Global Journal on Application of Data Science and Internet of Things
Vol 1, Issue 1 - 2017

BIGDATA ANALYTICS APPROACH FOR IMPLEMENTATION OF WEATHER FORCASTING SYSTEM

NITIN CHABRA

ABSTRACT

Present paper discuss the implementation of an weather forcasting system using Big-data analytics by using data which contains previous year weather details and study that data in order to predict the state of atmosphere for a particular location, and calculating deviation from actual result. It will help in reducing overhead satellite and sensor costs.

This will help us in easily forecasting weather based on different parameters like temperature, humidity etc. Thus reducing high cost .it will also led to less research, less manpower in predicting future atmospheric conditions. It will also provide platform for future research in this field.

KEYWORDS: Big Data, Weather Forecasting, Cloud Computing, Hadoop, Hadoop Distributed File System (HDFS).

INTRODUCTION

In today scenario climate change is a problem, whatever its causes. it randomness of getting effected by several factors and parameter, this factors forces us to perform some mysterious or complex mathematical calculations which makes it's even harder to analyze and interpret, but even to understand the impact at local level, we need more than back-of-the napkin mathematics, so that is where we need big-data technology. In this paper we used Big Data for weather forecasting.

Big-Data can be understood as blanket term for any collection of data set so large and complex that it becomes difficult to process using on-hand data management tools or traditional data processing applications. Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process the data within a tolerable elapsed time. Big data sizes are a constantly moving target, as of 2012 ranging from a few dozen terabytes to many petabytes of data in a single data set.

WHY BIG-DATA FOR PRESENT WORK

Our aim is to tell people about their climate locally in ways they can understand, and the only way to do that is with big data analysis, its allows to say thing in simple and clear way.

It will be able to take data from any source, harness relevant data and analyze it to find answers that enable

1) Cost reductions,
2) Time reductions,
3) New product development and optimized offerings, and
4) Smarter business decision making.

OVERALL DESCRIPTION AND FEATURES:

PRODUCT PERSPECTIVE

During this we will gather data from different sources and store then in different set ,to analyze data we divide it into different node which is further divided as data node and task tracker known as slaves, which are controlled by master node, master node consists of job tracker, name node and secondary name node. This slave analyze in parallel all the data and report to master node, then reported result will be store in some file. We will then compare this reported result with actual result and will notice the deviation from actual result, then try to minimize the deviation by performing different operation.

TECHNOLOGY USED

The strength of any project depends upon the technology on which the project is based. Today we are living in a world where technologies are evolving every day new technologies are taking an edge over the older ones. Every new technology is provides some new benefits, but only small part of them remains in the competitive world. HADOOP is the latest technology, which is in use nowadays and has proved to be the most reliable way to use Big Data.

WHY HADOOP?

It's the explosion of the era of Big Data, companies now need to leverage all their available data in full to drive the decisions that will support their future growth. The scale and variety of this data now challenges Relational Databases and that is where Hadoop can really add benefit.

Hadoop development is hosted by the Apache OpenSource Community and all major technology companies have contributed to the codebase enabling Hadoop to leap ahead of proprietary solutions in a relatively short time period.

Hadoop's main components are its file system (HDFS) that provides cheap and reliable data storage, and its MapReduce engine that provides high performance parallel data processing. In addition Hadoop is self-managing and can easily handle hardware failures as well as scaling up or down its deployment without any change to the codebase.

The team utilized several communication methods to collaborate with each other; including email, instant web meeting technology, voice calls and voicemail. The team also used both email as well as Microsoft OneNote to keep notes on meetings, and other conversations.

We utilized regular meetings with both the customer as well as the business practitioner to solicit feedback for improvements and changes.

HADOOP COMPONENTS

PROCESSING: MAPREDUCE

HadoopMapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

A MapReducejob usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the *reduce tasks*. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks. The MapReduce framework consists of a single master JobTracker and one slave Task-tracker per cluster-node.

The MapReduce framework operates exclusively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.

