Addressing Big Data Problem Using Hadoop and Map Reduce

Aditya B. Patel, Manashvi Birla, Ushma Nair

Abstract-- **The size of the databases used in today's enterprises has been growing at exponential rates day by day. Simultaneously, the need to process and analyze the large volumes of data for business decision making has also increased. In several business and scientific applications, there is a need to process terabytes of data in efficient manner on daily bases. This has contributed to the big data problem faced by the industry due to the inability of conventional database systems and software tools to manage or process the big data sets within tolerable time limits. Processing of data can include various operations depending on usage like culling, tagging, highlighting, indexing, searching, faceting, etc operations. It is not possible for single or few machines to store or process this huge amount of data in a finite time period. This paper reports the experimental work on big data problem and its optimal solution using Hadoop cluster, Hadoop Distributed File System (HDFS) for storage and using parallel processing to process large data sets using Map Reduce programming framework. We have done prototype implementation of Hadoop cluster, HDFS storage and Map Reduce framework for processing large data sets by considering prototype of big data application scenarios. The results obtained from various experiments indicate favorable results of above approach to address big data problem.**

Index Terms-- Big Data Problem, Hadoop cluster, Hadoop Distributed File System, Parallel Processing, Map Reduce

I. INTRODUCTION

In this electronic age, increasing number of organizations are facing the problem of explosion of data and the size of the databases used in today's enterprises has been growing at exponential rates. Data is generated through many sources like business processes, transactions, social networking sites, web servers, etc. and remains in structured as well as unstructured form [1]. Today's business applications are having enterprise features like large scale, data-intensive, web-oriented and accessed from diverse devices including mobile devices. Processing or analyzing the huge amount of data or extracting meaningful information is a challenging task.

II. BIG DATA

The term "Big data" is used for large data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time. Big data sizes are a constantly moving target currently ranging from a few dozen terabytes to many petabytes of data in a single data set [2]. Difficulties include capture, storage, search, sharing, analytics and visualizing. Typical examples of big data found in current scenario includes web logs, RFID generated data, sensor networks, satellite and geo-spatial data, social data from social networks, Internet text and documents, Internet search indexing, call detail records, astronomy, atmospheric science, genomics, biogeochemical, biological, and other complex and/or interdisciplinary scientific research, military surveillance, medical records, photography archives, video archives, and large-scale eCommerce. Big Data impacts include Walmart handles more than 1 million customer transactions every hour, which is imported into databases estimated to contain more than 2.5 petabytes of data - the equivalent of 167 times the information contained in all the books in the US Library of Congress, Facebook handles 40 billion photos from its user base and so on.

A. What is Big Data Problem?

Big Data has emerged because we are living in a society which makes increasing use of data intensive technologies. One current feature of big data is the difficulty working with it using relational databases and desktop statistics/visualization packages, requiring instead "massively parallel software running on tens, hundreds, or even thousands of servers"[3]. The various challenges faced in large data management include – scalability, unstructured data, accessibility, real time analytics, fault tolerance and many more. In addition to variations in the amount of data stored in different sectors, the types of data generated and stored—i.e., whether the data encodes video, images, audio, or text/numeric information—also differ markedly from industry to industry[4].

B. Big data techniques and technologies

Big data requires exceptional technologies to efficiently process large quantities of data within tolerable elapsed times. Technologies being applied to big data include massively parallel processing (MPP) databases, data mining grids, distributed file systems, distributed databases, cloud computing platforms, the Internet, and scalable storage systems. Real or near-real time information delivery is one of the defining characteristics of Big Data Analytics. Latency is

therefore avoided whenever and wherever possible. A wide variety of techniques and technologies has been developed and adapted to aggregate, manipulate, analyze, and visualize big data [2]. These techniques and technologies draw from several fields including statistics, computer science, applied mathematics, and economics. This means that an organization that intends to derive value from big data has to adopt a flexible, multidisciplinary approach.

C. Hadoop [5]

The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using a simple programming model. It enables applications to work with thousands of computational independent computers and petabytes of data. Hadoop was derived from Google's MapReduce and Google File System (GFS).

D. HDFS (Hadoop Distributed File System) [5]

The Hadoop Distributed File System (HDFS) is a distributed file system providing fault tolerance and designed to run on commodity hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. Hadoop provides a distributed filesystem (HDFS) that can store data across thousands of servers, and a means of running work (Map/Reduce jobs) across those machines, running the work near the data. HDFS has master/slave architecture. Large data is automatically split into chunks which are managed by different nodes in the hadoop cluster.

