ISSN No. 0976-5697

Volume 4, No. 6, May 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

A Review: Oracle Big Data Aplliance

Vivek D. Honule
Department of Information Technology
Datta Meghe Collage of Engineering
Mumbai, India
vivekhonule@gmail.com

Mrunali G. Bukkawar
Department of Information Technology
Jawaharlal Darda Institute of Engineering and Technology
Yavatmal, India
Mrunal.58@gmail.com

Abstract: Big data, which refers to the data sets that are too big to be handled using the existing database management tools, are emerging in many important applications, such as Internet search, business informatics, social networks, social media, genomics, and meteorology. Big data presents a grand challenge for database and data analytics research. The challenges include capture, curation, storage, search, sharing, analysis, and visualization. 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. Big data is difficult to work with using most relational database management systems and desktop statistics and visualization packages, requiring instead "massively parallel software running on tens, hundreds, or even thousands of servers".

Keywords: Big Data, Data analytics, terabyte, petabyte, data set, server, social network, social media.

I. INTRODUCTION

Big Data is large in quantity, is captured at a rapid rate, and is structured or unstructured, or some combination of the above. These factors make Big Data difficult to capture, mine, and manage using traditional methods. Big data typically refers to the following types of data:

Traditional Enterprise Data:

Includes customer information from CRM systems, transactional ERP data, web store transactions, general ledger data.

Machine Generated/Sensor Data:

Includes Call Detail Records ("CDR"), weblogs, smart meters, manufacturing sensors, equipment logs (often referred to as digital exhaust), trading systems data.

Social Data:

Includes customer feedback streams, micro-blogging sites like Twitter, social media platforms like Face book

II. ADVANTAGE OF BIG DATA

Information gleaned from nontraditional sources such as blogs, social media, email, sensors, photographs, video footage, etc., and therefore typically unstructured and voluminous—holds the promise of giving enterprises deeper insight into their customers, partners, and business. This data can provide answers to questions they may not have even thought to ask. What's more, companies benefit from a multidimensional view of their business when they add insight from big data to the traditional types of information they collect and analyze. For example, a company that operates a retail Web site can use big data to understand site visitors' activities, such as paths through the site, pages viewed, and comments posted. This knowledge can be combined with purchasing history and stored in a corporate relational database. From this, the company gains a better understanding of customers, and can fine-tune offers to target their interests.

III. EXAMPLES OF BIG DATA

Examples include Big Science, web logs, RFID, sensor networks, social networks, social data (due to the social data revolution), Internet text and documents, Internet search indexing, call detail records, astronomy, atmospheric science, genomics, biogeochemical, biological, and other complex and often interdisciplinary scientific research, military surveillance, medical records, photography archives, video archives, and large-scale e-commerce.

IV. CHARACTERISTICS THAT DEFINE BIG DATA

A. Volume:

Machine-generated data is produced in much larger quantities than non-traditional data. For instance, a single jet engine can generate 10TB of data in 30 minutes. With more than 25,000 airline flights per day, the daily volume of just this single data source runs into the Peta bytes. Smart meters and heavy industrial equipment like oil refineries and drilling rigs generate similar data volumes, compounding the problem.

B. Velocity:

Social media data streams – while not as massive as machine-generated data – produce a large influx of opinions

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

and relationships valuable to customer relationship management. Even at 140 characters per tweet, the high velocity (or frequency) of Twitter data ensures large volumes (over 8 TB per day).

C. Variety:

Traditional data formats tend to be relatively well described and change slowly. In contrast, non-traditional data formats exhibit a dizzying rate of change. As new services are added, new sensors deployed, or new marketing campaigns executed, new data types are needed to capture the resultant information.

D. Value:

The economic value of different data varies significantly. Typically there is good information hidden amongst a larger body of non-traditional data; the challenge is identifying what is valuable and then transforming and extracting that data for analysis.

V. IMPORTANCE OF BIG DATA

To make the most of big data, enterprises must evolve their IT infrastructures to handle the rapid rate of delivery of extreme volumes of data, with varying data types, which can then be integrated with an organization's other enterprise data to be analyzed. When big data is distilled and analyzed in combination with traditional enterprise data, enterprises can develop a more thorough and insightful understanding of their business, which can lead to enhanced productivity, a stronger competitive position and greater innovation – all of which can have a significant impact on the bottom line.

