhere. The objective is to store the data persistently along with the past history of the data set and performing the report analysis of those data set. The main aim of this system is to improve performance through parallelization of various operations such as loading the data, index building and evaluating the queries. In this system we analysis different medical problems occurred to the user. We use the medical data set for calculate the how may user have problems.**

ARTICLE INFO

**Article History**

Received: 1st June 2017

Received in revised form :

1st June 2017

Accepted: 5st June 2017

**Published online :**

**6th June 2017**

**Keywords: Hadoop, Big data, Medical Data, Calculation, k-mean cluster.**

## I. INTRODUCTION

The proposed system is implemented using Hadoop for real time processing of the Sensex data to generate results to establish the hadoop processing. Data are analyzed with Hadoop using MapReduce programming. In smart city Datasets generated by smart homes, smart parking weathser, pollution, Sensex value and vehicle data sets are used for analysis and evaluation. This type of system with full functionality does not currently exist. Similarly, the results demonstrate that the proposed system is more scalable and efficient than existing systems. Moreover, system efficiency is measured in terms of throughput and processing time. MapReduce is a popular computing framework for large-scale data processing. Practical experience shows that inappropriate configurations can result in poor performance of MapReduce jobs, however, it is challenging to pick out a suitable configuration in a short time. Also, current central resource scheduler may cause low resource utilization, and degrade the performance of the cluster.

Big data is an one of the emerging hot research topic because its mostly used in data center application in human society, such as government, climate, finance, and science. Currently, most research work on big data falls in data mining, machine learning, and data analysis. The name itself contains the meaning of data will be so big‖ in large volume of both structured and unstructured data present. These data centers run hundreds of or thousands of servers, so it consumes megawatts of power with massive carbon footprint, and also incur electricity bills of millions of dollars. Data explosion is one of the rising demand for big data processing in recent years and modern data centers that are usually distributed at different geographic regions. efficient.

Hadoop is java based framework that allows to process large data sets in distributed environment. Hadoop has been used by many large scale companies like Amazon, Facebook, and Yahoo. Hadoop consist of two important concepts:Hadoop Distributed File System (HDFS) and Hadoop Map Reduce. Map Reduce workloads may be very heterogeneous in terms of theirdata size and their re- source requirements , and mixing them within a single instanceof a computing framework may lead to conflicting optimization goals. Therefore,isolating Map Reduce workloads and their data while dynamically balancing theresources across them is very attractive for many organizations .Hadoop is an open

 

source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming mode. It is designedto scale up from single servers to thousands of machines, each offering local computationand storage. Our System relaxes the slot allocation constraint to allow slots to be reallocated to either map or reduce tasks depending on their needs. Second, thespeculative execution can tackle the straggler problem, which has shown to improvethe performance for a single job but at the expense of the cluster efficiency. In viewof this, we propose Speculative Execution Performance Balancing to balance theperformance tradeoff between a single job and a batch of jobs. Third, delay schedulinghas shown to improve the data locality but at the cost of fairness. Alternatively,we propose a technique called Slot PreScheduling that can improve the data localitybut with no impact on fairness. Finally, by combining these techniques together, weform a step-by-step slot allocation system called Dynamic MR that can improve theperformance of Map Reduce workloads substantially.

Goal and Objective:

- Stores large medical database at the same time it can analyze the data mining Algorithm.
- Hadoop processes data fast which is very useful for Real Time System.
- Improves the performance of workloads with maintaining the fairness.

Objectives:

- To minimize the processing
- To reduce the overhead server.

### A. MapReduce Framework:

The mapreduce framework consists of two steps namely Map step and reduce step. Master node takes large problem input and slices it into smaller sub problems and distributes these to worker nodes. Worker node may do this again and leads to a multi-level tree structure.

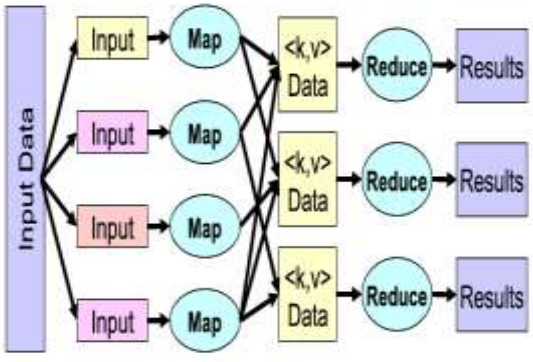


Fig.1 MapReduce Framework

Worker processes smaller problem and hands back to master. In Reduce step Master node takes the answers to the sub problems and combines them in a predefined way to get the output/answer to original problem. The MapReduce framework is fault-tolerant because each node in the cluster is expected to report back periodically with completed work and status updates.

