GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

HADOOP:FRAMEWORK TO STORE AND PROCESS BIG DATA

Deokar Anuja Tushar*[1], Kuchik Ashwini Shantaram[2], Jadhav Shweta Chandrakant[3] & Shejwal Asmita Sanjay[4]
*[1,2,3&4] Department of Computer Engineering, JCEI's Jahind College of Engineering Pune,India

ABSTRACT

Electric load forecasting in summer season is an important task do to avoid any irregularity in the power system .This paper provides a study on the short term load forecasting for summer season. Load Data for the study is collected from Madhya Pradesh Poorva Kshetra Vidyut Vitaran Company Ltd of the summer season i.e. from March2014 –June 2014 and March 2015 –June 2015 Forecasted load is calculated for the 1 July 2016.Different parameters affecting the ELF such as Temperature ,Humidity , hourly load etc. are incorporated in the methods to get a most appropriate model with least error. Also separate study is done by using only certain parameter at each time.An analysis is done with the Artificial Neural Network(ANN) and Regression method using MATLAB 13.0 to get a better model for Short Term Load Forecasting (STLF) .Result shows that ANN gives the result with Mean Absolute Percentage Error (MAPE) 1.267% and MAPE for Regression Analysis is 2.623%. Further the results are compared and shown both graphically and in tabular form

Keywords: *Load Forecasting, Artificial Neural Network (ANN), Regression Analysis, BackPropagation Method, MAPE*

I. INTRODUCTION

Due to the arrival of new technologies, devices, and communication means like social networking sites, the amount of data produced by mankind is growing rapidly every year. The amount of data produced by us from the commencement of time till 2003 was 5 billion gigabytes. If we pile up the data in the form of disks it may fill an entire football field. The same amount was generated in every two days in 2011, and in every ten minutes in 2013. The rate is still growing enormously. Though all this information produced is meaningful and can be useful when processed, it is being neglected.Big data means massively big data, it is a collection of large datasets that cannot be processed using traditional computing techniques. Big data is not merely a data, instead it has become a complete subject, which involves various tools, technqiues and frameworks.Over the past few years several designs, prototypes, methodologies have been developed to tackle parallel computing problems. Furthermore, specially designed servers were customised to meet the parallel computing requirements. The major problem was, these servers were too expensive to handle and yet did not produce expected results. With the arrival of multi core processors and virtualization technology, the problems seems to be diminishing and hence effective and powerful tools are built to achieve parallelization using the commodity machines. Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

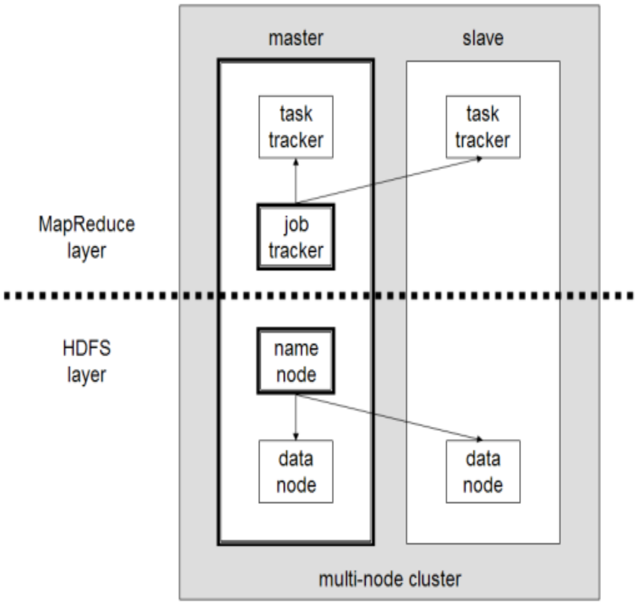
II. HADOOP ARCHITECTURE

Hadoop comprises of the Hadoop Common package, which provides file system and OS level abstractions, a MapReduce engine (either MapReduce/MR1 or YARN/MR2) and the Hadoop Distributed File System (HDFS). The Hadoop Common package contains the necessary Java ARchive (JAR) files and scripts needed to start Hadoop.

