A Comparison of Big Data Analytics Approaches Based on Hadoop MapReduce

Kyar Nyo Aye, Ni Lar Thein
University of Computer Studies, Yangon, Myanmar
kyarnyoaye@gmail.com, nilarthein@gmail.com

Abstract

Data continue a massive expansion in scale, diversity, and complexity. Data underpin activities in all sectors of society. Achieving the full transformative potential from the use of data in this increasingly digital world requires not only new data analysis algorithms but also a new generation of systems and distributed computing environments to handle the dramatic growth in the volume of data, the lack of structure for much of it and the increasing computational needs of massive-scale analytics. In this paper, we propose big data platform that is built upon open source and built on Hadoop MapReduce, Gluster File System, Apache Pig, Apache Hive and Jaql and compare our platform with other two big data platforms – IBM big data platform and Splunk. Our big data platform can support large scale data analysis efficiently and effectively.

Keywords: big data, big data analytics, big data platform, Hadoop MapReduce, Gluster file system, Apache Pig, Apache Hive, Jaql

1. Introduction

Every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. This data is big data.

Such huge amount of data needs to be stored for various reasons. Sometimes any compliance demands more historical data to be stored. Sometimes organizations want to store, process and analyze this data for intelligent decision making to get the competitive advantage. For example analyzing CDR data can help a service provider know their quality of service and then make the necessary improvements. A credit card company can analyze the customer transactions for fraud detection. Server logs can be analyzed for fault detection. Web logs can help understand the user navigation patterns. Customer emails can help understand the customer behavior, interests and some time the problems with the products as well.

Now the important question that arises at this point of time is how do we store and process such huge amount of data most of which is Semi structured or Unstructured. There is a high-level categorization of big data platforms to store and process them in a scalable, fault tolerant and efficient manner [10]. The first category includes massively parallel processing or MPP Data warehouses that are designed to store huge amount of structured data across a cluster of servers and perform parallel computations over it. Most of these solutions follow shared nothing architecture which means that every node will have a dedicated disk, memory and processor. All the nodes are connected via high speed networks. As they are designed to hold structured data so there is a need to extract the structure from the data using an ETL tool and populate these data sources with the structured data.

These MPP Data Warehouses include:

1) MPP Databases: these are generally the distributed systems designed to run on a cluster of commodity servers. E.g. Greenplum,, IBM DB2, Teradata.

2) Appliances: a purpose-built machine with preconfigured MPP hardware and software designed for analytical processing. E.g. Oracle Optimized Warehouse, Netezza Performance Server and so on.

3) Columnar Databases: they store data in columns instead of rows, allowing greater compression and faster query performance. E.g. Sybase IQ, Vertica, InfoBright Data Warehouse, ParAccel.

Another category includes distributed file systems like Hadoop to store huge unstructured data and perform Map Reduce computations on it over a cluster built of commodity hardware. Hadoop is a popular open-source map-reduce implementation which is being used in companies like Yahoo, Facebook etc. to store and process extremely large data sets on commodity hardware. However, using Hadoop was not easy for end users, especially for those users who were not familiar with MapReduce. The MapReduce programming model is very low level and requires developers to write custom programs which are hard to maintain and reuse. End users had to write map-reduce programs for simple tasks like getting raw counts or averages. Hadoop lacked the expressiveness of popular query languages like SQL and as a result users ended up spending hours to write programs for even simple analysis. In order to analyze this data more productively, the query capabilities of Hadoop need to be improved. So, several application development languages have emerged to make it easier to write MapReduce programs in Hadoop and that run on top of Hadoop. Among them, Hive, Pig, and Jaql are popular.

The purpose of this paper is to propose big data platform that is built upon open source and built on Hadoop MapReduce, Gluster File System, Apache Pig, Apache Hive and Jaql and compare proposed platform with other two big data platforms – IBM big data

platform and Splunk. The rest of the paper is organized as follows: In section 2, we present related work and explain background theory such as Big Data and Big Data Analytics in section 3. In section 4, we introduce our proposed big data platform, IBM big data platform, Splunk and compare them. Then conclusion is described in section 5.