Retailers usually know who buys their products. Use of social media and web log files from their ecommerce sites can help them understand who didn't buy and why they chose not to, information not available to them today. This can enable much more effective micro customer segmentation and targeted marketing campaigns, as well as improve supply chain efficiencies.

Finally, social media sites like Face book and LinkedIn simply wouldn't exist without big data. Their business model requires a personalized experience on the web, which can only be delivered by capturing and using all the available data about a user or member.

VI. BUILDING A BIG DATA PLATFORM

As with data warehousing, web stores or any IT platform, an infrastructure for big data has unique requirements. In considering all the components of a big data platform, it is important to remember that the end goal is to easily integrate your big data with your enterprise data to allow you to conduct deep analytics on the combined data set.

A. Infrastructure Requirement:

The requirements in a big data infrastructure span data acquisition, data organization and data analysis.

B. Aquire Big Data:

The acquisition phase is one of the major changes in infrastructure from the days before big data. Because big data refers to data streams of higher velocity and higher variety, the infrastructure required to support the acquisition of big data must deliver low, predictable latency in both capturing data and in executing short, simple queries; be able to handle very high transaction volumes, often in a distributed environment; and support flexible, dynamic data structures.

NoSQL databases are frequently used to acquire and store big data. They are well suited for dynamic data structures and are highly scalable. The data stored in a NoSQL database is typically of a high variety because the systems are intended to simply capture all data without categorizing and parsing the data.

For example, NoSQL databases are often used to collect and store social media data. While customer facing applications frequently change, underlying storage structures are kept simple. Instead of designing a schema with relationships between entities, these simple structures often just contain a major key to identify the data point, and then a content container holding the relevant data. This simple and dynamic structure allows changes to take place without costly reorganizations at the storage layer.

C. Organize Data:

In classical data warehousing terms, organizing data is called data integration. Because there is such a high volume of big data, there is a tendency to organize data at its original storage location, thus saving both time and money by not moving around large volumes of data. The infrastructure required for organizing big data must be able to process and manipulate data in the original storage location; support very high throughput (often in batch) to deal with large data processing steps; and handle a large variety of data formats, from unstructured to structured.

Apache Hadoop is a new technology that allows large data volumes to be organized and processed while keeping the data on the original data storage cluster. Hadoop Distributed File System (HDFS) is the long-term storage system for web logs for example. These web logs are turned into browsing behavior (sessions) by running MapReduce programs on the cluster and generating aggregated results on the same cluster. These aggregated results are then loaded into a Relational DBMS system.

D. Analyze Big Data:

Since data is not always moved during the organization phase, the analysis may also be done in a distributed environment, where some data will stay where it was originally stored and be transparently accessed from a data warehouse. The infrastructure required for analyzing big data must be able to support deeper analytics such as statistical analysis and data mining, on a wider variety of data types stored in diverse systems; scale to extreme data volumes; deliver faster response times driven by changes in behavior; and automate decisions based on analytical models. Most importantly, the infrastructure must be able to integrate analysis on the combination of big data and

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

traditional enterprise data. New insight comes not just from analyzing new data, but from analyzing it within the context of the old to provide new perspectives on old problems.

VII. SOLUTION SPECTRUM

Many new technologies have emerged to address the IT infrastructure requirements outlined above. At last count, there were over 120 open source key-value databases for acquiring and storing big data, with Hadoop emerging as the primary system for organizing big data and relational databases expanding their reach into less structured data sets to analyze big data. These new systems have created a divided solutions spectrum comprised of:

a. Not Only SQL (NoSQL): solutions: developer-centric specialized systems
b. SQL solutions: the world typically equated with the manageability, security and trusted nature of relational database management systems (RDBMS)

NoSQL systems are designed to capture all data without categorizing and parsing it upon entry into the system, and therefore the data is highly varied. SQL systems, on the other hand, typically place data in well-defined structures and impose metadata on the data captured to ensure consistency and validate data types.