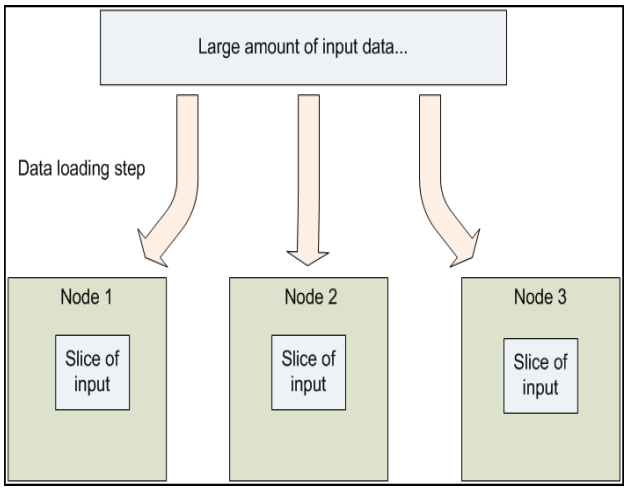

Figure 1: Data is distributed across nodes at load time

E. MapReduce Programming Framework [6]

MapReduce is a software framework introduced by Google in 2004 to support distributed computing on large data sets on clusters of computers . MapReduce is a programming model for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs and a reduce function that merges all intermediate values associated with the same intermediate key [7].

"Map" step: The master node takes the input, partitions it up into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node. Map takes one pair of data with a type in one data domain, and returns a list of pairs in a different domain:

Map (k1, v1) → list (K2, v2)

"Reduce" step: The master node then collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.
The Reduce function is then applied in parallel to each group, which in turn produces a collection of values in the same domain:

Reduce (K2, list (v2)) → list (v3)

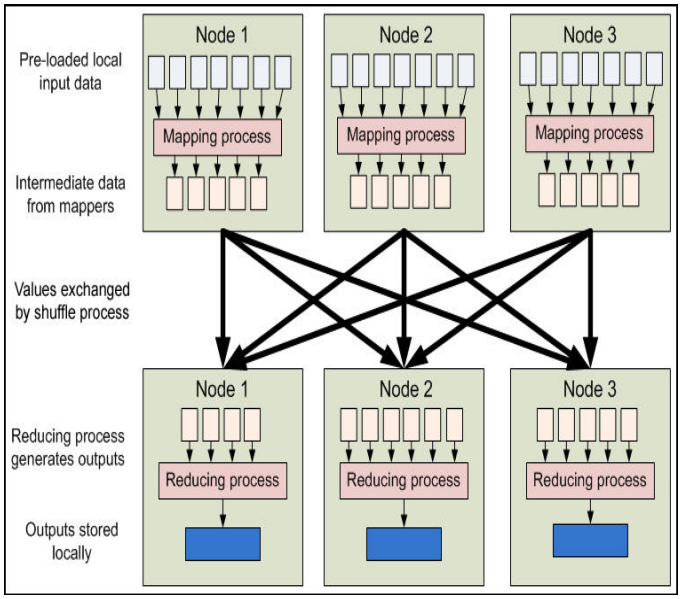

Figure 2: Distributed Map and Reduce processes

III. SYSTEM ARCHITECTURE

The system architecture comprises of hadoop architecture, hadoop multi-node cluster setup, setup of HDFS and implementation of Map Reduce programming work to solve the data intensive problem.

A. HDFS Architecture [6]

As show in Figure 3, an HDFS cluster consists of a single NameNode, a master server that manages the file system

namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manages storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNode. NameNode determines the mapping of blocks to Datanodes.
HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks.

B. Hadoop Cluster High Level Architecture [8]

Hadoop cluster comprises of a single master and multiple slaves or "worker nodes". The JobTracker is the service within Hadoop that farms out MapReduce tasks to specific nodes in the cluster, ideally the nodes that have the data, or at least are in the same rack.