If a node remains silent for longer than the expected interval, a master node makes note and re-assigns the work to other nodes. The detailed MapReduce framework is shown in Fig. 1

### B. Hadoop Distributed File System(HDFS):

It is distributed file system designed to run on commodity hardware. This system provides high- throughput access to application data. HDFS is highly fault- tolerant and is designed to be deployed on low-cost hardware. Application that run on HDFS has large data sets. Typically file in HDFS is gigabytes to terabytes in size. It should support tens of millions of files in a single instance. HDFS is designed more for batch process in grather than interactive use by users. Detection of faults and quick, automatic recovery from them is a core goal of HDFS. HDFS has been designed to e easily poratable from one platform to another. HDFS has a Master-slave architecture. An HDFS cluster consist of a single NameNode, a master serves that manages the file system namespaces and regulates access to files by clients. In addition, there are number of datanodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on.

## II. LITERATURE SURVEY

[1] G. Lee, J. Lin, C. Liu, A. Lorek, and D. Ryaboy, "The Unified Logging Infrastructure for Data Analytics at Twitter," in Proceedings of Very Large Data Base Endow., vol. 5, no. 12, pp. 1771–1780, 2012.
Analysis: Analysis Twitter's production logging infrastructure and its evolution from application-specific logging to a uni-fied "client events" log format, where messages are captured in common, well-formatted, flexible Thrift messages.
Finding: This approach afforded significant flexibility and allowed for very fast application logging development.

[2] G. Mishne, J. Dalton, Z. Li, A. Sharma, and J. Lin, "Fast data in the era of big data: Twitter's real-time related query suggestion architecture," in Proceedings of the 2013 International Conference on Management of Data, ACM, pp. 1147–1158, 2013.
Analysis: He present the architecture behind Twitter's real-time related query suggestion and spelling correction service.
Finding: Build a generic data processing platform capable of handling both "big data" and "fast data.

[3] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley,M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster com-puting," in Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, pp. 2–2, 2012.
Analysis: He present Resilient Distributed Datasets (RDDs), a distributed memory abstraction that lets programmers perform in-memory computations on large clusters in a fault-tolerant manner.

 

Finding: RDD benefits distributed memory abstraction, we compare them against distributed shared memory (DSM)

[4] Z. Zhang, M. Zhang, A. G. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian, "Optimizing Cost and Performance in Online Service Provider Networks," in Proceedings of the USENIX Network System Design and Implementation, USENIX Association, pp. 33–48, 2010.

Analysis: He present a method to jointly optimize the cost and the performance of delivering traffic from an online service provider (OSP) network to its users.

Finding: He find that by OSP can reduce its traffic cost by 40% without any increase in path latency and with acceptably low overheads.

[5] P. Bod´ ık, I. Menache, M. Chowdhury, P. Mani, D. A. Maltz, and I. Stoica, "Surviving Failures in Bandwidth-constrained Datacen-ters," in Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, ACM, pp. 431–442, 2012.

Analysis: He propose and evaluate a novel optimization framework that achieves both high fault tolerance and significantly reduces bandwidth usage in the network core by exploiting the skewness in the observed communication patterns.

Finding: explore the tradeoff between improving fault tolerance and reducing bandwidth usage.

[6] K. yin Chen, Y. Xu, K. Xi, and H. Chao, "Intelligent virtual machine placement for cost efficiency in geo-distributed cloud systems," in Proceedings of International Conference on Communications, IEEE, pp. 3498–3503, 2013.

Analysis: In the CAVP problem formulation, we capture the intrinsic trade-off between electricity cost and WAN communication cost, as well as the electricity price diversity at different geographic locations.

Finding: The results shows that the potential of performance improvement is significant, and partial-optimizing heuristics.

