Evaluation of Efficient Implementation of Big Data Switch for Iraqi Cellular Phone Service Providers

R.A. Mahmood
Dep. of Comp. sciences, Faculty of Comp. and Info., Mansoura University, Egypt

M.Z. Rashad
Dep. of Comp. sciences, Faculty of Comp. and Info., Mansoura University, Egypt

M.A. El-Dosuky
Dep. of Comp. sciences, Faculty of Comp. and Info., Mansoura University, Egypt

ABSTRACT
In this paper, we perform extensive evaluation of an implementation of cellular phone service providers switching based on big data technology.

Evaluation is based upon size, speed and usability measures.

The system is faster, with less size, and more usability than other implementations.

Keywords
Big Data, MapReduce, Hadoop, Hadoop Distribute File System, cellular phone service provider

1. INTRODUCTION
Big data is opening up new chance for enterprises to extract insight from large volumes of data in real time and across multiple relational and nonrelational data types [1].

It is becoming one of the most important technology direction that has the potential for dramatically changing the way organizations use information to enhance the customer experience and transform their business models [2]. Big Data analysis is best left to software programs. Not so. When data analysts go straight to the complex calculations, before they perform a simple estimation, they will find themselves accepting wildly measly calculations [3].

Big Data is a relatively new term that came from the need of big companies like Google, Yahoo, Facebook to analyze big amounts of unstructured data, but this need could be identified in a number of other big enterprises as well in the development and research field [4].

This paper is organized as follows:

Section 2: reviews the related work. Sectoin 3: Evaluation metrics.

2. RELATED WORK

2.1 Big Data
Big data is defined as large amount of data which requires new technologies and architectures so that it becomes possible to extract value from it by capturing and analysis process. Due to such large size of data it becomes very difficult to perform effective analysis using the existing traditional techniques [5]. big data is a term for massive data sets having large, more varied and complex structure with the difficulties of storing, analyzing and visualizing for further processes or results. The process of research into massive amounts of data to reveal hidden patterns and secret correlations named as big data analytics[6].

It's concern large-volume, complex, growing data sets with multiple, autonomous sources. With the fast development of networking, data storage, and the data collection capacity, Big Data are now rapidly expanding in all science and engineering domains, including physical, biological and biomedical sciences[7].

Big data due to its various properties like volume, velocity, variety, value and complexity put forward many challenges. Since Big data is a recent upcoming technology in the market which can bring huge benefits to the business organizations, it becomes necessary that various challenges and issues associated in bringing and adapting to this technology are brought into light[5].

Velocity of the data is used to define the speed with which different types of data enter the enterprise and are then analyzed[8].

Volume refers to the growth and run rates of data. It may be in KB, MB, GB, TB, or PB based on the type of the application that generates or receives the data.

Variety refers to the various types of the data that can exist, for example, text, audio, video, and photos [9].

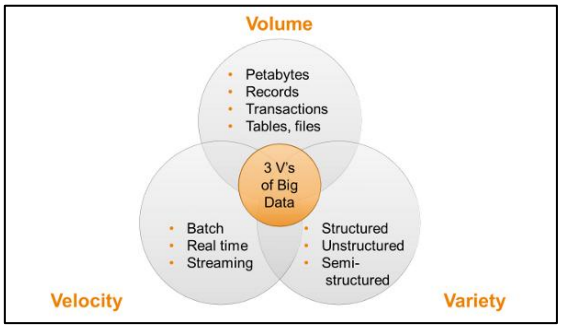

Fig 1: 3Vs Model Big Data

The Data warehouse model is constructed from two relational data model schemas covering demographics and inventory-accounting. The inventory-accounting database has millions of rows, providing a reasonable amount of data to demonstrate the tuning process. There are multifarious methods that can be used to regulation a data warehouse data model [10].

It is a database that containing data from several operational systems that has been consolidated, integrated, aggregated, and structured, so that it can be used to support the analysis and decision-making process of a business [11].

