

Big Data Storage System Handling and Analytic Platform on Technology

M.Chithik Raja

Research scholar, AMET University, Chennai, Tamilndau, India

Dr.M.Munir Ahmed Rabbani

B.S.Abdur Rahman University, Chennai, Tamilndau, India

Chithik Raja.M

Lecturer, Salalah College of Technology, Oman

Abstract

Large-scale data processing systems released by leading Internet firms, such as Google, are being recreated by open-source territory, which can be used in a development environment. Efficient storage of large data and processing capability according to all the environmental factors are closely related to business competition. Platformization of data storage and processing functionality by many software organizations and companies enable easy use of services. Small groups or companies are now able to utilize the big data technology, which used to be exclusive to large firms, due to the difficulty in securing specialized manpower and high costs. The reason why we need to be concerned over problems that have not yet occurred is to secure sufficient response time. Sufficient time will enable us to take full consideration before making decisions as well as preparation. Therefore, it is invaluable to predict upcoming events and be concerned about future problems. To solve matters in hand, explicit recognition is needed.

Index Terms: CDR, OLTP ,OLAP,RDBMS,IaaS,PaaS ,VoltDB

Introduction

Big data means immense amount of data, so much so that it is difficult to collect, store, manage, and analyse via general database software. In general, the meaning of “immense amount of data” is classified into three types as follows:

- Volume: There is too much data to be stored and require too many processes—semantic analysis/data processing. These are the two elements that we need to understand.
- Velocity: It means storage and processing speed.
- Variety: The demand for unstructured data, such as text and images is increasing as well as refined-type data that can be standardized and previously defined like the RDBMS table record.

In addition to the three aspects, “value” is added in the standpoint that value can be created only when data is analysed. What kind of data should be stored:-We can think of data produced by corporate environment, including customer information, service log, and ERP (Enterprise Resource Planning). There are also CDR (Call Detailed Record) generated from telecom companies, machinery such as planes, oil pipelines, and power cables, or sensor data and metering data of platforms. Web servers or application logs could be a target, and there is user created data from social media blogs, mail, and bulletin boards. In short, everything that can be measured and stored could be a target for analysis. Such data has been managed and analysed from long before.

Big Data Environmental Factors

Environmental factors that make it big data attractive. With the rise of personalized and social services, restructuring is occurring in the existing Internet service environment. From the Internet Web service environment that focused on searching/porter sites, there is demand for personalized & social services in the whole service area, such as telecommunication, game, music, search, shopping, etc. This is why the scale-out technology is becoming more important than the scale-up storage. Moreover, for complicated functions, storage size, and processing requirement that analyse the relationship between individuals or personal preference, data processing technology beyond the scope of OLTP is becoming increasingly important. The universalization of the mobile environment generated the existing data and many changes were made to the consuming source environment. Users’ mobile and activity information is becoming data to be stored and analysed. Cloud-based information processing technology for portability between different devices, such as smartphones, PCs, and TVs, is a reality. It is now feasible to provide mass data storage, processing, and servicing at the appropriate level of investment through enhanced trend of the IT environment (IaaS), platform environment (PaaS), and service environment (SaaS), which are represented as cloud computing. Large-scale data processing systems released by leading Internet firms, such as Google, are being recreated by open-source territory, which can be used in a development environment. Efficient storage of large data and processing capability according to all the environmental factors are closely related to business competition.