## III. PROPOSED SYSTEM

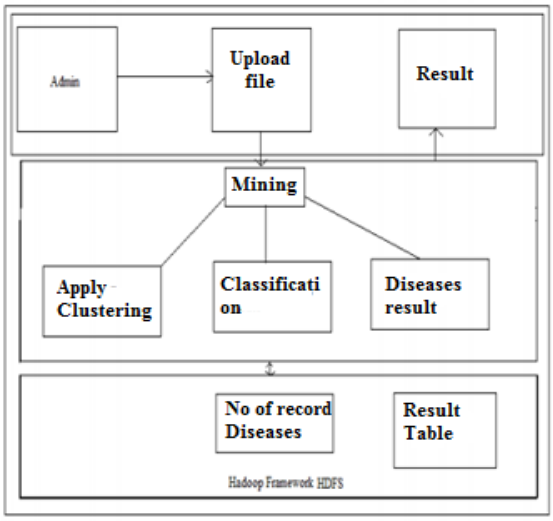


Fig 2. System architecture

## IV. SYSTEM ANALYSIS

After the analysis of the data, Hadoop is installed on a windows Machine and the sample data is inserted into the repository. And Later on, the Map-Reduce code is executed along with an initialization cluster. We can browse all the files in the default URL using the localhost. We need to specify all the paths for running the map-reduce like the input directory, output directory and the cluster initialization directory along with the algorithm we used. Hadoop provides a reliable output in mapping and analyzing of the data. This Data can be employed for further representation.

**Algorithm:**

**Input:** A set J of n MapReduce jobs. Di is the attribute of job Ji as defined above.

**Output:** Schedule σ (order of jobs execution.)

1: Sort the original set J of jobs into the ordered list L using their stage duration attribute $D^1i$
2: head ← 1,tail ← n
3: for each job Ji in L do
4: if $D^2i$ = m then
5: // Put job Ji from the front
6: σ head ← Ji, head ← head + 1
7: else
8: // Put job Ji from the end
9: σ tail ← Ji, tail ← tail - 1
10: end if
11: end for

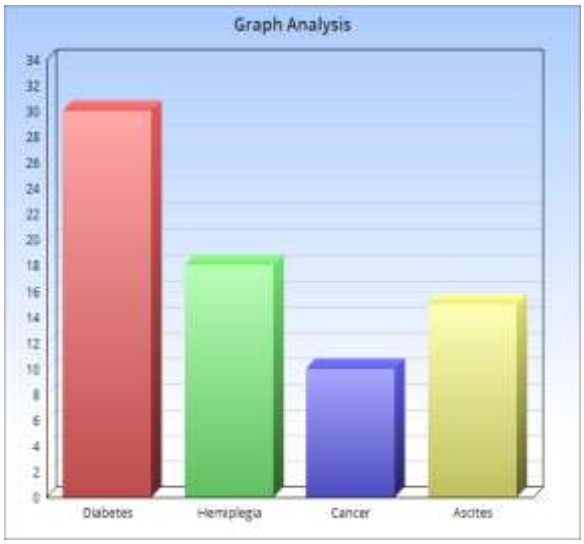


Fig 3. Analysis result

## V. CONCLUSION

The aim of the proposed system is to improve the performance of Map Reduce workloads. It considered three techniques: Hadoop Mining we implement the system for analysis the big data set on medical environment, we calculate the how many people male/female, and area wise have diseases.

 

**REFRENCES**

[1] Lin Gu, Student Member, IEEE, Deze Zeng, Member, IEEE, Song Guo, Senior Member, IEEE, Yong Xiang, Senior Member, IEEE, Jiankun Hu, Member, IEEE A General Communication Cost OptimizationFramework for Big Data Stream Processing in Geo-distributed Data Centers.

[2] G. Lee, J. Lin, C. Liu, A. Lorek, and D. Ryaboy, "The Unified Log-ging Infrastructure for Data Analytics at Twitter," in Proceedings of Very Large Data Base Endow., vol. 5, no. 12, pp. 1771–1780, 2012.

[3] G. Mishne, J. Dalton, Z. Li, A. Sharma, and J. Lin, "Fast data in the era of big data: Twitter's real-time related query suggestion architecture," in Proceedings of the 2013 International Conference on Management of Data, ACM, pp. 1147–1158, 2013.

[4] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley,M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster com-puting," in Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, pp. 2–2, 2012.

[5] Z. Zhang, M. Zhang, A. G. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian, "Optimizing Cost and Performance in Online Service Provider Networks," in Proceedings of the USENIX Network System Design and Implementation, USENIX Association, pp. 33–48, 2010.

[6] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks,"SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 68–73, 2008.

[7] Sarannia, N. Padmapriya, Asst prof, "Survey on Big Data Processing in Geo Distributed Data Centers" Volume 4, Issue 11, November 2014.

[8] Harshawardhan S. Bhosale "A Review Paper on Big Data and Hadoop", International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014 1ISSN 2250-315.

 