For efficient scheduling of work, every Hadoop-compatible file system must provide location awareness: the name of the rack (more precisely, of the network switch) where a worker node is present. Hadoop applications can use the information to execute code on the node where the data is, and, failing that, on the same rack/switch to decrease backbone traffic. HDFS uses this method when creating duplicate data for data redundancy across multiple racks.

This paper was presented in Technophilia 2017 at Jaihind Polytechnic, A/P. Kuran, Tal. Junnar, Dist. Pune

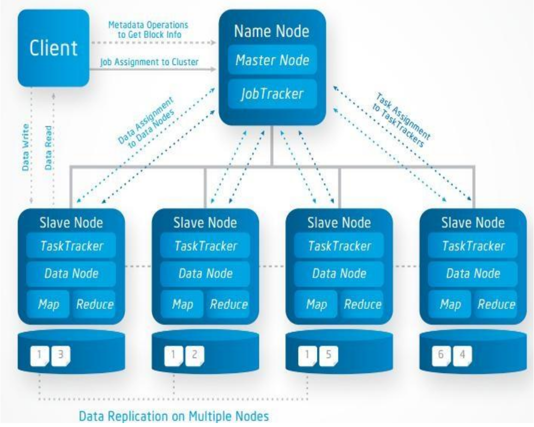
The approach brings down the impact of a rack power outage or switch failure; if one of these hardware failures occurs, the data will be available.Vasic function of MapReduce and Hadoop Distributed file system are:Hadoop Distributed File System : A distributed file system that provides high-throughput access to application data.Hadoop MapReduce: This is YARN-based system for parallel processing of large data sets.



Map Reduce Layer

Hadoop MapReduce is a software framework for easily writing applications which process huge amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a dependable, fault-tolerant manner.

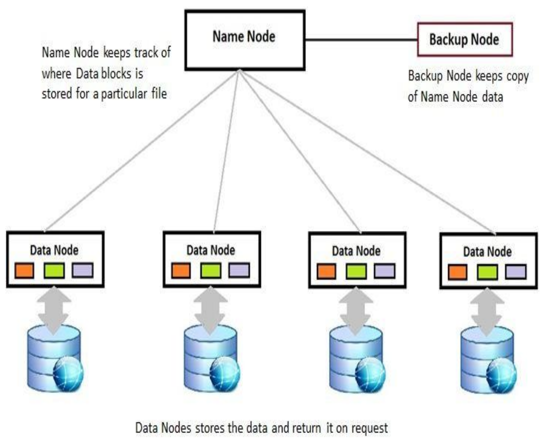
The MapReduce framework consists of a individual master JobTracker and one slave TaskTracker per cluster-node. The master is accountable for resource management, tracking resource consumption/availability and scheduling the jobs component tasks on the slaves, monitoring the job and re-executing the unsuccessful tasks. The slaves TaskTracker execute the tasks as instructed by the master and provide task-status information to the master periodically.Job Tracker is a service in Hadoop which guides the mapreduce task to point to specific data node that has desirable data required to process through mapreduce task.



Client applications put forward jobs to the Job tracker. The JobTracker talks to the Name Node to find out the location of the data. The Job Tracker give the work to the selected Task Tracker nodes. The Job Tracker situate Task Tracker nodes with available slots at or nearby the data. The Task Tracker nodes are monitored. If they do not give heartbeat signals often enough, they are consider to have failed and the work is scheduled on a different Task Tracker. A Task Tracker will inform the Job Tracker when a task fails. The Job Tracker decides what to do then that is it may resubmit the job elsewhere or it may mark that particular record as thing to avoid, and it may may even blacklist the Task Tracker as untrust worthy. When the work is accomplished, the Job Tracker modify its status. Client applications can poll the Job Tracker for information. The Job Tracker is a element of failure for the Hadoop MapReduce service. If it goes down, all continual jobs are halted.