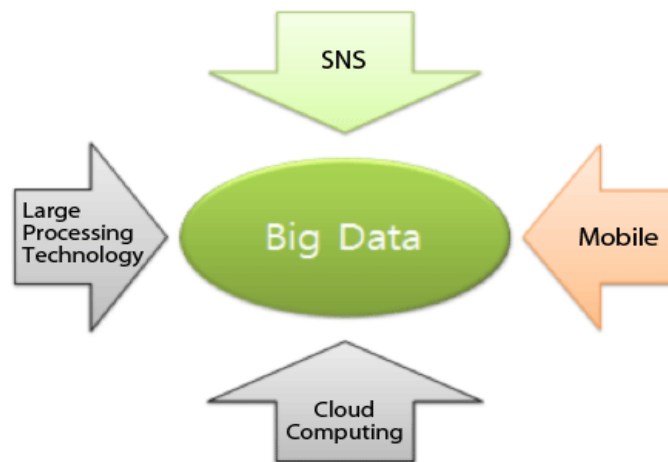


Figure 1. Big Data Environmental Factors

Storage System In Parallel DBMS

Michael Stonebraker, a pioneer who led PostgreSQL development, an open source database, made prototype systems, such as H-Store and C-Store, and demonstrated that while the existing RDBMS technology was designed to fit all areas with a single system, systems that have specialized in architecture are far more superior in handling OLAP, text, streaming, and high-dimensional data. According to Stonebraker, there have been many environmental changes compared to the time when RDBMS was first designed and implemented; thus, such changes should be reflected to the existing OLTP processing area. For example, he said that disk in the 1970s used to be regarded as memory nowadays. The disk log and disaster recovery system by tape backup should be changed to a high-cost structure of the “k-safe environment” (that is, duplication), and reconsider the concurrency technique of conventional lock method according to changes in process technology.

VoltDB

VoltDB is a system consisting of a suitable format to a high-performance OLTP environment. The system is not memory-based data processing or SQL, but it performs sequential processing for data split based on stored procedure and reduces lock overhead with communication, helping to configure the high-speed OLTP system through horizontal split for table data^[2]. Figure 2 displays that a certain task that requires to operate in just one partition is executed sequentially in the corresponding partition, and that a certain task that needs to be handled in several partitions are processed by a coordinator. If there are many operations that need to be processed in several partitions, large rows and sizes may not be good.

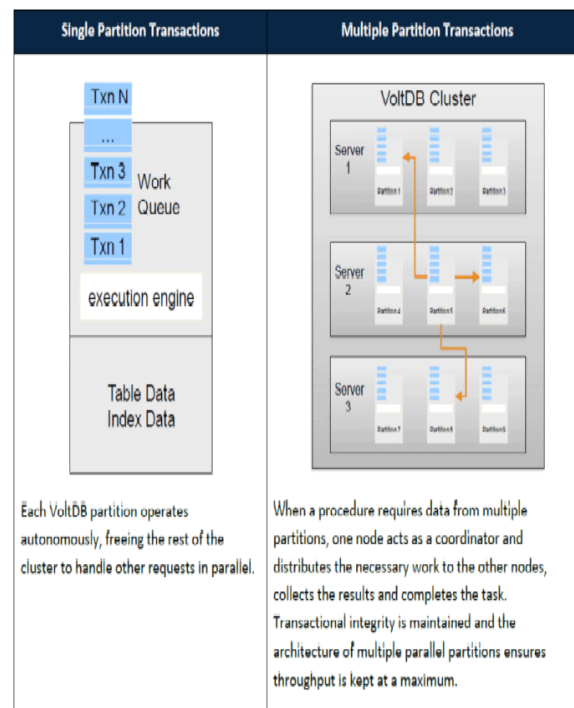


Figure 2. VoltDB Architecture

Sap Hana

SAP HANA is a memory-based storage made from SAP. Its characteristic is to organize a system optimized to analysis tasks, such as OLAP. If all data is inside system memory, maximizing CPU utilization is crucial and the key point is to reduce bottlenecks between memory and CPU cache. In order to minimize Cache miss, consecutive data for processing within the given time is more advantageous, meaning that configuration of column-oriented tables could be favorable when analyzing many OLAP.

There are many advantages of the column-oriented table configuration and typical examples are a high data compression ratio and processing speed. In case of the same data domain, several data domains are better for data compression than when they are combined together. Moreover, the configuration enables reducing CPU operations through a lightweight compression, such as RLE (Run length encoding) or dictionary encoding or executing desired operations without a recovery process for compressed data. The following figure shows a brief comparison with Row-oriented and Column-oriented methods.

