Performance Evaluation and Comparing of Big Data Platforms on Implementing K-Nearest Neighbors (K-NN) Algorithm

Mohammad Ahmadi and Amir Jahangard Rafsanjani
Computer Engineering Department, Faculty of Engineering, Yazd University, Yazd, Iran
Ahmadi_m@stu.yazd.ac.ir, jahangard@yazd.ac.ir

Ali Mostafaeipour
Industrial Engineering Department, Yazd University, Yazd, Iran
mostafaei@yazd.ac.ir

Abstract

One of the major challenges in the current prospect of big data is the inability to process a large volume of data at an acceptable time. Hadoop and Spark are two framework for distributed data processing. Hadoop is a very popular and general framework for big data processing. Spark is also as an open source framework for in-memory programming model to process return algorithms. In this paper, Hadoop and spark data processing framework have been evaluated and compared in terms of runtime, memory usage, Central Processing Unit (CPU), and network utilization. Thus, K-Nearest Neighbors (k-NN) Common Machine Learning Algorithm was implemented on data collection with various sizes and run on Hadoop and Spark framework. The obtained results show 2 to 4 times superiority of Spark compared to Hadoop within the implementation of the program. Evaluations show that Hadoop use sources including central processor and network more. On the other hand, memory usage is more in spark than that in Hadoop.

Keywords
BigData; Hadoop; Spark; Ganglia, K-Nearest Neighbors (K-NN).

1. Introduction

Along with the advancement of technology and convenient accessibility to Internet, today, sources producing information such as social networking, personal information, Website and blog content, searches of search engines, and news have been increased. Subsequently, the volume of information produced in the world has significantly grown. Accordingly, we have faced a new concept called big data will be faced. Big data refers to big and complex data set whose processing is impossible or difficult by traditional data processing software (Chen et al, 2014). The volume of information is a big challenge in traditional data processing systems so that in order to cope with these challenges, the types of cluster computing frameworks aiming to support large-scale data on conventional machines have been created (Gu and Li, 2013).

Map-reduce is one of the successful frameworks for processing big data sets with scalable, reliable, and high fault-tolerant features introduced in 2004 by Google (Dean and Ghemawat, 2008). Hadoop developed by Apache is an implementation of open source of map-reduce model. It does not maintain map-reduce data reused and the information state during the run. Thus, map-reduce should read duplicate data and intermediate results of the study from disk in each repeat so that this operation has high costs such as accessibility to the disk, I/O, and unnecessary arithmetic operation. However, Hadoop is not suitable for return operation, although it is common in many applications (White et al, 2009). Spark is a cluster computing framework such as Hadoop (Zaharia et al, 2010) while it has been designed to meet the shortage of Hadoop in the iterative operations. Spark introduced data structure called flexible distributed data set by which re-used data and intermediate results can be saved in memory of

machines in the cluster during running iterative processes. It has been proven that this feature has effectively improved the performance of repetitive tasks requiring low latency (Zaharia et al, 2012).
In this paper, extensive tests have been done to evaluate and compare the efficiency of framework of spark and Hadoop. For this purpose, KNN machine learning algorithm was run on different data sets and runtime, memory usage, and CPU of framework were compared and evaluated .
The paper was organized as follows: In the second section, the review of the previous studies has been done. In the third section, Hadoop, Spark, and Ganglia systems were examined. In the fourth section, the implementation describing the laboratory context and used data set was done. In the fifth section, the obtained results were analyzed and evaluated. The conclusions are discussed in sixth section. Finally, a part of references used in this study has been provided.