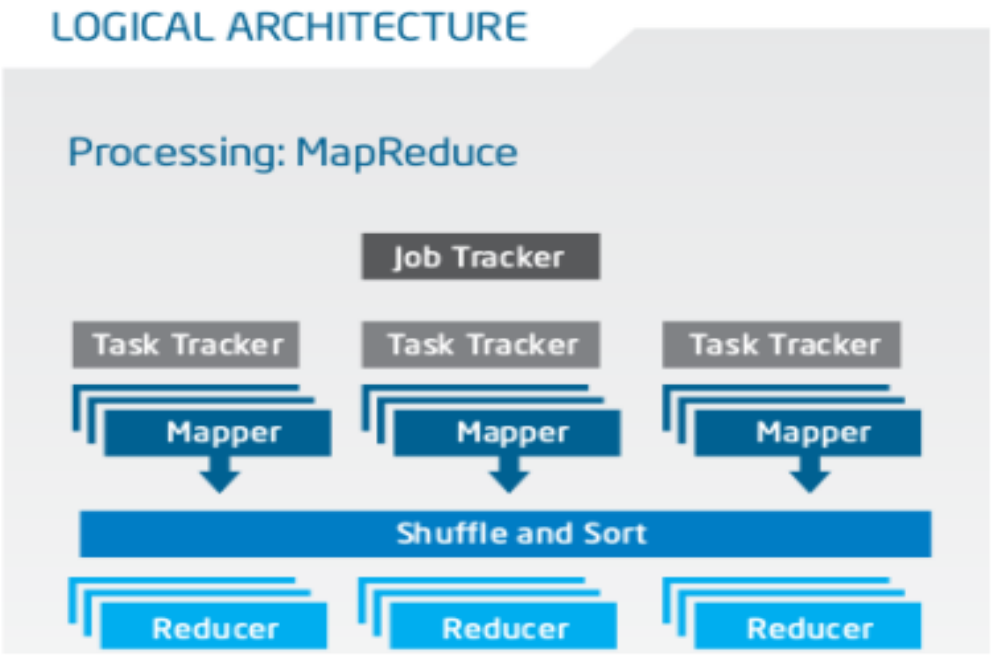

Figure 1.Logical Architecture

STORAGE: HDFS

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is now an Apache Hadoop subproject.

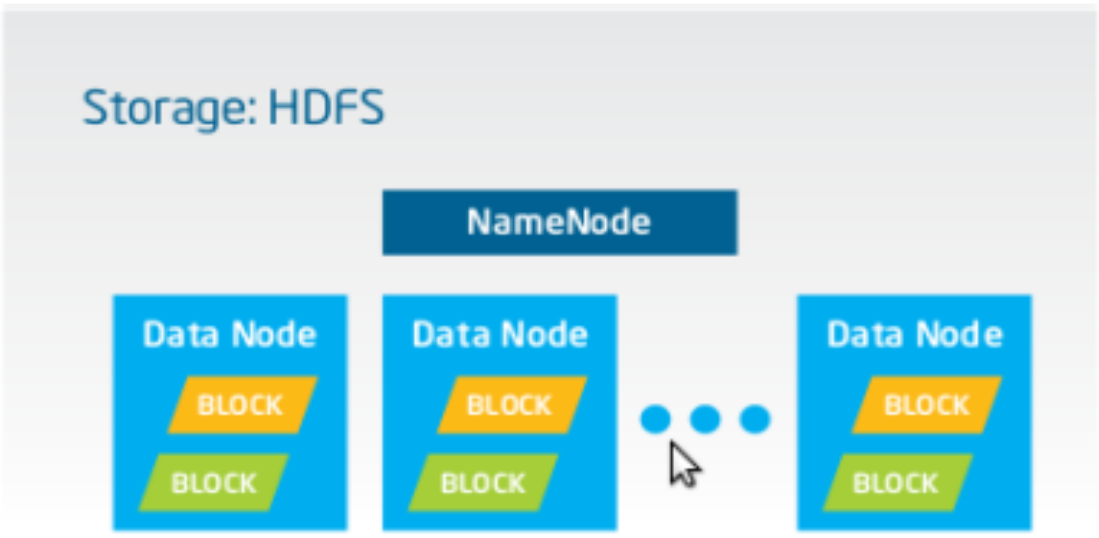

Figure 2.Storage HDFS

- **NAMENODEMETADATA**

Meta-data in Memory

– The entire metadata is in main memory

– No demand paging of meta-data

Types of Meta data

– List of files

– List of Blocks for each file

– List of Data Nodes for each block

– File attributes, e.g creation time, replication factor

A Transaction Log

– Records file creations, file deletions. Etc

➢ DATANODE

A Block Server

– Stores data in the local file system (e.g. ext3)