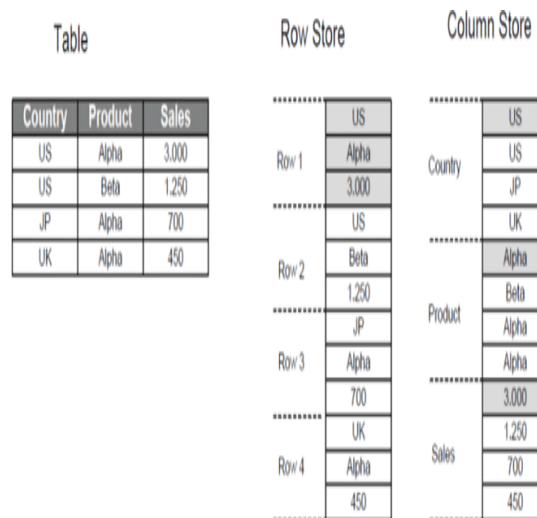


Figure 3.1 Comparison with Row-oriented and Column-oriented methods.

Vertica

Vertica is database specialized for OLAP, which stores data on disk via the column method. The Shared-nothing-oriented MPP structure comprises a storage optimized for writing so as to load data fast, a reading storage in a compressed type, and tuple mover that manages bilateral data flow. Figure 4 below helps to understand the Vertica structure.

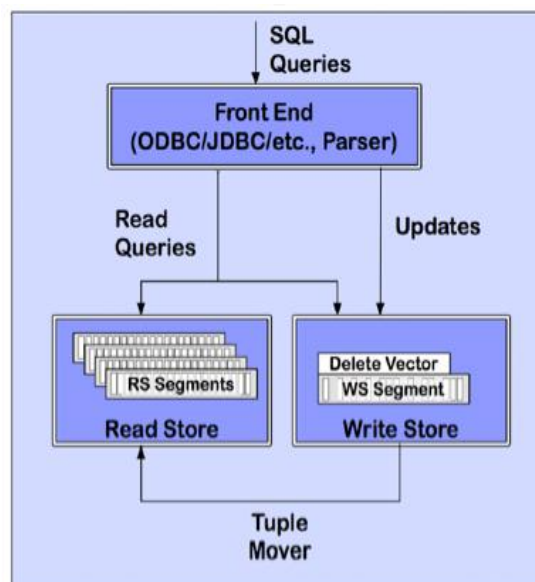


Figure 4. VERTICA Structure

Greenplum

Greenplum database is a shared-nothing MPP structure, generated based on PostgreSQL. Data to be stored can select Row-oriented or Column-oriented methods

accordingly to operations apply to the corresponding data. Data is stored in a server in segment and have availability because of segment unit replication of the log shipping method. A query engine, which was developed based on PostgreSQL, is configured to execute SQL basic operation (hash-join, hash-aggregation) or a map-reduce program so as to effectively process parallel query or map-reduced type programs. Each process node is connected to software-oriented data switch component.

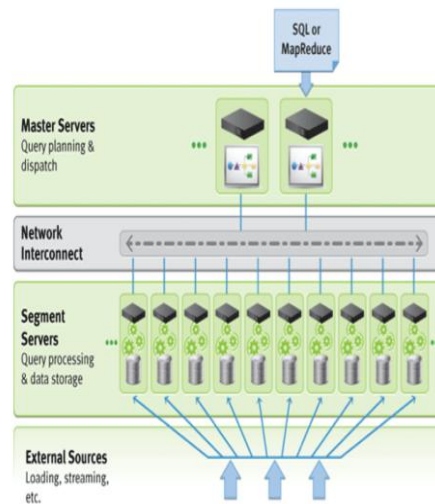


Figure 5. Greenplum Architecture

IBM Netezza Data Warehouse

IBM Netezza data warehouse has a two-tier type architecture consisted of SMP and MPP, called AMPP (Asymmetric Massively Parallel Processing). A host with a SMP structure operates query execution plan and aggregation results, while S-blade nodes with a MPP structure handles query execution.