2. Literatures

In 2013, Li Guo et al evaluated two framework of Hadoop and Spark in terms of memory usage and runtime. The assessment was done on several graph data sets through running Page Rank algorithm whose result indicated the Spark superiority in less implementation duration and more consumption ram compared to Hadoop (Gu and Li, 2013).
In 2014, Wang et al. evaluated map-reduce and Spark in terms of runtime. In this study, C4.5 algorithm used to produce decision tree was examined on data sets which were different in terms of volume. The results from this study was that the efficiency of Spark when the size of datasets is small is 950% better than the time when Hadoop is used. When the volume of dataset is large, the efficiency of Spark is 73% better than the time when Hadoop is used (Wang et al, 2014). There have been research works regarding the big data which prediction was implemented (Shamshirband et al., 2015; (Shamshirband et al., 2016).
In their study, Zaharia et al. (2010) compared framework of Hadoop and Spark on Logistic Regression algorithm whose result indicated the superiority of Spark (Zaharia et al, 2010).
In 2014, Hadoop, Spark, and Big Data MPI framework were examined by Liang et al. They compared them in terms of run-time, memory usage, and CPU. According to their results, MPI substrate was better than Hadoop substrate and Spark was also better than the Hadoop (Liang et al, 2014).

3. Basic concepts

3.1 MapReduce

Map-reduce model is a programming model for big datasets processing as distributed across multiple machines and was introduced by Google. According to Figure 1, in this model, big data was generally divided into smaller units using divide and conquer method and processed in parallel. Map-reduce is consisted of two main parts, namely, map and reduce. The map part receives the input data in key/value format and produces the output as pair middle key/value by processing on them. The reduce part receives the output of the map part and integrates the values with the same key together. At the runtime, the system automatically performs all tasks such as details of sharing input data, planning to run the programs on a set of machines, management, control of failure machines, and management of relations among machines (Dean and Ghemawat, 2008).

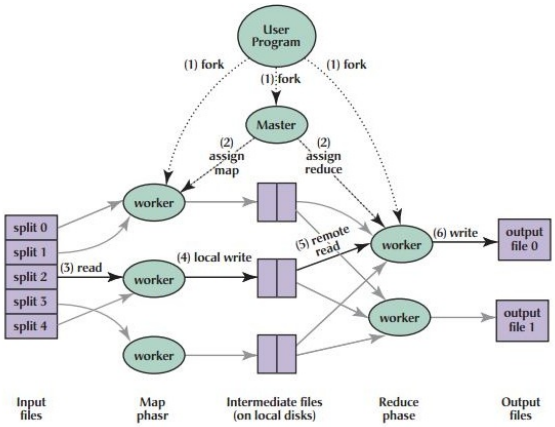

Figure 1: How do Map-reduce model (Dean and Ghemawat, 2008).

3.2 Hadoop
Hadoop is an open source framework written in Java programming language. Hadoop was created through inspiring by the studies published by Google in the field of distributed computations and file system specified to this Company called GFS and provided the possibility of distributed process on datasets distributed on the connected computers using a simple programming model. Hadoop was first created by someone named Doug Cutting to support distributing in Nutch search engine project. One of the considerable features of Hadoop includes its scalable ability from a server to thousands of machines with local ram and processing power. Hadoop with the ability to recognize and manage errors in the applied layer has a hardware-independent operation and thus, provides highly available services for users. Today, Hadoop is used in many provided commercial projects such as Yahoo, IBM, Oracle, and Microsoft (White et al, 2009).

3.3 Spark
Spark is an open source framework for big data processing designed for increasing the speed, ease of use, and complex processes. This model was created in a laboratory at UC Berkeley in 2009 and recognized an open source as one of the Apache projects in 2010. Many programs are return with repeating a series of operations such as return machine learning algorithms, interactive data analysis tools, graph algorithms, and many other uses. So, a framework called Spark covering these types of programs and with scalability and fault tolerance in map-reduce model was designed (Zaharia et al, 2010). Spark has introduced an abstraction called RDD resilient distributed datasets. RDD is a read-only set of objects which has been divided between a set of machines and if the division is eliminated, it is easily reconstructed. A user can cache a RDD on a machines' ram and repeat parallel operation like map- reduce several times. That's why Spark has a significant efficiency in algorithms that should do return process on a data set (Zaharia et al, 2012).

3.4 Ganglia
Ganglia was developed at UC Berkeley under the BSD license. The system is a robust solution without using resources to monitor the performance of clusters which include several thousand nodes. Ganglia can evaluate the performance of different components of a system such as CPU, ram, I/O, network traffic, and productivity of disk. Ganglia has been formed from two main services called Gmond and Gmetad. Gmond service is installed and run on each node in order to collect various parameters. Gmetad service integrates information collected by Gmond services, stores it in the database, and displays it to users by its web service (Massie et al, 2012).