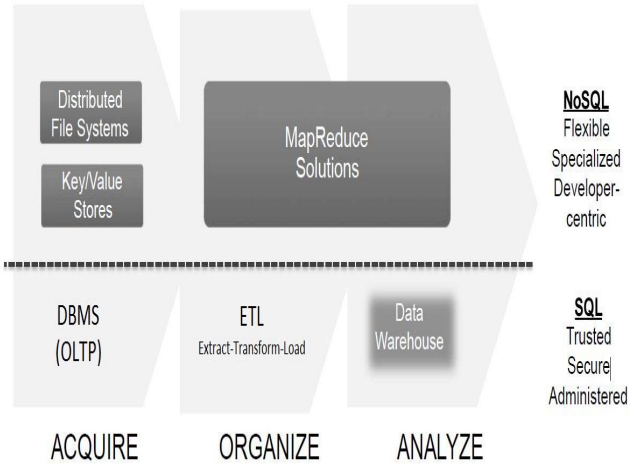

Figure .1 Divided solution spectrum

Distributed file systems and transaction (key-value) stores are primarily used to capture data and are generally in line with the requirements discussed earlier in this paper. To interpret and distill information from the data in these solutions, a programming paradigm called MapReduce is used. MapReduce programs are custom written programs that run in parallel on the distributed data nodes. After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

VIII. ORACLE'S BIG DATA SOLUTION

Oracle is the first vendor to offer a complete and integrated solution to address the full spectrum of enterprise big data requirements. Oracle's big data strategy is centered on the idea that you can evolve your current enterprise data architecture to incorporate big data and deliver business value, leveraging the proven reliability, flexibility and performance of your Oracle systems to address your big data requirements.

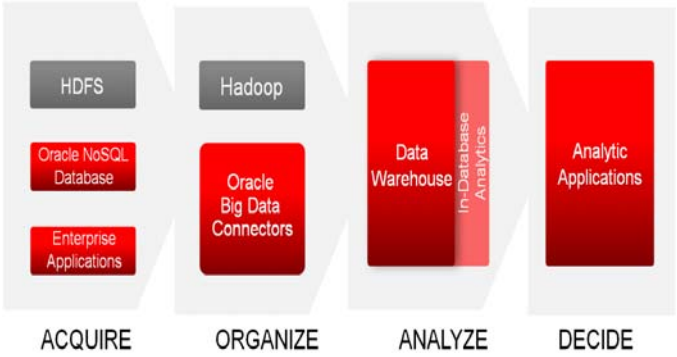

Figure 2. Oracle's Big Data Solution.

Oracle is uniquely qualified to combine everything needed to meet the big data challenge – including software and hardware – into one engineered system. The Oracle Big Data Appliance is an engineered system that combines optimized hardware with the most comprehensive software stack featuring specialized solutions developed by Oracle to deliver a complete, easy-to-deploy solution for acquiring, organizing and loading big data into Oracle Database 11g. It is designed to deliver extreme analytics on all data types, with enterprise-class performance, availability, supportability and security. With Big Data Connectors, the solution is tightly integrated with Oracle Exadata and Oracle Database, so you can analyze all your data together with extreme performance.

IX. ORACLE BIG DATA APLLIANCE

Oracle Big Data Appliance comes in a full rack configuration with 18 Sun servers for a total storage capacity of 648TB. Every server in the rack has 2 CPUs, each with 6 cores for a total of 216 cores per full rack.

Oracle Big Data Appliance includes a combination of open source software and specialized software developed by Oracle to address enterprise big data requirements.

The Oracle Big Data Appliance integrated software includes:

a. Full distribution of Cloudera's Distribution including Apache Hadoop (CDH)
b. Cloudera Manager to administer all aspects of Cloudera CDH
c. Open source distribution of the statistical package R for analysis of unfiltered data on Oracle Big Data Appliance
d. Oracle NoSQL Database Community Edition 3
e. And Oracle Enterprise Linux operating system and Oracle Java VM.