Each S-blade is connected by a special data processor called FPGA (Field Programmable Gate Array) and disk. Each S-blade and host is connected to network that use IP addresses.

Unlike other systems, FPGA has filtering for data compression, record or column; in transaction processing, it enables filtering or transformation functions, such as visibility check during retrieving data from disk memory for real-time processing. When processing large-date, it adheres to the principles (processing close to the data source), which is to reduce as much unnecessary data as possible from transmission by performing data operation where data is located.

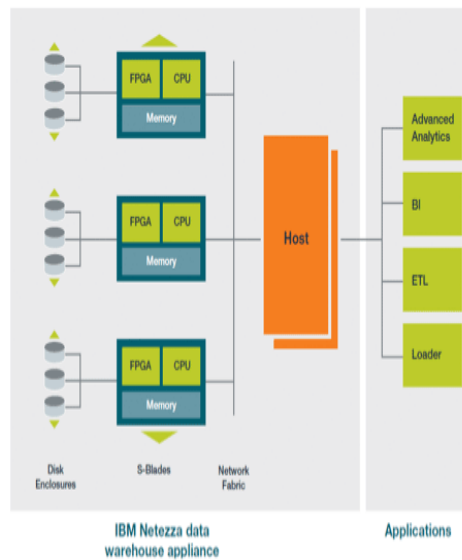


Figure 6. IBM Netezza data Architecture

There are many other solutions for big data processing, namely Teradata, Sybase, and Essbase, traditional and powerful programs. Parallel DBMS is advanced form of old RDBMS, which in many cases have a MPP structure. In addition, companies and organizations that develop parallel DBMS are taken over by IT conglomerates and are in the progress of development in the appliance type. The names of conglomerates and date of acquisition of aforementioned parallel DBMS are shown in the following table:

| Table 1: Names of companies which acquired parallel RDBMS. | | |
|---|------------------------------|------|
| The names of companies acquired | Database | Year |
| SAP | Sybase | 2010 |
| HP | Vertica | 2011 |
| IBM | Netezza | 2010 |
| Oracle | Essbase (Hyperian Solutions) | 2007 |
| Teradata | Aster Data | 2011 |
| EMC | Greenplum | 2010 |

NOSQL

In RDBMS, scaling out while supporting ACID (Atomicity, Consistency, Isolation, and Durability) is almost impossible. For storage, data had to be divided into several devices; to be satisfied with ACID that has divided data, you have to use complicated locking and replication methods, which will lead to performance degradation.

NoSQL, a general term for a new storage system has emerged in order to simplify data models for easy definition of shard, which is the basic of distribution, and to make requirements less strict (*Eventual Consistency*) in a distribution replication environment or constraint isolation. Since NoSQL is covered many times in our DevPlatform Blogs and there are many places to obtain information, we will not

go over the NoSQL products. Processing Aspects The key point of parallel processing is Divide and Conquer. That is, data is divided in an independent type and process it in parallel. Just imagine the matrix multiplication that can divide and process each operation. The meaning of big data process is dividing a problem into several small operations, and combine them into a single result. If there is operation dependence, it is certainly impossible to make the best use of the parallel operation. It is necessary to save and process data considering these factors.

Map-Reduce

The most widely known technology that helps to handle large-data would be a distribution data process framework of the Map-Reduce method, such as Apache Hadoop.

Data processing via the Map-reduce method has the following characteristics:

1. It operates via regular computer that uses built-in hard disk, not a special storage. Each computer has extremely weak correlation where expansion can be hundreds and thousands of computers.
2. Since many computers are participating in processing, system errors and hardware errors are assumed as general circumstances, rather than exceptional.
3. With a simplified and abstracted basic operation of Map and Reduce, you can solve many complicated problems. Programmers who are not familiar with parallel programs can easily perform parallel processing for data.
4. It supports high throughput by using many computers.