2. Related Work

We survey some of existing big data platforms for large scale data analysis. There are many types of vendor products to consider for big data analytics. More recently, vendors have brought out analytic platforms based on MapReduce, distributed file system, and no-SQL indexing. ParAccel Analytic Database (PADB), the world's fastest, most cost-effective platform for empowering analytics-driven businesses. When combined with the WebFOCUS BI platform, ParAccel enables organizations to tackle the most complex analytic challenges and glean ultra-fast, deep insights from vast volumes of data.

The SAND Analytic Platform is a columnar analytic database platform that achieves linear data scalability through massively parallel processing (MPP), breaking the constraints of shared-nothing architectures with fully distributed processing and dynamic allocation of resources [8]. The Vertica Analytics Platform offers a robust and ever growing set of Advanced In-Database Analytics functionality. It has a high-speed, relational SQL database management system (DBMS) purpose-built for analytics and business intelligence. It offers a shared-nothing, Massive Parallel Processing (MPP) column-oriented architecture [4]. 1010data offers a data and analytics platform that is the only complete approach to performing the deepest analysis and getting the maximum insight directly from raw data, at a fraction of the cost and time of any other solution [6].

Netezza, a leading developer of combined server, storage, and database appliances designed to support the analysis of terabytes of data and provide companies with a powerful analytics foundation that delivers maximum speed, reliability, and scalability [5]. Pavlo et al. [7] described and compared MapReduce paradigm and parallel DBMSs for large scale data analysis and defined a benchmark consisting of a collection of tasks to be run on an open source version of MR as well as on two parallel DBMSs.

3. Background Theory

This section provides an overview of big data, big data Analytics, Hadoop and MapReduce framework, Apache Pig, Apache Hive, Jaql and Gluster File System. Due to space constraints, some aspects are explained in a highly simplified manner. A detailed description of them can be found in [1] [2] [8].

3.1 Big Data

Big Data are data sets that grow so large that they become awkward to work with using on-hand database management tools. Difficulties include capture, storage, search, sharing, analytics, and visualizing. There are three characteristics of Big Data: volume, variety, and velocity.

1) Volume: Volume is the first and most notorious feature. It refers to the amount of data to be handled. The sheer volume of data being stored today is exploding. In the year 2000, 800,000 petabytes (PB) of data were stored in the world. This number is expected to reach 35 zettabytes (ZB) by 2020. Twitter alone generates more than 7 terabytes (TB) of data every day. Facebook 10 TB, and some enterprises generate terabytes of data every hour of every day of the year. As implied by the term "Big Data", organizations are facing massive volumes of data. Organizations that don't know how to manage this data are overwhelmed by it. But the opportunity exists, with the right technology platform, to analyze almost all of the data to gain better insights.

2) Variety: Variety represents all types of data. With the explosion of sensors, and smart devices, as well as social collaboration technologies, data in an enterprise has become complex, because it includes not only traditional relational data, but also raw, semistructured, and unstructured data from web pages, web log files (including click-stream data), search indexes, social media forums, e-mail, documents, sensor data, and so on. Traditional analytic platforms can't handle variety. However, an organization's success will rely on its ability to draw insights from the various kinds of data available to it, which includes both traditional and nontraditional. So to capitalize on the Big Data opportunity, enterprises must be able to analyze all types of data, both relational and nonrelational: text, sensor data, audio, video, transactional, and more.

3) Velocity: A conventional understanding of velocity typically considers how quickly the data is arriving and stored, and its associated rates of retrieval. However, today the term velocity is defined to data in motion: the speed at which the data is flowing. Today's enterprises are dealing with petabytes of data instead of terabytes, and the increase in RFID sensors and other information streams has led to a constant flow of data at a pace that has made it impossible for traditional systems to handle. In addition, more and more of the data being produced today has a very short shelf-life, so organizations must be able to analyze this data in near real time if they hope to find insights in this data. So dealing effectively with Big Data requires performing analytics against the volume and variety of data while it is still in motion, not just after it is at rest.