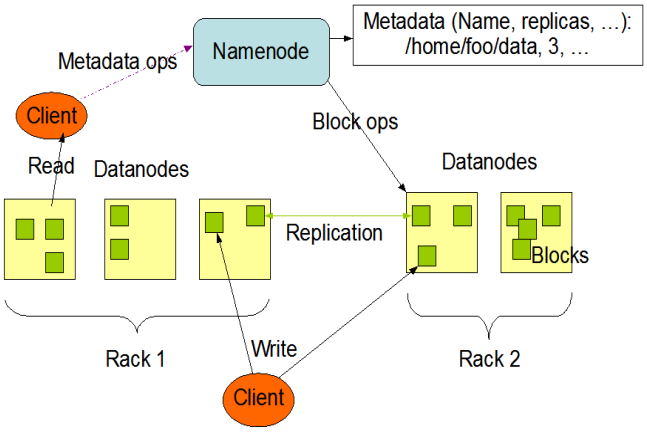

Figure 3: HDFS Architecture

A TaskTracker is a node in the cluster that accepts tasks - Map, Reduce and Shuffle operations - from a JobTracker. The master node consists of a JobTracker, TaskTracker, NameNode, and DataNode. A slave or worker node acts as both a DataNode and TaskTracker. In a larger cluster, the HDFS is managed through a dedicated NameNode server to host the file system index, and a secondary NameNode that can generate snapshots of the name node's memory structures, thus preventing file system corruption and reducing loss of data.

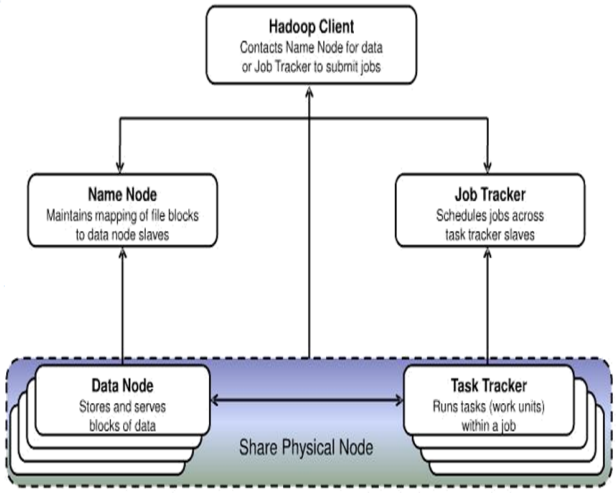

Figure 4: Hadoop high-level architecture

IV. EXPERIMENTAL SETUP

For performing the big data experiments, setup of Hadoop data cluster comprising of four nodes and Hadoop Distributed File System (HDFS) for storage was used. Before moving to multi-node cluster, single node cluster was first configured and tested. Hadoop has too many configuration parameters to describe here, but the most relevant for the purpose of this evaluation is the number of concurrent Map and Reduce tasks that are allowed to run on each node. We configured our cluster to run eight concurrent tasks per server. Each Map/Reduce program that is run is partitioned into M map tasks and R reduce tasks. Input and output data for the Map/Reduce programs is stored in HDFS, while input and output data for the data-parallel stack-based implementation is stored directly on the local disks.One node was configured as Master node and other nodes were designated as slave nodes. The master node runs the "master" daemons: NameNode for the HDFS storage layer and JobTracker for the MapReduce processing layer. The slave nodes run the "slave" daemons: DataNode for the HDFS layer and TaskTracker for MapReduce processing layer. The master node was also used as slave node to increase the processing nodes. The software used for master and slave nodes was Sun Java 1.6, Ubuntu Linux 10.04 LTS and hadoop 1.0.3.

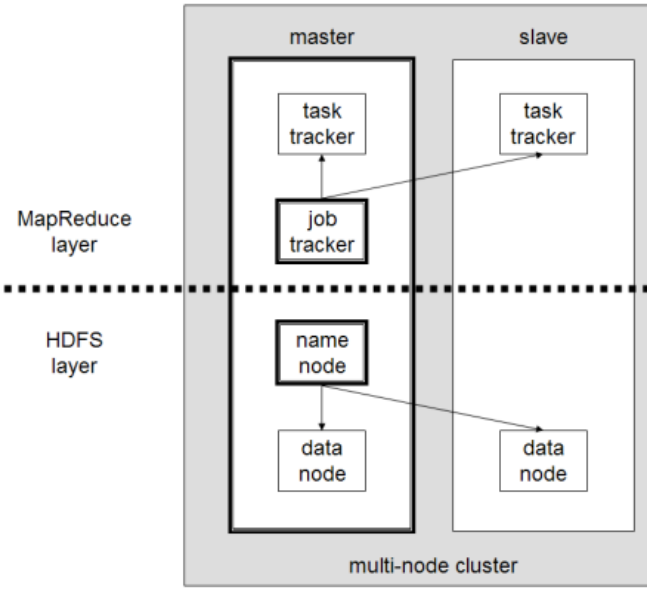

Figure 5: Hadoop multi-node cluster setup

V. EXPERIMENT AND RESULTS

A. Text processing application

The first experiment was text processing word count experiment to count the number of words that occur within a set of large sized documents. The map function splits each document into words and outputs each word together with the digit "1." The output records are therefore of the form (word, 1). The MapReduce framework groups all the records with the same key (i.e., word) and feeds them into the reduce function. The reduce function sums the input values and outputs the word and the total number of occurrences in the document(s).