The following figure displays the implementation flow of the map-reduce method. Data stored in the HDFS storage is divided to available worker and expressed (Map) a value type, and results are stored in a local disk. The data is complied by reducing worker and generate a result file.

Depending on the characteristics of a data storage, make the best use of locality by reducing the gap between a node which is processing data and source data location by placing worker in the location (based on network switch) where data is stored. Each worker can be implemented in various languages through streaming interface (standard in/out).

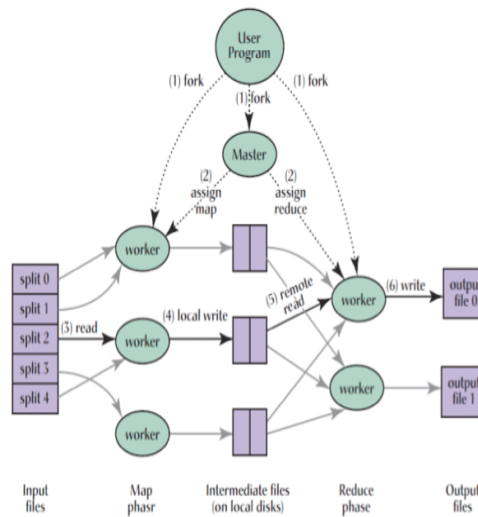


Figure 7 : Map-Reduce execution.

Dryad

Like Dryad, there is a framework that enables processing parallel data by forming the data channel between programs in a graph type. Developers who use Map/reduce framework should write Map and Reduce functions; thus, if using Dryad, they need to make a graph which processes the corresponding data. Figure 8 below describes data processing in Dryad in comparison with the UNIX's pipe.

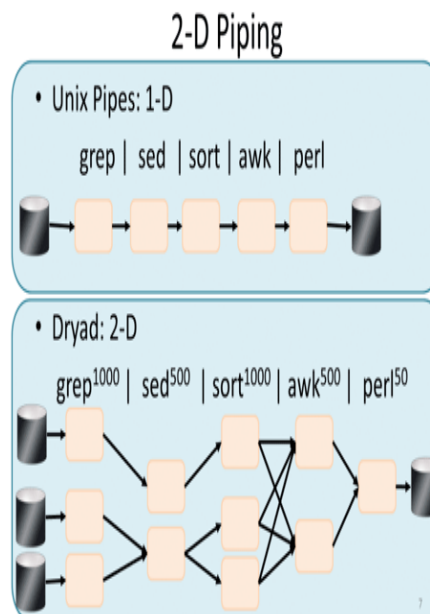


Figure 8: Dryad Data Processing

Dryad enables data flow processing of the DAG (Direct Acyclic Graph) type. Despite parallel data operation framework offers enough functions for processing big data, there are the barriers to entry in using the system for inexperienced

developers, data analysers, and data minors. As a result, we need a method that helps easier data processing through higher standard abstract.

Apache Pig

Apache Pig provides a high-standard data processing structure, allowing large data processing through combination. It supports a language called, Pig Latin, and shows the following characteristics:

1. It provides a high-standard structure, such as relation, bag, and tuple, including int, long, and double basic types.
2. It supports relation (relation and table) operation, such as FILTER, FOREACH, GROUP, JOIN, LOAD, and STORE.

A data processing program, which is specified via Pig Latin, converts to a logical execution plan and this is again converted to a Map-Reduce execution plan for execution. Figure 10 below shows Pig operation process.