The big data analytics community has accepted MapReduce as a programming model for processing massive data on distributed systems such as a Hadoop cluster. MapReduce has been evolving to improve its performance [12].

MapReduce is a scalable parallel programming model for big data processing, and it was inspired by the Map and Reduce primitives from functional languages. Its first implementation was designed to run on large clusters of homogeneous machines. Though, in the last years, the model was ported to different types of environments, such as desktop grid and volunteer computing. [13].

Google engineers designed MapReduce to solve a specific practical problem. Therefore, it was designed as a programming model combined with the implementation of that model — in essence, a reference implementation.[14]

Hadoop is the Apache Software Foundation top-level Apache project, It allows for the distributed processing of large datasets across clusters of computers [15].

Hive is another component in the Hadoop-based architecture, it is a data warehousing infrastructure, and provide analysis like data store. It's built on top of Hadoop providing table based abstraction over HDFS, which makes it easy to load structured data [16].

Hive was created to make it possible for analysts with strong SQL skills [17] with a simple SQL-lite implementation called HiveQL without sacrificing access via mappers and reducers. As a result, Hive is best used for data mining and deeper analytics that do not require real-time behaviors [14]. Hive uses three mechanisms for data organization: Tables, Partitions and Buckets [18].

Pig is a high-level language which is a procedural data processing language designed for Hadoop[19]. Like actual Pigs, who eat almost anything, the Pig programming language is designed to handle any kind of data [20]. Pig was designed to make Hadoop more approachable and usable by nondevelopers [18].

2.2 Hadoop Distributed File System (HDFS)

HDFS is Hadoop's own rack-aware file system, which is a UNIX-based data storage layer of Hadoop. It's derived from concepts of Google file system [9]. This file system is meant to support enormous amounts of structured as well as unstructured data [21].

It is consists of: Name Nodes, contacted by clients to locate information and DataNode that retains a segment of data in the HDFS and acts as a computer platform for running jobs, some of which will utilize the local data within the HDFS[22].

The HDFS is a distributed file system designed [23] for the performance and reliability requirements of huge datasets [24], it have characteristics common with other distributed file schemes. The difference is one noticeable aspect is HDFS's write-once-read-many model that relaxes concurrency control requirements, and simplifies data coherency. HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients [25].

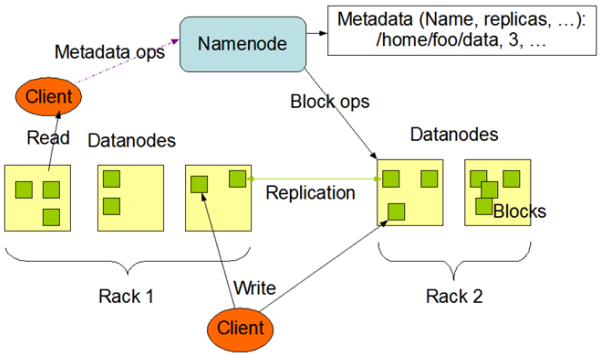

Fig 2: HDFS Architecture

HDFS provides interfaces for applications to migrate them closer to where the data is located [26]. Hadoop doesn't require expensive, highly reliable hardware. It's designed to run on clusters of commodity hardware for which the chance of node failure across the cluster is high, at least for large clusters. HDFS is designed to carry on working without a noticeable interruption to the user in the face of such failure [17].

2.3 k – mean Algorithms

k-means is mostly used as the clustering algorithm in data science. This algorithm asks for a number of clusters to be the input parameters from the user side [9]. It gathers points around a predefined number of clusters, k. The idea is to help uncover clusters that occur in your data so you can investigate unusual or previously unknown patterns in your data.

The selection of the k clusters is somewhat dependent on the data set you want to cluster. You can start out with a small number, run the algorithm, look at the results, increase the number of clusters, rerun the algorithm, look at the results, and so on [27].