– Stores meta-data of a block (e.g. CRC32)

– Serves data and meta-data to Clients

– Periodic validation of check-sums

Block Report

Periodically sends a report of all existing blocks to the Name Node

PROCESS FLOW

Hadoop Jobs is an Oracle application that statistically processes Analytics data and stores the results in the Analytics database.In a functional Analytics installation, raw site visitor data is continuously captured by the Analytics Sensor (Data Capture Application), which then stores the data to the local file system. The raw data in the file system is called on periodically by the Hadoop Distributed File System (HDFS) Agent, which then copies the raw data to the Hadoop Distributed File System, where Hadoop jobs process the data. Hadoop jobs consist of locations and Oracle-specific processors that read site visitor data in one location, statistically process that data, and write the results to another location for pickup by the next processor. When processing is complete, the results (statistics on the raw data) are injected into the Analytics database.

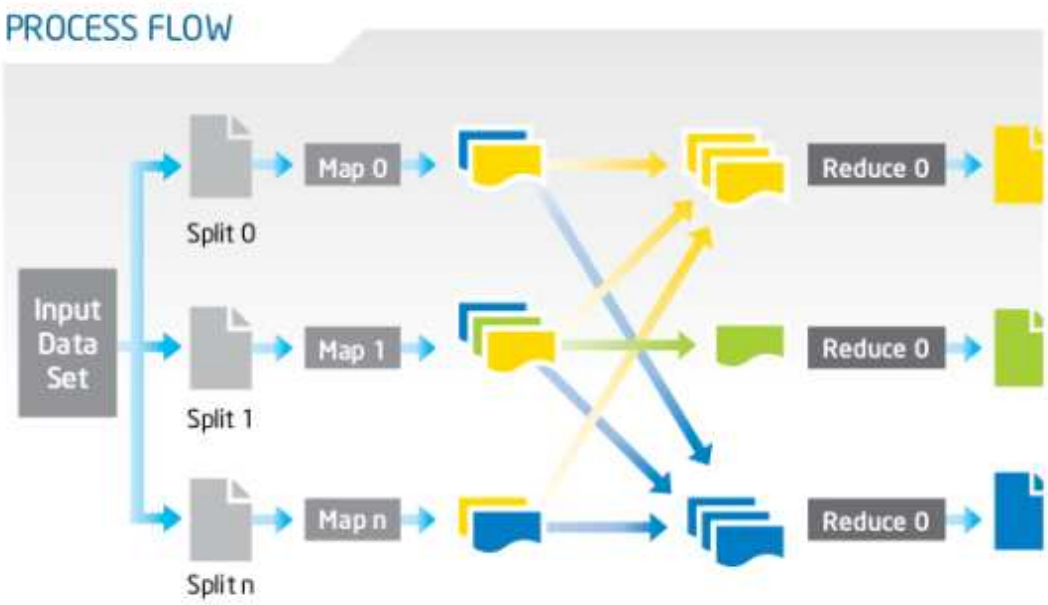

Figure 3.Process Flow

DATABASE DESIGN AND IMPLEMENTATION CONSTRAINTS

If you expect your Hadoop cluster to grow beyond 20 machines, we recommend that the initial cluster be configured as if it were to span two racks, where each rack has a top-of-rack 10 GigE switch. As the cluster grows to multiple racks, you will want to add redundant core switches to connect the top-of-rack switches with 40GigE. Having two logical racks gives the operations team a better understanding of the network requirements for intra-rack and cross-rack communication.