3.2 Big Data Analytics

Big data analytics is the application of advanced analytic techniques to very big data sets. Advanced analytics is a collection of techniques and tool types, including predictive analytics, data mining, statistical analysis, complex SQL, data visualization, artificial intelligence, natural language processing, and database methods that support analytics (such as MapReduce, in-

database analytics, in-memory database, columnar data stores). Big Data analytics requires massive performance and scalability- common problems that old platforms can't scale to big data volumes, load data too slowly, respond to queries too slowly, lack processing capacity for analytics and can't handle concurrent mixed workloads. There are two main techniques for analyzing big data: the store and analyze approach, and the analyze and store approach [9].

3.3 Hadoop and MapReduce Framework

Apache Hadoop is an open source software project that enables the distributed processing of large data sets across clusters of commodity servers. It is designed to scale up from a single server to thousands of machines, with a very high degree of fault tolerance. Rather than relying on high-end hardware, the resiliency of these clusters comes from the software's ability to detect and handle failures at the application layer.

Hadoop enables a computing solution that is:

1) Scalable: New nodes can be added as needed and added without needing to change data formats, how data is loaded, how jobs are written, or the applications on top.

2) Cost effective: Hadoop brings massively parallel computing to commodity servers. The result is a sizeable decrease in the cost per terabyte of storage, which in turn makes it affordable to model all data.

3) Flexible: Hadoop is schema-less, and can absorb any type of data, structured or not, from any number of sources. Data from multiple sources can be joined and aggregated in arbitrary ways enabling deeper analyses than any one system can provide.

4) Fault tolerant: When a node fails, the system redirects work to another location of the data and continues processing.

A MapReduce framework typically divides the input data-set into independent tasks 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 jobs are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and reexecuting the failed tasks [12]. Hadoop is supplemented by an ecosystem of Apache projects, such as Pig and Hive, that extend the value of Hadoop and improves its usability.

3.4 Apache Pig

Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. Pig is made up of two components: the first is the language itself, which is called PigLatin, and the second is a runtime environment where PigLatin programs are executed. The following is an example of a Pig program that takes a file composed of Twitter feeds, selects only those tweets that are using the en(English) iso_language code, then group them by the user who is tweeting, and displays the sum of the number of retweets of that user's tweets.

```
L = LOAD 'hdfs://node/tweet_data';
FL = FILTER L BY iso_language_code EQ 'en';
G = GROUP FL BY from_user;
RT = FOREACH G GENERATE group, SUM (retweets);
DUMP RT;
```

There are three ways to run a Pig program: embedded in a script, embedded in a Java program, or from the Pig command line, called Grunt. The Pig runtime environment translates the program into a set of map and reduce tasks and runs them. This greatly simplifies the work associated with the analysis of large amounts of data and lets the developer focus on the analysis of the data rather than on the individual map and reduce tasks.

3.5 Apache Hive

Apache Hive is an open-source data warehousing solution built on top of Hadoop. Hive supports queries expressed in a SQL-like declarative language - *HiveQL*, which are compiled into mapreduce jobs that are executed using Hadoop. In addition, HiveQL enables users to plug in custom map-reduce scripts into queries. The language includes a type system with support for tables containing primitive types, collections like arrays and maps, and nested compositions of the same. The underlying IO libraries can be extended to query data in custom formats. Hive also includes a system catalog - *Metastore* – that contains schemas and statistics, which are useful in data exploration, query optimization and query compilation. The following shows an example of creating a table, populating it, and then querying that table using Hive:

```
CREATE TABLE Tweets (from_user STRING, userid BIGINT, tweettext STRING, retweets INT)
COMMENT 'This is the Twitter feed table'
STORED AS SEQUENCEFILE;
LOAD DATA INPATH 'hdfs://node/tweetdata' INTO TABLE TWEETS;
SELECT from_user, SUM (retweets)
FROM TWEETS
GROUP BY from_user;
```

3.6 Jaql

Jaql is a functional, declarative query language that is designed to process large data sets. For parallelism, Jaql rewrites high-level queries into low-level queries consisting of MapReduce jobs. Jaql is primarily a query language for JavaScript Object Notation (JSON). JSON is the popular data interchange format because it is easy for humans to read, and because of its structure, it is easy for applications to parse or generate. The following shows an example of a JSON representation of a Twitter feed:

```
results: [
{
```

created_at: “Thurs, 14 Jul 2011 09:47:45 +0000”
from_user: “david”
geo: { coordinates: [
43.866667
78.933333
]
type: “Point” }
iso_language_code:”en”
text: “Reliance Life Insurance migrates from #Oracle to #DB2 and cuts costs in half. Read what they say about their migration http://bit.ly/pP7vaT”
retweet: 3
to_user_id: null
to_user_id_str:;null
}