While looping give every point in proximity to closes mean. And position the 'mean' to the center of its cluster [28].

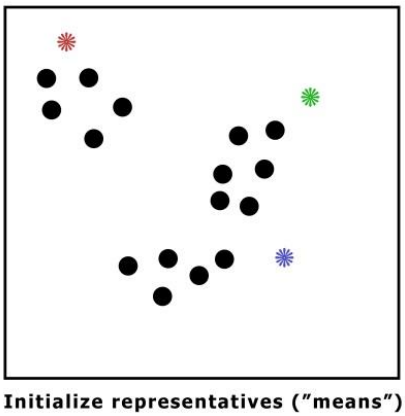

Fig 3: k – mean algorithm

2.4 Map and Reduce

MapReduce is a computing model that decomposes large data manipulation jobs into individual tasks that can be executed in parallel across a cluster of servers. The results of the tasks can be joined together to compute the final results [29].

MapReduce programming model serves as an example for processing large data sets in a enormous parallel fashion, so the computation takes a set of input key/value pairs, and produces a set of output key/value pairs [30]. MapReduce comes from the two fundamental data-transformation operations used, map and reduce. The map function divides a query into multiple parts and processes data at the node level. The reduce function aggregates the results of the map function to determine the answer to the query [16].

A map operation converts the elements of a collection from one form to another. In this case, input key-value pairs are converted to zero-to-many output key-value pairs, where the input and output keys might be completely different and the input and output values might be completely different [29].

The map and reduce functions in Hadoop MapReduce consist of the following format:

map: (K1, V1) → list(K2, V2)

reduce: (K2, list(V2)) → list(K3, V3) [31]

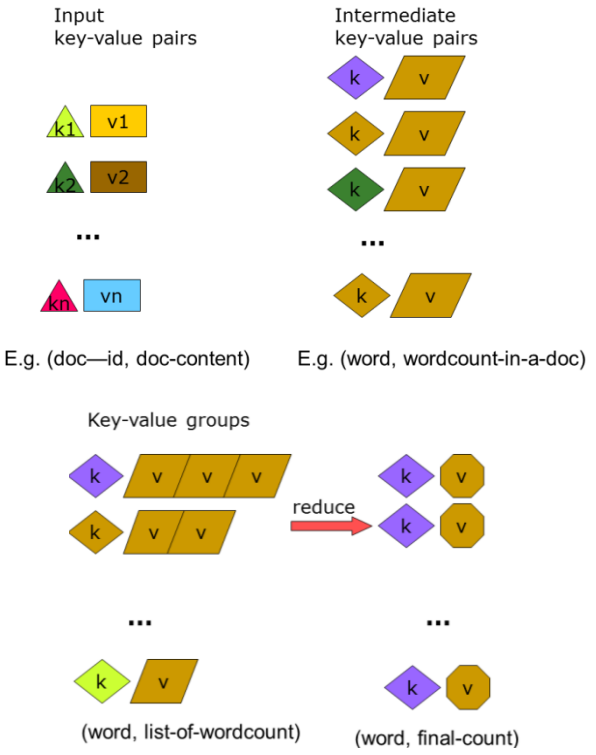

Fig 4: MapReduce in action

All the key-pairs for a given key are sent to the same reduce operation. Specifically, the key and a collection of the values are passed to the reducer. A final key-value pair is emitted by the reducer. Again, the input versus output keys and values may be different. Note that if the job requires no reduction step, then it can be skipped [31].

3. IMPLEMENTATION OF BIG DATA SWITCH

The major three cell phone networks in Iraq: AsiaCell, Korek, and Zain. Recently a framework is proposed to build a switch for Iraqi cellular phone service providers is shown in the next figure [32].