Hadoop Distributed File System Layer

The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file system that is written in Java for the Hadoop framework. Some consider HDFS to rather be a data store due to its lack of POSIX compliance and unfitness to be mounted, but it does give shell commands and Java API methods that are similar to other file systems.A Hadoop cluster has nominally a individual namenode and a cluster of datanodes, although redundancy options are available for the namenode due to its criticalness. Each datanode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses TCP/IP sockets for communication. Clients use remote procedure call (RPC) to communicate between each other.HDFS uses a master/slave architecture where master consists of a single NameNode that carry off the file system metadata and one or more slave DataNodes that store the actual data.A file in an HDFS namespace is split into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes takes care of read and write operation with the file system. They also take care of block creation, deletion and replication depending on instruction given by NameNode.NameNode stores MetaData(No of Blocks, On Which Rack which DataNode the data is stored and other details) about the data being stored in DataNodes whereas the DataNode stores the actual Data. In a multinode cluster NameNode and DataNodes are normally on different machines. There is only one NameNode in a cluster and many DataNodes; Thats why we call NameNode as a single point of failure. Although there is a Secondary NameNode (SNN) that can exist on different machine which doesn't actually act as a NameNode but stores the image of primary NameNode at certain checkpoint and is also used as backup to restore NameNode.


Name Node
Backup Node
Name Node keeps track of where Data blocks is stored for a particular file
Backup Node keeps copy of Name Node data
Data Node
Data Node
Data Node
Data Node
Data Nodes stores the data and return it on request

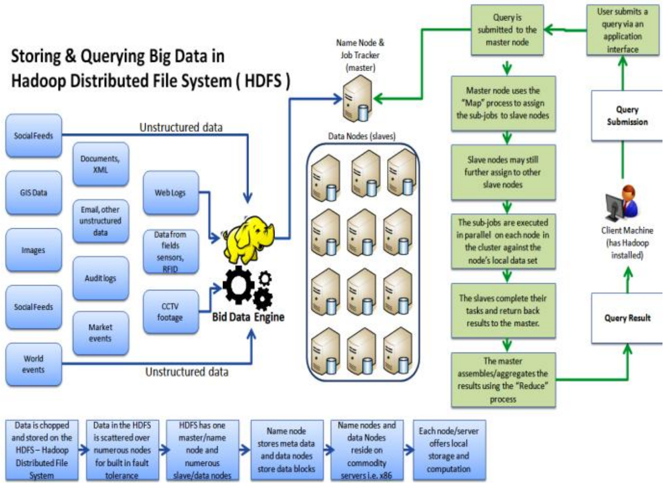
III. COMPLETE WORKING

Stage 1

A user/application can assign a job to the Hadoop (a hadoop job client) for required process by specifying the following items:The location of the input and output files in the distributed file system.The java classes in the form of jar file having the implementation of map and reduce functions.The job configuration by setting various parameters specific to the job.

Stage 2
The Hadoop job client then submits the job (jar/executable etc) and configuration to the JobTracker which then carry on the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

Stage 3
The TaskTrackers on various nodes execute the task as per MapReduce implementation and output of the reduce function is stored into the output files on the file system.


Storing & Querying Big Data in
Hadoop Distributed File System (HDFS)
Social Feeds
GIS Data
Images
Social Feeds
World events
Documents, XML
Email, other unstructured data
Audit logs
Market events
Web Logs
Data from fields sensors, RFID
CCTV footage
Unstructured data
Unstructured data
Bid Data Engine
Name Node & Job Tracker (master)
Data Nodes (slaves)
Query is submitted to the master node
User submits a query via an application interface
Master node uses the "Map" process to assign the sub-jobs to slave nodes
Slave nodes may still further assign to other slave nodes
The sub-jobs are executed in parallel on each node in the cluster against the node's local data set
The slaves complete their tasks and return back results to the master.
The master assembles/aggregates the results using the "Reduce" process
Query Submission
Client Machine (has Hadoop installed)
Query Result
Data is chopped and stored on the HDFS – Hadoop Distributed File System
Data in the HDFS is scattered over numerous nodes for built in fault tolerance
HDFS has one master/name node and numerous slave/data nodes
Name node stores meta data and data nodes store data blocks
Name nodes and data Nodes reside on commodity servers i.e. x86
Each node/server offers local storage and computation