Both Jaql and JSON are record-oriented models, and thus fit together perfectly. JSON is not the only format that Jaql supports, Jaql is extremely flexible and can support many semistructured data sources such as XML, CSV, flat files and more. The following shows a simple Jaql example that counts the number of tweets written in English by user:

```
$tweets = read(hdfs(“tweet_log”));
$tweets
➔ filter $.iso_language_code = “en”
➔ group by u = $.from_user
into {user: $.from_user, total: sum ($.retweet) };
```

3.6 Gluster File System

Gluster file system is a scalable open source clustered file system that offers a global namespace, distributed front end, and scales to hundreds of petabytes without difficulty. It is also a software-only, highly available, scalable, centrally managed storage pool for unstructured data. It is also scale-out file storage software for NAS, object, big data. By leveraging commodity hardware, Gluster also offers extraordinary cost advantages benefits that are unmatched in the industry. There are many advantages of Gluster over any other file systems. These advantages are:

1) It is faster for each individual operation because it calculates metadata using algorithms and that approach is faster than retrieving metadata from any storage media.
2) It is faster for large and growing individual systems because there is never any contention for any single instance of metadata stored at only one location.
3) It is faster and achieves true linear scaling for distributed deployments because each node is independent in its algorithmic handling of its own metadata, eliminating the need to synchronize metadata.
4) It is safer in distributed deployments because it eliminates all scenarios of risk which are derived from out-of-synch metadata.

GlusterFS 3.3 beta 2 includes compatibility for Apache Hadoop and it uses the standard file system APIs available in Hadoop to provide a new storage option for Hadoop deployments [3].

4. Big Data Platform

Big data platform cannot just be a platform for processing data; it has to be a platform for analyzing that data to extract insight from an immense volume, variety, and velocity of that data. The main components in the big data platform provide:

1) Deep analytics: a fully parallel, extensive and extensible toolbox full of advanced and novel statistical and data mining capabilities
2) High agility: the ability to create temporary analytics environments in an end-user driven, yet secure and scalable environment to deliver new and novel insights to the operational business
3) Massive scalability: the ability to scale analytics and sandboxes to previously unknown scales while leveraging previously untapped data potential
4) Low latency: the ability to instantly act based on these advanced analytics in the operational, production environments [11].

4.1 IBM Big Data Platform

IBM offers a platform for big data including IBM InfoSphere Biginsights and IBM InfoSphere Streams. IBM InfoSphere Biginsights represents a fast, robust, and easy-to-use platform for analytics on Big Data at rest. IBM InfoSphere Streams is a powerful analytic computing platform that delivers a platform for analyzing data in real time with micro-latency [2]. To the best of our knowledge, there is no other vendor that can deliver analytics for Big Data in motion (InfoSphere Streams) and Big Data at rest (BigInsights) together. Figure 1 describes IBM big data platform.

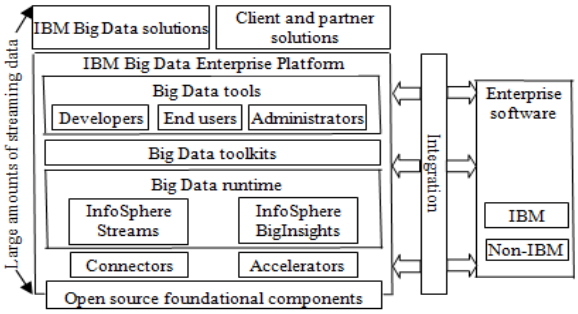

Figure 1. IBM big data platform

4.2 Splunk

Splunk is a general-purpose search, analysis and reporting engine for time-series text data, typically machine data. It provides an approach to machine data processing on a large scale, based on the MapReduce model. Machine data is a valuable resource. It contains a definitive record of all user transactions, customer behavior, machine behavior, security threats, fraudulent activity and more. It’s also dynamic, unstructured, non-standard and makes up the majority of the data in the organization. The Splunk search language is simple enough to allow exploration of data by almost anyone with a little training, enabling many people to explore data. It is powerful enough to support a complicated

data processing pipeline. Figure 2 describes the Splunk architecture [14].