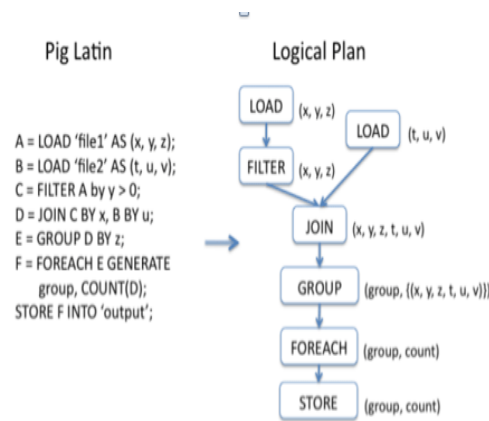


Figure 9: Converting Process of Pig Latin To Map-Reduce

Apache Pig is an access method, enabling a large data processing program in procedural programming languages, namely C or Java. Another similar access method is Google's Sawzall. Some technologies help declarative data processing, like SQL, without specifying data processing procedures as in programming languages. Typical examples are Apache Hive, Google Tenzing, and SCOPE of Microsoft.

Apache Hive

Apache Hive helps to analyse large data by using the query language called HiveQL for data source, such as HDFS or HBase. Architecture is divided into Map-Reduce-oriented execution, meta data information for a data storage, and an execution part that receives a query from user or applications for execution.

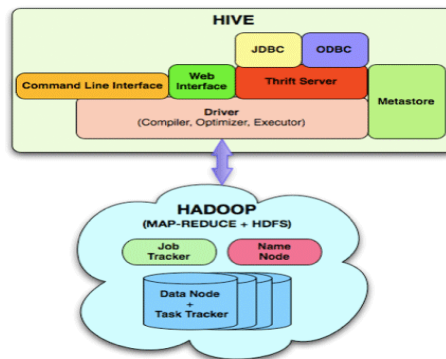


Figure 10: HIVE Architecture

To support expansion by user, it allows user specified function at the scalar value, aggregation, and table level.

Analysis Aspects

We reviewed systems that store big data and procedural/declarative technologies that display processing and how to process large data. Finally, let us look into technology that analyzes big data.

The process of finding meaning in data is called KDD (Knowledge Discovery in Databases). KDD is to store data, process/analyze the whole or part of interested data in order to extract progress or meaning value, or discover facts that were so far unknown and make them into knowledge ultimately.

For this, various technologies are comprehensively applied, such as artificial intelligence, machine learning, statistics, and database.

GNU R

GNU R is a software environment comprising program languages specialized for statistics analysis and graphics (visualization) and packages. It ensures a smooth process of vector and matrix data so as to be optimized for statistical calculations in terms of language. You can easily acquire desired statistics process library because of the R package site known as CRAN (Comprehensive R Archive Network). It can be touted as an open source in the field of statistics. In the past, R used to put data to be processed into the memory of a computer for analyzing using a single CPU. There has been much progress due to ever increasing data to be processed. Some of the typical examples are as follows:

1. doSMP package: It uses multi-core.
2. Bigmemory package: It stores data in the shared memory. It stores only the reference for memory value and stores the value in a disk.
3. snow package: It enables the R program execution in the computer cluster environment.
 - a. The following figure demonstrates how snow package processes in a cluster environment. A typical division and conquer method.

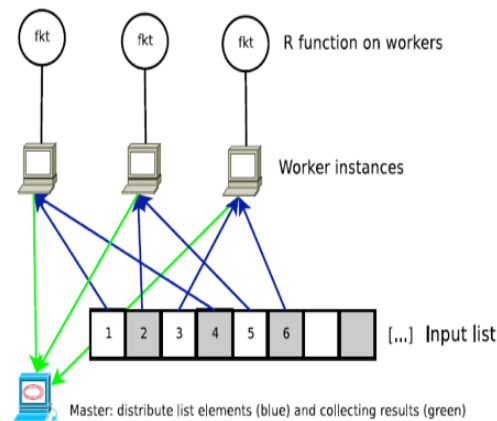


Figure 11: Execution environment for GNU R and Snow Package.

As the Map-Reduce operation is becoming more common, we can see technologies, such as RHIPE, RHadoop package, and Ricardo, where MR operation and R are in the form of integration. There is also the rhive package, which combined with high-standard data processing technology, like Apache Hive.