A. CDH and Cloudera Manager:

Oracle Big Data Appliance contains Cloudera's Distribution including Apache Hadoop (CDH) and Cloudera Manager. CDH is the #1 Apache Hadoop-based distribution in commercial and non-commercial environments. CDH consists of 100% open source Apache Hadoop plus the comprehensive set of open source software components needed to use Hadoop. Cloudera Manager is an end-to-end management application for CDH. Cloudera Manager gives a cluster-wide, real-time view of nodes and services running; provides a single, central place to enact configuration changes across the cluster; and incorporates a full range of reporting and diagnostic tools to help optimize cluster performance and utilization.

B. Oracle Big Data Connector:

Where Oracle Big Data Appliance makes it easy for organizations to acquire and organize new types of data, Oracle Big Data Connectors enables an integrated data set for analyzing all data. Oracle Big Data Connectors can be installed on Oracle Big Data Appliance or on a generic Hadoop cluster

C. Oracle Loader for Hadoop:

Oracle Loader for Hadoop (OLH) enables users to use Hadoop MapReduce processing to create optimized data sets for efficient loading and analysis in Oracle Database 11g. Unlike other Hadoop loaders, it generates Oracle internal formats to load data faster and use less database system resources. OLH is added as the last step in the MapReduce transformations as a separate map – partition – reduce step. This last step uses the CPUs in the Hadoop cluster to format the data into Oracle-understood formats, allowing for a lower CPU load on the Oracle cluster and higher data ingest rates because the data is already formatted for Oracle Database. Once loaded, the data is permanently available in the database providing very fast access to this data for general database users leveraging SQL or Business Intelligence tools.

D. Oracle Direct Connector for Hadoop Distributed File System:

Oracle Direct Connector for Hadoop Distributed File System (HDFS) is a high speed connector for accessing data on HDFS directly from Oracle Database. Oracle Direct Connector for HDFS gives users the flexibility of querying data from HDFS at any time, as needed by their application.

It allows the creation of an external table in Oracle Database, enabling direct SQL access on data stored in HDFS. The data stored in HDFS can then be queried via SQL, joined with data stored in Oracle Database, or loaded into the Oracle Database. Access to the data on HDFS is optimized for fast data movement and parallelized, with automatic load balancing. Data on HDFS can be in delimited files or in Oracle data pump files created by Oracle Loader for Hadoop.

E. Oracle Data Integrator Application Adapter for Hadoop:

Oracle Data Integrator Application Adapter for Hadoop simplifies data integration from Hadoop and an Oracle Database through Oracle Data Integrator's easy to use interface. Once the data is accessible in the database, end users can use SQL and Oracle BI Enterprise Edition to access data. Enterprises that are already using a Hadoop solution, and don't need an integrated offering like Oracle Big Data Appliance, can integrate data from HDFS using Big Data Connectors as a stand-alone software solution.

F. Oracle R for Hadoop:

Oracle R Connector for Hadoop is an R package that provides transparent access to Hadoop and to data stored in HDFS.

R Connector for Hadoop provides users of the open-source statistical environment R with the ability to analyze data stored in HDFS, and to run R models at scale against large volumes of data leveraging MapReduce processing – without requiring R users to learn yet another API or language. End users can leverage over 3500 open source R packages to analyze data stored in HDFS, while administrators do not need to learn R to schedule R MapReduce models in production environments.

R Connector for Hadoop can optionally be used together with the Oracle Advanced Analytics Option for Oracle Database. The Oracle Advanced Analytics Option enables R users to transparently work with database resident data without having to learn SQL or database concepts but with R computations executing directly in-database.

G. Oracle NoSQL Database:

Oracle NoSQL Database is a distributed, highly scalable, key-value database based on Oracle Berkeley DB. It delivers a general purpose, enterprise class key value store adding an intelligent driver on top of distributed Berkeley DB. This intelligent driver keeps track of the underlying storage topology, shards the data and knows where data can be placed with the lowest latency. Unlike competitive solutions, Oracle NoSQL Database is easy to install, configure and manage, supports a broad set of workloads, and delivers enterprise-class reliability backed by enterprise-class Oracle support.