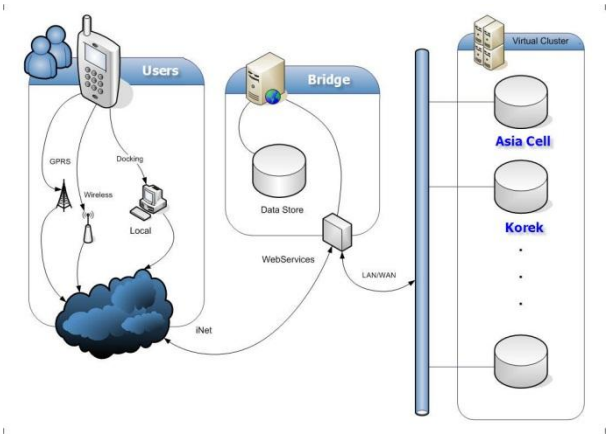

Fig 5: proposed framework

The Bridge utilizes the Namenode facility in HDFS, and is easily implemented as a map.

```
$bridge = array(
        "0770" => "asiacell",
        "0750" => "korek"
        );
```

Notice that each network has its own cluster.

3.1 Test Scenario

The following scenario is executed for user with id 07701234567.

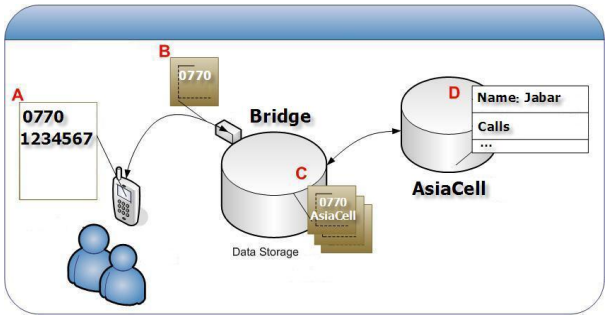

Fig 6: Case study

3.2 Characteristics of the architecture

The solution architecture have the following characteristics:

- Accessibility of data and content anywhere and anytime via mobile devices.
- Scalability to accommodate large number of users.
- Extensibility of functionality in the system environment
- Clustering/acceleration—offers a framework to cluster application components for load balancing setup.
- Caching—offers a framework to cluster application components to share runtime data, as well as data caching mechanism for increased performance.
- Event logging—it has a centralized logging framework to enables tracking user operations done via the exposed user interfaces.
- Security —as it supports a secure (SSL) login.

4. EVALUATION METRICS

4.1 Size

The usual phpmyadmin implementation has extra space for managing the Database.

This implementation has no extra space.

The size of all the data base is the size of the data itself.

Table 1. Comparison of size

	Size of phpmyadmin implementation	Size of proposed implementation
100 users	170 KB	200 KB
1.000.000 users	2.6 GB	1.9 GB

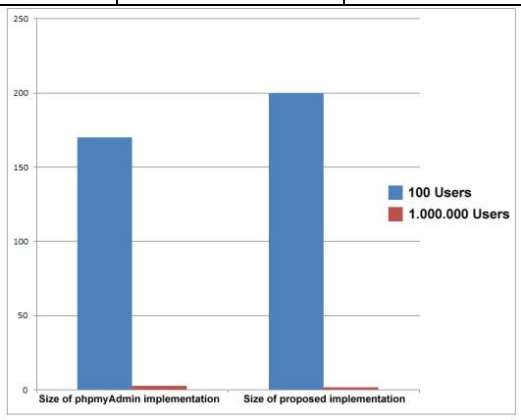

Fig 7: Comparison of size

4.2 Speed

Given the id, the proposed technique can behave in a random access way. This is compared to the traditional implementation of phpmyadmin which has an index.