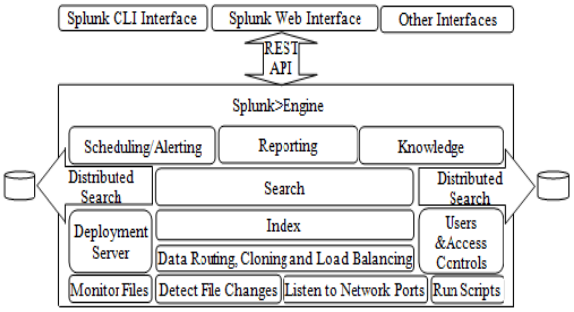

Figure 2. Splunk architecture

4.3 Proposed Big Data Platform

In general for big data analytics, there are three approaches: 1) direct analytics over massively parallel processing data warehouses, 2) indirect analytics over hadoop and 3) direct analytics over hadoop. The proposed approach performs analytics over Hadoop MapReduce framework and Gluster file system. All the queries for analytics are executed as Map Reduce jobs over big unstructured data placed into Gluster file system. By using this approach, a highly scalable, fault tolerant and low cost big data solution can be achieved.

By combining scalability to petabytes and beyond, affordability (use of commodity hardware), flexibility (deploy in any environment), linearly scalable performance, high availability, unified files and objects, file system for apache hadoop and superior storage economics of Gluster file system with parallel data processing, schema free processing and simplicity of Map Reduce programming model, its open source implementation Hadoop and other Hadoop open source projects (pig, hive, jaql), the proposed big data platform can perform large scale data analysis efficiently and effectively. Proposed big data platform is shown in Figure 3.

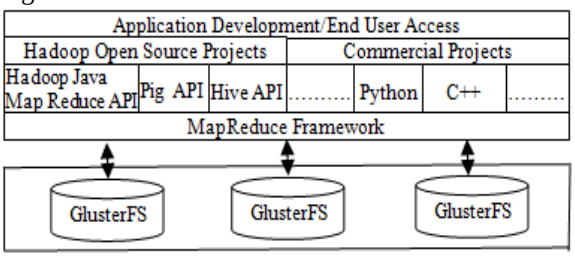

Figure 3. Proposed big data platform

4.4 A Comparison of Three Big Data Platforms

Volume, Variety, and Velocity

IBM offers the broadest platform for big data, addressing all three dimensions of the big data challenge – volume, variety and velocity. It can support both batch processing (IBM InfoSphere BigInsights) and real-time processing (IBM InfoSphere Streams). Splunk supports both batch (Splunk Hadoop Connect) and real time processing. Our platform addresses volume and variety features of Big Data and only supports batch processing.

Big Data Storage

For Big Data storage, IBM uses GPFS-SNC (General Parallel File System – Shared Nothing Cluster) and it is a distributed storage cluster with a shared-nothing architecture. A GPFS-SNC cluster consists of multiple racks of commodity hardware, where the storage is attached to the compute nodes. Splunk uses HDFS via Splunk Hadoop Connect and our platform uses Gluster File System. HDFS is not fully POSIX-compliant but GPFS-SNC and Gluster File system are 100 percent POSIX-compliant.

Processing Model

The three big data platforms apply MapReduce processing model for large-scale data analytics. IBM Big Data platform uses Adaptive MapReduce, which extends Hadoop by making individual mappers self aware and aware of other mappers. This approach enables individual map tasks to adapt to their environment and make efficient decisions. Splunk uses a Temporal MapReduce model even when running on a single node and Spatial MapReduce to speed computation by distributing processing throughout a cluster of many machines.

Cloud Support

Though Hadoop was not designed for virtualized environments such as the ones provided with the cloud, the cloud still provides an environment that is easy to set up and cost effective. IBM supports cloud by setting up a Hadoop cluster on the IBM Cloud. Splunk Storm takes the machine data from cloud services and applications and harnesses that data to generate critical operational insights and application intelligence. It is a cloud service, based on powerful Splunk software, used by thousands of enterprise customers worldwide. However, our Big Data platform cannot support cloud.