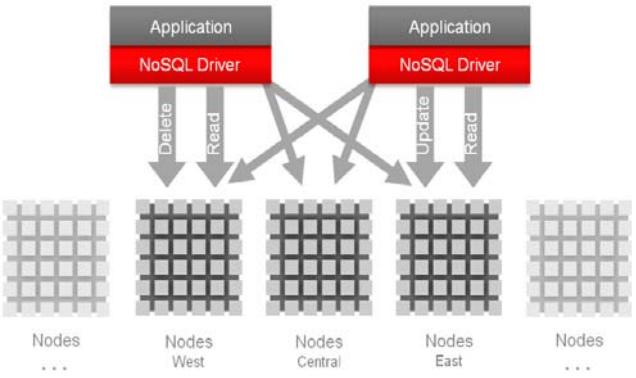

Figure .3 NoSQL Database Architecture

X. IN-DATABASE ANALYTICS

Once data has been loaded from Oracle Big Data Appliance into Oracle Database or Oracle Exadata, end users can use one of the following easy-to-use tools for in-database, advanced analytics:

a. Oracle R Enterprise – Oracle's version of the widely used Project R statistical environment enables statisticians to use R on very large data sets without any modifications to the end user experience. Examples of R usage include predicting airline delays at a particular airports and the submission of clinical trial analysis and results.
b. In-Database Data Mining – the ability to create complex models and deploy these on very large data volumes to drive predictive analytics. End-users can leverage the results of these predictive models in their BI tools without the need to know how to build the models. For example, regression models can be used to predict customer age based on purchasing behavior and demographic data.
c. In-Database Text Mining – the ability to mine text from micro blogs, CRM system comment fields and review sites combining Oracle Text and Oracle Data Mining. An example of text mining is sentiment analysis based on comments. Sentiment analysis tries to show how customers feel about certain companies, products or activities.
d. In-Database Semantic Analysis – the ability to create graphs and connections between various data points and data sets. Semantic analysis creates, for example, networks of relationships determining the value of a customer's circle of friends. When looking at customer churn customer value is based on the value of his network, rather than on just the value of the customer.
e. In-Database Spatial – the ability to add a spatial dimension to data and show data plotted on a map. This ability enables end users to understand geospatial relationships and trends much more efficiently. For example, spatial data can visualize a network of people and their geographical proximity. Customers who are in close proximity can readily influence each other's purchasing behavior, an opportunity which can be easily missed if spatial visualization is left out.
f. In-Database MapReduce – the ability to write procedural logic and seamlessly leverage Oracle Database parallel execution. In-database MapReduce allows data scientists to create high-performance routines with complex logic. In-database MapReduce can be

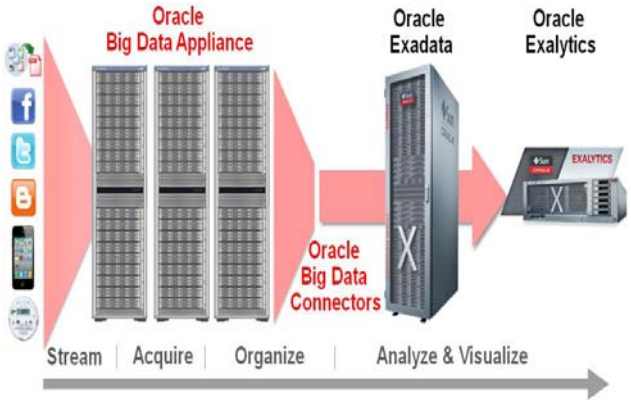

Figure. 4 Usage model for Big Data Appliance and Exadata

XI. APACHE HADOOP

Following are several definitions of Hadoop, each one targeting a different audience within the enterprise:

a. For the executives: Hadoop is an Apache open source software project to get value from the incredible volume/velocity/variety of data about your organization. Use the data instead of throwing most of it away.
b. For the technical managers: An open source suite of software that mines the structured and unstructured Big Data about your company. It integrates with your existing Business Intelligence ecosystem.
c. Legal: An open source suite of software that is packaged and supported by multiple suppliers. Please see the Resources section regarding IP indemnification.
d. Engineering: A massively parallel, shared nothing, Java-based map-reduce execution environment. Think hundreds to thousands of computers working on the same problem, with built-in failure resilience. Projects in the Hadoop ecosystem provide data loading, higher-level languages, automated cloud deployment, and other capabilities.
e. Security: A Kerberos-secured software suite.