Table 2. Time of finishing a request

	Time phpmyadmin	Time proposed system
1000.000 users	1.006 seconds	0.0047 seconds

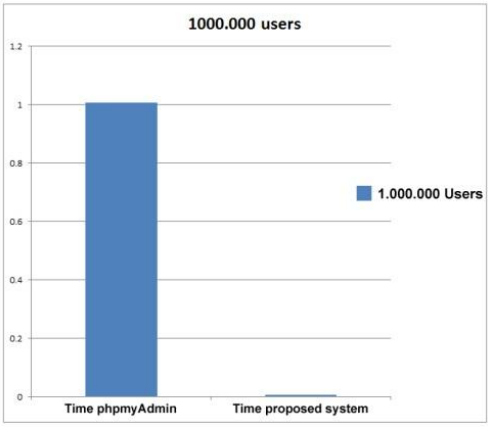

Fig 8: Time of finishing a request

4.3 Usability metrics

Table 3. Usability metrics

Usability metrics	Phpmyadmin	Proposed system
Availability and Accessibility	Yes	Yes
Clarity	Yes	Yes
Learnability	No	Yes
Credibility	No	Yes
Relevancy	Yes	Yes

5. CONCLUSION and FUTURE WORK

This paper reviews Big data technology, then proposes a framework to build a switch for Iraqi cellular phone service providers. The major three cell phone networks in Iraq: AsiaCell, Korek, and Zain. Investigations show that it is very promising and could be seen as a good optimization.

One problem to consider is the evaluation criterion. Open Shortest Path First (OSPF), a traditional link-state routing protocol for Internet Protocol (IP), can be specified [33], with back-compatibility for current infrastructures [34]. This encourage us to propose the OSPF overhead as one of evaluation measures for comparing algorithms proposed for networks of increasing size. show the OSPF overhead generated by the OPNET [35].

6. REFERENCES

[1] Nitin, S. and Himanshu, S. 2013. Big Data Application Architecture Q & A., President and Publisher: Paul Manning, ISBN: 978-1-4302- 6292-3.

[2] Judith, H., Alan, N., Dr. Fern H. and Marcia K. 2013. Big Data For Dummies.by John Wiley & Sons, Inc., Hoboken, New Jersey,ISBN: 978-1-118-50422-2

[3] J. B. Jules. 2013. Simple but Powerful Big Data Techniques, Principles of Big Data. Morgan Kaufmann, pp. 99-127, 2013.

[4] Garlasu, D., Sandulescu, V., Halcu, I. and Neculoiu, G. 2013. A big data implementation based on Grid computing. Roedunet International Conference (RoEduNet), 11th, Page(s): 1 – 4, ISBN: 978-1-4673-6114-9.

[5] Katal, A., Wazid, M., and Goudar, R.H. 2013. Big data: Issues, challenges, tools and Good practices. Page(s): 404 – 409, ISBN:978-1-4799-0190-6.

[6] Sagiroglu, S., and Sinanc, D. 2013. Big data: A review. Collaboration Technologies and Systems (CTS). Page(s):42 – 47, ISBN: 978-1-4673-6403-4.

[7] Wu, X., Zhu, X., Wu, G., and Ding, W. 2014. Data Mining with Big Data. Knowledge and Data Engineering, IEEE Transactions, Page(s): 97 – 107 , ISSN :1041-4347.

[8] Nitin S., and Himanshu S. 2013. Big Data Application Architecture Q & A. President and Publisher: Paul Manning, ISBN: 978-1-4302- 6292-3.

[9] Vignesh, P. 2013. Big Data Analytics with R and Hadoop. Packt Publishing, ISBN 978-1-78216-328-2.

[10] Gavin P. 2006. Warehouse Tuning for 10g. Digital Press, Burlington, Pages 31-47, Oracle Data Warehouse Tuning for 10g, ISBN 9781555583354, http://dx.doi.org/10.1016/B978-155558335-4/50003-8.

[11] T. Dimitri, and S. Timos. 1999. Designing data warehouses, Data & Knowledge Engineering. vol 31, no. 3, pp 279-301, 1999.