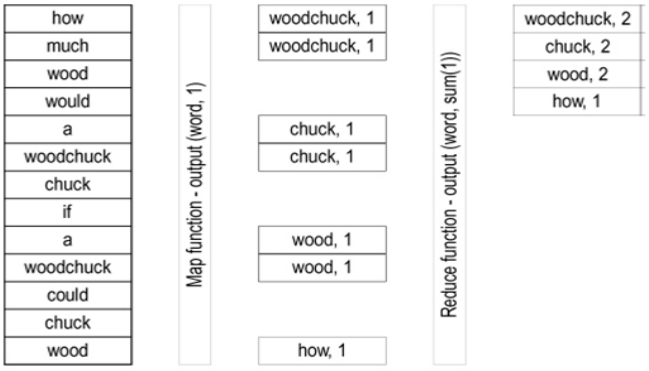

Figure 6: Map Reduce for word count

1) Experiment with increase in number of nodes

Dataset: Size of files used = 100 Mb

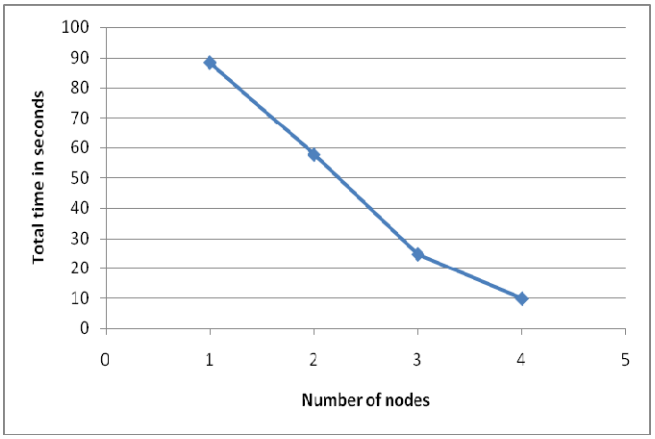

Figure 7: Execution time with varying number of nodes

2) Experiment with increase in size of dataset and nodes

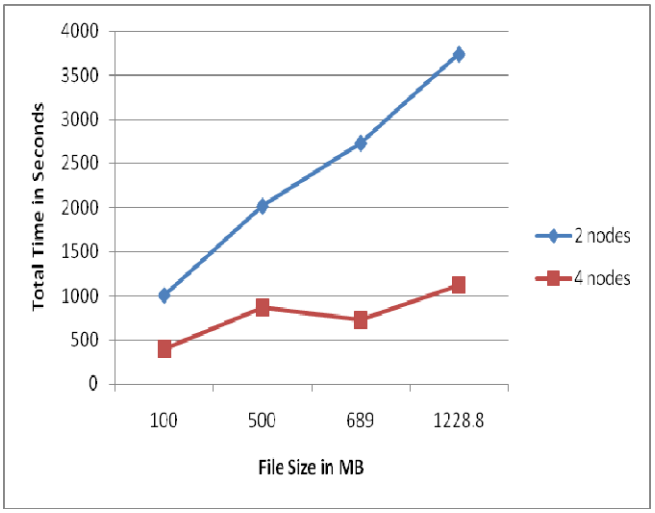

Figure 8: Execution time with varying dataset and nodes

The above result indicates the execution time decreases with increase in number of nodes used in Hadoop cluster.

B. Earthquake Data Analysis

In the second experiment, we have analyzed the earth quake data published by U.S. Geological Survey (USGS). The earth quake data is available in the form of CSV (comma-separated values) files published at periodic intervals. The analysis of earth quake data provides answer to where the earthquakes occurred by location, number of earth quakes on particular date.