XII. COMPONENTS OF HADOOP

The Apache Hadoop project has two core components, the file store called Hadoop Distributed File System (HDFS), and the programming framework called MapReduce. There are a number of supporting projects that leverage HDFS and MapReduce. This article will provide a summary, and encourages you to get the oReily book "Hadoop the Definitive Guide", 3rd Edition, for more details.

The definitions below are meant to provide just enough background for you to use the code examples that follow. This article is really meant to get you started with hands-on experience with the technology. This is a how-to article more than a what-is orlets-discuss article.

a. HDFS: HDFS is a distributed, scalable, and portable file system written in Java for the Hadoop framework. If you want 4000+ computers to work

on your data, then you'd better spread your data across 4000+ computers. HDFS does this for you. HDFS has a few moving parts. The Data nodes store your data, and the Name node keeps track of where stuff is stored. There are other pieces, but you have enough to get started.

b. MapReduce: MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogeneous hardware). Computational processing can occur on data stored either in a file system (unstructured) or in a database (structured). MapReduce can take advantage of locality of data, processing data on or near the storage assets to decrease transmission of data. This is the programming model for Hadoop. There are two phases, not surprisingly called Map and Reduce. To impress your friends tell them there is a shuffle-sort between the Map and Reduce phase. The Job Tracker manages the 4000+ components of your MapReduce job. The Task Trackers take orders from the Job Tracker. If you like Java then code in Java. If you like SQL or other non-Java languages you are still in luck, you can use a utility called Hadoop Streaming.

c. Hadoop Streaming: A utility to enable MapReduce code in any language: C, Perl, Python, C++, Bash, etc. The examples include a Python mapper and an AWK reducer.

d. Hive and Hue: Hue is mature and used from fortune 500 companies to startups, students and data scientist/analysts. With Hue, one can get started with Hadoop and become familiar with and explore different angles of the platform. Hue's target is the Hadoop user experience and lets users avoid the command line interface and have them focus on visibility and getting results quickly. Hue is composed of various applications. The major query and scripting solutions are leveraged in specific editors (e.g. Hive, Impala) for quick interaction with big data. A Hadoop Job Browser and File Browser are included for monitoring MapReduce tasks and intuitively manipulating the data. If you like SQL, you will be delighted to hear that you can write SQL and have Hive convert it to a MapReduce job. No, you don't get a full ANSI-SQL environment, but you do get 4000 notes and multi-Petabyte scalability. Hue gives you a browser-based graphical interface to do your Hive work

e. Pig: The Hadoop execution environment supports additional distributed data processing capabilities which are designed to run using the Hadoop MapReduce architecture. These include Pig – a high-level data-flow programming language and execution framework for data-intensive computing. Pig was developed at Yahoo! to provide a specific data-centric language notation for data analysis applications and to improve programmer productivity and reduce development cycles when using the Hadoop MapReduce environment. Pig programs are automatically translated into sequences of MapReduce programs if needed in the execution environment. Pig provides capabilities in the language for loading, storing, filtering, grouping, de-duplication, ordering, sorting, aggregation, and joining operations on the data. A higher-level programming environment to do MapReduce coding. The Pig language is called Pig Latin. You may find the naming conventions somewhat unconventional, but you get incredible price-performance and high availability.

f. Sqoop: Sqoop is a command-line interface application for transferring data between relational databases and Hadoop. It supports incremental loads of a single table or a free form SQL query as well as saved jobs which can be run multiple times to import updates made to a database since the last import. Import can also be made to populate tables in Hive or HBase. Exports can be used to put data from Hadoop into a relation database. Sqoop became a top level Apache project in March 2012. Provides bi-directional data transfer between Hadoop and your favorite relational database.