[12] Martha, V.S., Weizhong Zh., and Xiaowei Xu. 2013. h-MapReduce: A Framework for Workload Balancing in MapReduce. Advanced Information Networking and Applications (AINA). IEEE 27th International Conference on 25-28 March 2013 , Page(s): - 637 – 644, ISBN:978-1-4673-5550-6

[13] Wagner, K., Pedro de B., Marcos, Julio C.S. Anjos., Alexandre K., Claudio R., and Luciana B. MRSG – A MapReduce simulator over SimGrid. Parallel Computing, Volume 39, Issues 4–5, April– May 2013, Pages 233-244, ISSN 0167-8191.

[14] K. Wagner, D. M. Pedro, C.S. A. Julio, K.S. M. Alexandre, R. G. Claudio, and B. A. Luciana. 2013. MRSG – A MapReduce simulator over SimGrid", Parallel Computing, vol 39, no. 4–5, pp 233-244.

[15] Paul, C. Zikopoulos , Chris, E., Dirk, d., Thomas, D., and George, L. 2012. Understanding Big Data. Printed in USA, ISBN:978-0-07-179053-6 IBM.

[16] Soumendra, M., Madhu, J., and Harsha S. 2013. Big Data Imperatives. President and Publisher: Paul Manning, ISBN: 978-1-4302-4872-9.

[17] Tom, W. 2012. Hadoop: The Definitive Guide. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2012, ISBN: 978-1-449-31152-0

[18] H. Judith, N. Alan, H. Fern, and K. Marcia. 2013. Big Data For Dummies. Wiley and sons Inc.

[19] Pete, W. 2011. Big Data Glossary. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA, ISBN:978-1-449-31459-0.

[20] C. Z. Paul, E. Chris, D. Dirk, D. Thomas, and L. George, 2011. Understanding Big Data. IBM.

[21] Robert, D. Schneider. 2012. Hadoop For Dummies. by John Wiley & Sons Canada , Ltd., ISBN: 978-1-118-25051-8.

[22] Dell, Hadoop White Paper Series http://i.dell.com/sites/content/business/solutions/ whitepapers/en/Documents/hadoop- introduction.pdf

[23] Dhruba, B. 2007. Hadoop Distributed File System Version Control System", The Apache Software Foundation. Website: http://hadoop.apache.org/hdfs/ version_control.html

[24] O'Reilly Media. 2011. Big Data Now. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, ISBN: 978-1-449-31518-4.

[25] Dhruba, B. HDFS Architecture. Website: http://hadoop.apache.org/common/docs/r0.20.0/hdfs_design.html

[26] Hadoop Distributed File System (HDFS). Web site http://www.j2eebrain.com/java-J2ee-hadoop-distributed-file-system-hdfs.html

[27] Kevin, S., and Christopher, P. 2014. Programming Elastic MapReduce. Printed in the United States of America, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, ISBN: 978-1-449-36362-8.

[28] Databases. 1998. Proc. 4 th International Conf. on Knowledge Discovery and Data Mining (KDD-98). AAAI Press, Aug.

[29] Edward, C., Dean, W. and Jason, R. 2012. Programming Hive. Edward Capriolo, Aspect Research Associates, and Jason Rutherglen.

[30] J. Dean, and S. Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters", in: OSDI'04, 6th Symposium on Operating Systems Design and Implementation, Sponsored by USENIX, in cooperation with ACM SIGOPS, pp. 137–150.

[31] Dhruba, B. 2007. Hadoop Distributed File System Version Control System , The Apache Software Foundation.Web site: http://hadoop.apache.org/hdfs/version_control.html

[32] Rebwar, A., Majdi, Z., and Muhammed, A. 2014. Efficient Implementation of Big Data Switch for Iraqi Cellular Phone Service Providers. vol 3, no.4.

[33] Moy, J. OSPF Version 2. RFC2328, Available at http://www.ietf.org/rfc/rfc2328.txt

[34] Baker, F. et al. 2013. Problem Statement for OSPF Extensions for Mobile Ad Hoc Routing. Available at http://tools.ietf.org/html/draft-baker-manet-ospfproblem-statement-00.

[35] Riverbed application and network performance management solutions. http://www.opnet.com/. 2013