Query Support

The Splunk Search Language is a concise way to express the processing of sparse or dense tabular data, using the MapReduce mechanism, without having to write code or understand how to split processing between the map and reduce phases. In our platform and IBM Big Data platform, Pig, Hive and Jaql facilitate analysis of both structured and non-traditional data types.

Scalability

One of the key components of Hadoop is the redundancy built into the environment. This redundancy allows Hadoop to scale out workloads across large clusters of inexpensive machines to work on Big Data problems. So IBM Big Data Platform supports massive scale-out and achieves high scalability because of Hadoop. The Splunk architecture is based on MapReduce and scales linearly across commodity servers to unlimited data volumes. Our platform is based on Hadoop and Gluster File System, so it can provide linear scalability.

Fault Tolerance

In Hadoop, not only is the data redundantly stored in multiple places across the cluster, but the programming model is such that failures are expected and are resolved automatically by running portions of the program on various servers in the cluster. Due to this redundancy, it is possible to distribute the data and its associated programming across a very large cluster of commodity components. It is well known that commodity hardware components will fail, but this redundancy provides fault tolerance and a capability for the Hadoop cluster to heal itself. So three Big Data platforms can provide fault tolerance because of Hadoop.

Visualization

BigInsights includes a browser-based visualization tool called BigSheets, which enables users to harness the power of Hadoop using a familiar spreadsheet interface. BigSheets requires no programming or special administration. BigSheets provides many visualization tools: Tag Cloud, Pie Chart, Map, Heat Map and Bar Chart. In Splunk, report builder can be used to generate analytical reports with advanced charts, graphs, tables and dashboards that show important trends, highs and lows, summaries of top values and frequency of occurrences. In our platform, currently we don't consider visualization aspects.

Enterprise Integration

IBM provides extensive integration capabilities, integrates wide variety of sources and leverages enterprise integration technologies. BigInsights supports data exchange with a number of sources, including Netezza: DB2 for Linux, UNIX, and Windows; other relational data stores via a JDBC interface; InfoSphere Streams; InfoSphere Information Sever; R Statistical Analysis Applications; and more. Splunk delivers a powerful platform for enterprise applications and Splunk Hadoop Connect integration enables users to leverage Splunk Enterprise to reliably collect massive volumes of machine data. Our platform does not consider for enterprise integration.

Ease of Use

IBM BigInsights includes a web-based administration console that provides a real-time, interactive view of cluster. The BigInsights console provides a graphical tool for examining the health of BigInsights environment, including the nodes in cluster, the status of jobs (applications), and the contents of GPFS-SNC file system. Through this console, we can add and remove nodes, start and stop nodes, inspect the status of MapReduce jobs, assess the overall health of platform, start and stop optional components, navigate files in the BigInsights cluster, and more. Splunk is enterprise software made easy. Splunk's interface runs as a Web server and dashboards can be created and maintained with the Splunk Web UI and so it is easy to use for any user. Our platform requires downloading, configuring, and testing the individual open source projects such as Hadoop, Gluster File System, Pig, Hive and Jaql.

4.5 Performance Evaluation

We have evaluated the performance of proposed system on a commodity Linux cluster with four virtual machines. The VMs are interconnected via a 1-gigabit Ethernet. The host machine runs Windows 7 Ultimate and has Intel Core i7-3.40GHz processor, 4GB physical memory, and 950-GB disk. In the Hadoop cluster implemented, one VM stands for JobTracker. The other three VMs stand for TaskTrackers. The virtual machine for JobTracker runs Ubuntu 11.10 with a kernel version of 3.0.0-12-generic and has one processor, 1GB physical memory and 50-GB disk. The other VMs run Ubuntu 11.10 with a kernel version of 3.0.0-12-generic and have one processor, 512MB physical memory and 50-GB disk each. Hadoop 0.20.2, Gluster 3.3.1, Hive 0.9.0, Pig 0.10.0 and Jaql 0.5.1 are installed and book publication data set [13] is used to test for query performance. The data set is about book and includes information such as isbn, booktitle, bookauthor, yearofpublication, publisher and so on. The query is to find the frequency of books published each year. The query is also tested on IBM InfoSphere BigInsights 1.3 and Splunk 5.0.1. The query format is different in these systems.