g. Oozie: Oozie is a workflow scheduler system to manage Hadoop jobs. It is a server based Workflow Engine specialized in running workflow jobs with actions that run Hadoop Map/Reduce and Pig jobs. Oozie is a Java Web-Application that runs in a Java servlet-container. For the purposes of Oozie, a workflow is a collection of actions (i.e. Hadoop Map/Reduce jobs, Pig jobs) arranged in a control dependency DAG (Direct Acyclic Graph). "Control dependency" from one action to another means that the second action can't run until the first action has completed. The workflow actions start jobs in remote systems (i.e. Hadoop, Pig). Upon action completion, the remote systems callback Oozie to notify the action completion, at this point Oozie proceeds to the next action in the workflow. Oozie workflows contain control flow nodes and action nodes. Control flow nodes define the beginning and the end of a workflow (start , end and fail nodes) and provide a mechanism to control the workflow execution path (decision , fork and join nodes).Action nodes are the mechanism by which a workflow triggers the execution of a computation/processing task. Oozie provides support for different types of actions: Hadoop map-reduce, Hadoop file system, Pig, SSH, HTTP, e-Mail and Oozie sub-workflow. Oozie can be extended to support additional type of actions .Manages Hadoop workflow. This doesn't replace your scheduler or BPM tooling, but it does provide

if-then-else branching and control within your Hadoop jobs.

h. HBase: A super-scalable key-value store. It works very much like a persistent hash-map (for python fans think dictionary). It is not a relational database despite the name HBase.

i. FlumeNG: A real time loader for streaming your data into Hadoop. It stores detain HDFS and HBase. You'll want to get started with FlumeNG, which improves on the original flume.

j. Whirr: Cloud provisioning for Hadoop. You can start up a cluster in just a few minutes with a very short configuration file.

k. Mahout: Machine learning for Hadoop. Used for predictive analytics and other advanced analysis.

l. Fuse: Makes the HDFS system look like a regular file system so you can use ls, rm, cd and others on HDFS data

m. Zookeeper: Used to manage synchronization for the cluster. You won't be working much with Zookeeper, but it is working hard for you. If you think you need to write a program that uses Zookeeper you are either very, very, smart and could be a committee for an Apache project, or you are about to have a very bad day.

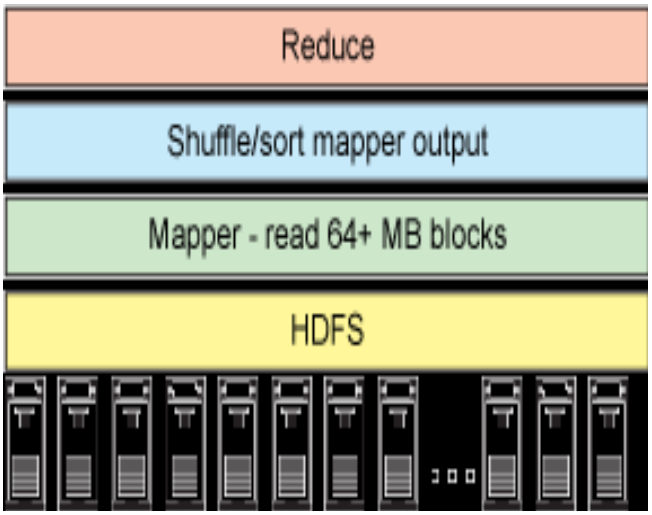

Figure.5 shows the key pieces of Hadoop.

XIII. ACKNOWLEDGMENT

I would like to express my greatest gratitude to TCS professionals who told us about the latest technologies currently using in the industries.

A special thank of mine goes to my colleague who helped me in completing the paper and she exchanged her interesting ideas, thoughts, and made this paper easy and accurate.

XIV.REFERENCES

[1] An Oracle white paper,- big data for enterprise, pp. 1-16, January 2012.

[2] Marty Lurie,- Open source big data for the impatient, part 1 || Hadoop tutorial:Hello world with java, pig, Hive, Flume, Fuse, Oozie, and Sqoop with Informix, DB2 and mySQl, 27 sept 2012, pp 3-5.

[3] ORACLE white paper-INTEGREAT FOR INSIGHT,pp 1.

[4] http://en.wikipedia.org/wiki/HDFS#Hadoop_Distributed_File_System

[5] http://en.wikipedia.org/wiki/Big_data

[6] http://en.wikipedia.org/wiki/Mapreduce

[7] http://cloudera.github.com/hue/

[8] http://en.wikipedia.org/wiki/Sqoop

[9] http://en.wikipedia.org/wiki/Data-centric_programming_language#Hadoop_Pig

[10] http://en.wikipedia.org/wiki/Oozie