Apache Mahout

Apache Mahout is implemented to expand data analysis algorithm. Not only Apache Hadoop, but it can be operated in a variety of environments. It also offers effective packages in terms of storing the mathematical library and Java Collection, such as vector matrix as well as a function to use DBMS or other NoSQL database as data sources. Besides, there is a trend in which integrates and supports R to its own technology that is provided from a big data storage product of the existing MPP structure. For example, Oracle big data appliance enables execution of the R program, targeting data stored in Oracle database by integrating with R. To summarize, the data analysis technology, such as GNU R is progressing towards technology convergence from the existing large data processing framework or operation on storage systems.

Conclusion

We have looked into platform technology that processes big data by classifying it into storage, process, and analysis. We also briefly reviewed storage systems, large data processing and analysis technologies. The commercial impacts of the Big data have the potential to generate significant productivity growth for a number of vertical sectors. They should also create healthy demand for talented individuals who are capable to help organizations in making sense of this growing volume of raw data. In short, Big Data presents opportunity to create unprecedented business advantages and better service delivery. A regulatory framework for big data is essential. That framework must be constructed with a clear understanding of the ravages that have been wrought on personal interests by the reduction of information to data, its centralization, and its expropriation but the biggest gap is the lack of the skilled

managers to make decisions based on analysis by a factor of 10x. Growing talent and building teams to make analytic-based decisions is the key to realize the value of Big Data.

References

- [1] Addressing Big Data Problem Using Hadoop and. Nirma University, Gujrat: Nirma University. Singh, S., & Singh, N. (2012).
- [2] Aveksa Inc. (2013). Ensuring “Big Data” Security with Identity and Access Management. Waltham, MA: Aveksa. Hewlett-Packard Development Company. (2012).
- [3] The Awesome Ways Big Data is used Today to Change Our World. Retrieved November 14, 2013, from LinkedIn: <https://www.linkedin.com/today/post/article/20131113065157-64875646-the-awesome-ways-big-data-is-used-today-tochange-our-world> Patel, A. B., Birla, M., & Nair, U. (2013).
- [4] Big Security for Big Data. L.P.: Hewlett-Packard Development Company. Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013).
- [5] Big Data: Issues and Challenges Moving Forward. International Conference on System Sciences (pp. 995-1004). Hawaii: IEEE Computer Society. Katal, A., Wazid, M., & Goudar, R. H. (2013).
- [6] Big Data: Issues, Challenges, Tools and Good Practices. IEEE, 404-409. Marr, B. (2013, November 13).
- [7] Data Analytics. International Conference on Communication, Information & Computing Technology (ICCICT) (pp. 1-4). Mumbai: IEEE.
- [8] J. François et al., “BotCloud: Detecting Botnets Using MapReduce,” Proc. Workshop Information Forensics and Security, IEEE, 2011, pp. 1–6.
- [9] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In WWW '10: 19th International Conference on World Wide Web, pages 591–600, New York, New York, USA, 2010. ACM, ACM Press. ISBN 9781605587998.
- [10] U. Kang, CE Charalampos E. Tsourakakis, Ana Paula Appel, Christos Faloutsos, and Jure Leskovec. HADI: Mining Radii of Large Graphs. ACM Transactions on Knowledge Discovery from Data, 5(2):1–24, 2011.
- [11] http://viewer.media.bitpipe.com/1240246133_118/1258558418_168/sCompliance_security_Data-Protection_final.pdf
- [12] http://www.deloitte.com/assets/Dcomobal/Local%20Assets/Documents/TMT/dttl_TMT%202011%20Global%20Security%20Survey_High%20res_191111.pdf
- [13] www.wordsinspace.net/course.../MatternLiteratureReviewTips.pdf
- [14] .T.-F. Yen et al., —Beehive: Large-Scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks, to be published in Proc. Ann. Computer Security Applications Conference (ACSAC 13), ACM, Dec. 2013.
- [15] .Zach Miller, Dan Bradley, Todd Tannenbaum, and Igor filigoi. Flexible session management in a distributed environment. Journal of Physics: Conference Series, 219(4):042017, 2010.