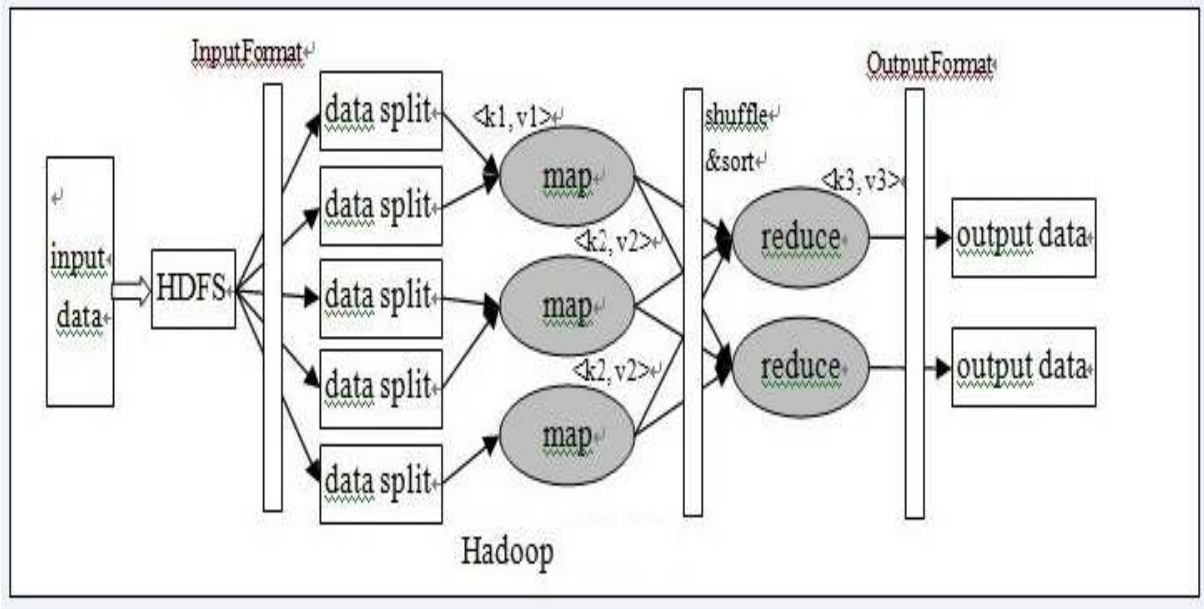

Figure 4.DATA FLOW DIAGRAM

OTHER NONFUNCTIONAL REQUIREMENTS

PERFORMANCE REQUIREMENTS

We still have to manage it for optimum performance. Balancing capacity for workloads and expectations is still important with Hadoop. The key to this is designing a balanced system that keeps the processors as busy as possible over extended amounts of time, keeping system overheads to a minimum, and providing low-latency interconnection and just-in-time highly parallel data movement to and from the processors. Quality of data movement is also important – one retry on a block of data can delay work getting to hundreds of processors

SAFETY REQUIREMENTS

We make duplicate copy of data for master node and store it at different location in case of master node failure. Master node make three copies of data before sending it to slave node,master node also maintain data index for each and every slave node,so in case of slave node failure it can be easily retract and restore data .

SECURITY REQUIREMENTS

When come to security we need to make sure that there should not be any data breach like:

->Data Transfer

->Data Storage

->Data Analyzing

→ Data distribution

->Employee mishandling

->Any business failure.

FUTURE TRADE: APACHE HIVEAND APACHE PIG

Apache Hive and Apache Pig are programming languages that simplify development of applications employing the Map Reduce framework. HiveQL is a dialect of SQL and supports a subset of the syntax. Although slow, Hive is being actively enhanced by the developer community to enable low-latency queries on Apache HBase and HDFS. Pig Latin is a procedural programming language that provides high-level abstractions for MapReduce. You can extend it with User Defined Functions written in Java, Python, and other languages.

REFERENCES

[1]. Alessi, L., Barigozzi, M., and Capasso, M. (2009). Forecasting Large Datasets with Conditionally Heteroskedastic Dynamic Common Factors. Working Paper No. 1115, European Central Bank.

[2]. Hassani, H., Saporta, G., and Silva, E. S. (2014). Data Mining and Official Statistics: The Past, The Present & The Future. Big Data, 2(1), BD1-BD10.

[3]. Hyndman, R. J. and Athanasopoulos, G. (2013). Forecasting: Principles and Practice. Otexts, Australia.

[4]. Jadhav, D. K. (2013). Big Data: The New Challenges in Data Mining. International Journal of Innovative Research in Computer Science & Technology, 1(2), pp. 39-42.

[5]. Bacon, T. (2013). Big Bang? When 'Big Data' gets too Big. Available via: http:// www.eyefortravel.com/mobile-and-technology/big-bang-when-%E2%80%98big-data%E2%80%99-gets-too-big.