In the proposed system, the HiveQl (Hive Query Language) is

hive> create table if not exists books (ISBN string, BookTitle string, BookAuthor string, YearOfPublication string, Publisher string) row format delimited fields terminated by '\,' stored as textfile;
hive>load data inpath 'books.csv' overwrite into table books;
hive>select YearOfPublication, count (BookTitle) from books group by YearOfPublication;

The PigLatin is

grunt>bks = load 'books.csv' as (ISBN:chararray, BookTitle:chararray, BookAuthor:chararray, YearOfPublication:int, Publisher:chararray);
grunt>grouped = group bks by YearOfPublication;
grunt>result = foreach grouped generate group, COUNT(bks.BookTitle);
grunt>dump result;

The Jaql is

jaql> $books = read(del("books.csv", {schema:schema {ISBN:long, BookTitle:string, BookAuthor:string, YearOfPublication:long, Publisher: string}}));
jaql> $books group by $YearOfPublication = {$.YearOfPublication} into ($YearOfPublication, count: count($)};

In IBM InfoSphere BigInsights, the queries for Hive, Pig and Jaql are the same to the above queries that are executed on the proposed system. However, in Splunk, Search Processing Language is used and **SPL** is

source ="books.csv" | stats count(BookTitle) by YearofPublication

Figure 4 illustrates the query execution time in seconds of three Big Data platforms. As a result of experiments, we can conclude that our big data platform can support large scale data analysis efficiently and

effectively.

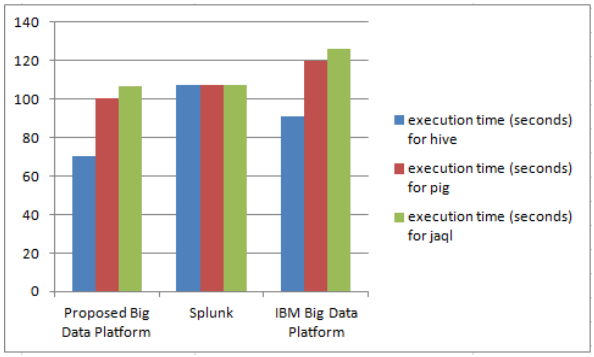

Figure 4. Query execution of three Big Data platforms

5. Conclusion

Data growth – particularly of unstructured data – poses a special challenge as the volume and diversity of data types outstrip the capabilities of older technologies such as relational databases. Organizations are investigating next generation technologies for data analytics. One of the most promising technologies is the Apache Hadoop and MapReduce framework for dealing with this big data problem. In this paper, we propose big data platform based on Hadoop MapReduce and Gluster File System and compare it with other two platforms.

References

[1]P. Carter, "Big Data Analytics: Future Architectures, Skills and Roadmaps for the CIO", IDC, September 2011.
[2]C. Eaton, T. Deutsch, D. Deroos, G. Lapis and P. Zikopoulos, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data",McGraw-Hill ,2011.
[3]Gluster, "Gluster File System 3.3-Beta 2 Hadoop Compatible Storage", August 2011.
[4]Hewlett-Packard, "HP Advanced Information Services for the Vertica Analytics Platform", January 2012.
[5]IBM Software, "IBM Netezza Analytics", Information Management, August 2011.
[6]Information Builders, "Advanced Business Analytics and the New Era of Data Warehousing", March 2011.
[7]A. Pavlo, E. Paulson and A. Rasin," A Comparison of Approaches to Large-Scale Data Analysis," in Proceedings of the 35th SIGMOD International Conference on Management of Data. ACM, 2009.
[8]P. Russom,"Big Data Analytics", TDWI best practices report, fourth quarter 2011.
[9]C. White, "Using Big Data for Smarter Decision Making", BI Research, July 2011.
[10]http://bigdataanalytics.blogspot.com
[11]http://blogs.oracle.com/datawarehousing/entry/big_data_achieve_the_impossible
[12]http://hadoop.apache.org/mapreduce
[13]http://www.infomatik.uni-greiburg.de/~cziegler/BX/BX-CSV-Dump.zip
[14]http://www.splunk.com