IV. ADVANTAGES

Scalable
Hadoop is a extremely scalable storage platform, because it can store and distribute huge data sets across hundreds of inexpensive servers that operate in parallel. Different from traditional relational database systems (RDBMS) that can't scale to process big amounts of data, Hadoop allows businesses to run applications on thousands of nodes involving thousands of terabytes of data.

Cost effective
Hadoop also supply a cost effective storage solution for businesses' exploding data sets. The problem with traditional relational database management systems is that it is very high cost prohibitive to scale to such a degree in order to manage such massive volumes of data. In an effort to decrease costs, many companies in the past would have had to down-sample data and categorize it depending on certain assumptions as to which data was the most valuable. The raw data would be erased, as it would be too cost-prohibitive to support. While this approach may have had worked in the short term, this meant that when business precedence changed, the complete raw data set was not available, as it was really expensive to store. Hadoop, on the other hand, is organized as a scale-out architecture that can affordably store all of a company's data for later use. The cost savings are staggering: instead of costing thousands to tens of thousands of pounds per terabyte, Hadoop gives computing and storage capabilities for hundreds of pounds per terabyte.

Flexible
Hadoop allows businesses to easily access new data sources and tap into various types of data (both structured and unstructured) to create value from that data. This means businesses can use Hadoop to derive valued business

insights from data reference such as social media, email conversations or click stream data. Also, Hadoop can be used for a large variety of purposes, such as log processing, recommendation systems, data warehousing, market campaign analysis and fraud detection.

Fast

Hadoop's unique storage method is based on a distributed file system that fundamentally 'maps' data wherever it is situated on a cluster. The tools for data processing are frequently on the same servers where the data is placed, resulting in much faster data processing. If you're dealing with huge volumes of unstructured data, Hadoop is able to with efficiency process terabytes of data in just minutes, and petabytes in hours.

Resilient to failure

A major advantage of Hadoop is its fault tolerance. When data is dispatched to an individual node, that data is also duplicated to other nodes in the cluster, which means that in the event of failure, there is another copy accessible for use.The MapR distribution goes on the far side that by eliminating the NameNode and replacing it with a distributed No NameNode architecture that furnish true high availability. Our architecture provides protection from both single and multiple failures.When it comes to handling massive data sets in a risk-free and cost-effective manner, Hadoop has the advantage over relational database management systems, and its value for any size business will carry on to increase as unstructured data continues to grow..

REFERENCES

1) *SIGMOD, 2009.*
2) *Apache Hadoop: http://Hadoop.apache.org*
3) *Dean, J. and Ghemawat, S., DeWitt & Stonebraker, "MapReduce: A major step backwards", 2008.*
4) *Hadoop Distributed File System, http://hadoop.apache.org/hdfs*
5) *HadoopTutorial: http://developer.yahoo.com/hadoop/tutorial/module1.html*
6) *J. Dean and S. Ghemawat, "Data Processing on Large Cluster", OSDI '04, pages 137–150, 2004*
7) *J. Dean and S. Ghemawat,"MapReduce: Simplified Data Processing on Large Clusters", p.10, (2004*
8) *Jean-Pierre Dijcks, "Oracle: Big Data for the Enterprise", 2013.*
9) *http://hadoopguru.blogspot.in/2013/02/hadoop-distributed-file-system-hdfs.html*
10) *http://a4academics.com/tutorials/83-hadoop/835-hadoop-architecture*
11) *http://www.itproportal.com/2013/12/20/big-data-5-major-advantages-of-hadoop/*