1) Experiment with increase in number of nodes

Dataset - 7 day Earthquake Report of USGS

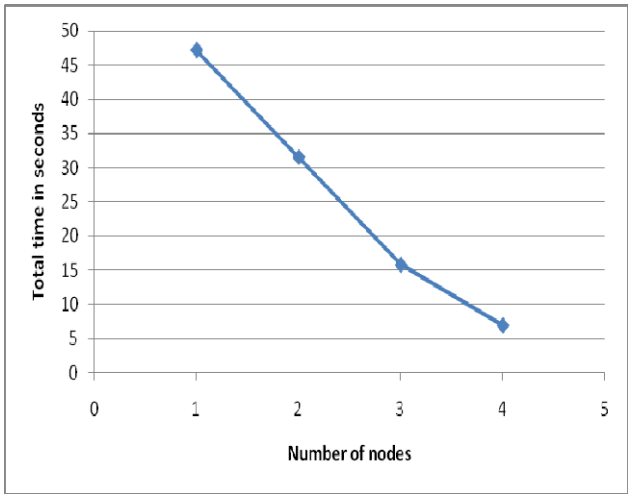

Figure 9: Quake analysis – No. of nodes v/s Execution time

2) Experiment with increase in size of dataset and nodes

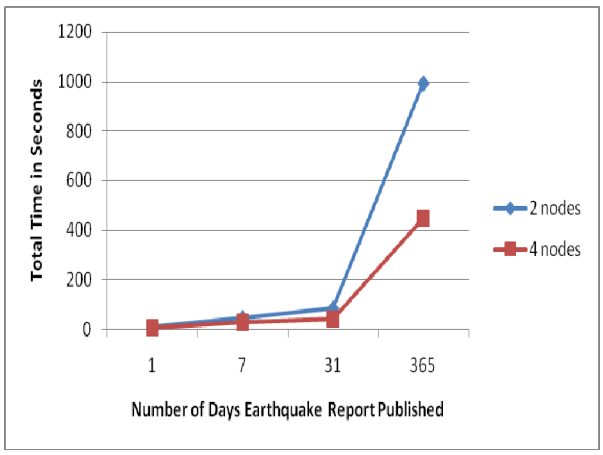

Figure 10: Quake analysis – No. of days report v/s Execution time

The above experiment result indicates that Hadoop cluster is scalable to support increased dataset and takes less job execution time with increase in number of nodes.

VI. CONCLUSION AND FUTURE WORK

In this work, we have explored the solution to big data problem using Hadoop data cluster, HDFS and Map Reduce programming framework using big data prototype application scenarios. The results obtained from various experiments indicate favorable results of above approach to address big data problem. Future work will focus on performance evaluation and modeling of hadoop data-intensive applications on cloud platforms like Amazon Elastic Compute Cloud (EC2).

VII. REFERENCES

[1] Impetus white paper, March, 2011, "Planning Hadoop/NoSQL Projects for 2011" by Technologies, Available: http://www.techrepublic.com/whitepapers/planning-hadoopnosql-projects-for-2011/2923717, March, 2011.

[2] McKinsey Global Institute, 2011, Big Data: The next frontier for innovation, competition, and productivity, Available: www.mckinsey.com/~/media/McKinsey/dotcom/Insights%20and%20pubs/MGI/Research/Technology%20and%20Innovation/Big%20Data/MGI_big_data_full_report.ashx, Aug, 2012.

[3] Thomas Herzog, Associate Commissioner, New York State, Thomas Kooy, IJIS Institute Big Data and the Cloud, IJIS Institute Emerging Technologies, Available: http://www.correctionstech.org/meeting/2012/Presentations/Red_01.pdf, Aug, 2012.

[4] Jacobs, A., The Pathologies of Big Data, ACM Queue, Available: http://queue.acm.org/detail.cfm?id=1563874, 6th July 2009.

[5] Apache Software Foundation. Official apache hadoop website, http://hadoop.apache.org/, Aug, 2012.

[6] The Hadoop Architecture and Design, Available: http://hadoop.apache.org/common/docs/r0.16.4/hdfs_design.html, Aug, 2012

[7] Hung-Chih Yang, Ali Dasdan, Ruey-Lung Hsiao, and D. Stott Parker from Yahoo and UCLA, "Map-Reduce-Merge: Simplified Data Processing on Large Clusters", paper published in Proc. of ACM SIGMOD, pp. 1029–1040, 2007.

[8] White, Tom. Hadoop The Definitive Guide 2nd Edition. United States : O'Reilly Media, Inc., 2010.