4. Implementation
4.1 Cluster Architectures
Figure 2 shows the topology of the lab cluster in this study. Our lab cluster is composed of six computers one of which has been determined as master and the rest as slaves. Ubuntu 64-bit version 14.0.4 operating system has been used for all computers. All computer hardware specifications are the same and are as follows: 5 core Intel CPU Intel Core i5-4440 3.10GHz, 4GB ram, and 3.8GB available ram. All the six computers have been connected in a local network by a switch on the model (D Link DSE1016A) at a rate of (100Mbps).
On all systems, Java version 8 and 14.0.4 LTS Ubuntu operating system have been used. Spark version 2.0.1, Hadoop version 2.7.2, and Ganglia version 3.6.2 have been used for all tests. In all the programs, their stable versions have been used.

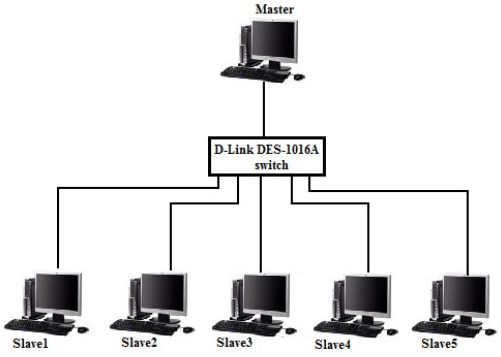

Figure 2: Cluster topology.

4.2 Data Set Description

In this study, different versions of Higgs data set have been used in a way that for a certain number of samples in Table 1, the samples were separated from the initiate of Higgs data sets and a new data sets was created. The dataset contains 11 million samples with 28 features measured by Monte Carlo simulator. The first 21 features are the movement properties of particles measured by acceleration particle simulators and 7 other features have been achieved by a function of the first 21 features. These top level features have been defined to distinguish between two available classes by physicists. In this study, in order to examine the data set with different volume and sizes, Higgs data set is divided into smaller data sets so that data set test includes500 samples at the end of Higgs data set (whitesonl, 2016).

Table 1: Data Sets

Name	Size	Number of Samples	Number of Attributes	Class Label
Higss	8Gb	11000000	28	2
Higss2	4Gb	5500000	28	2
Higss3	2Gb	2750000	28	2
Higss4	1Gb	1375000	28	2
Higss5	500Mb	687500	28	2
Higss6	250Mb	343750	28	2
Higss7	125Mb	171875	28	2
Higss8	63Mb	85937	28	2
Higss9	34Mb	43000	28	2
Higss10	16Mb	21000	28	2

4.3 Algorithm Description

In this section, one of the most important machine learning algorithms called KNN is addressed. KNN is one of the well-established algorithms with many applications in machine learning algorithms and data mining. Lack of training in this method and storing the input data have led the algorithm to be a lazy algorithm. Also, given that no default is considered in the input data, this method is a subset of non-parametric methods. We have a training data set whose samples are labeled as inputs. In this method, in order to classify a new unlabeled sample, its K nearest neighbor is first determined among from samples of the training set on the basis of used distance. Then, in terms of the majority vote of obtained K-nearest neighbor, the label of classification has been assigned to new data. In KNN method, only raw training data are used to determine the label of a new data without creating a model in the learning stage and thus, its implementation can be simply performed

5. Results and discussion

For each data set, KNN algorithm with k=5 was performed on Hadoop and Spark framework and then, run time was recorded for each data set. However, when KNN algorithm was running on both Hadoop and SPARC framework, using ganglia software, memory usage and productivity of processors were evaluated for lab cluster and recorded and saved.

For a fair comparison, K value was considered constant value and the convergent value of each data set was not used. Diagram 3 indicates the comparison of the run-time between Hadoop and Spark on data set with different sizes using which the advantage of Spark can be achieved compared to Hadoop at run time. Diagram 4 indicates memory usage of Hadoop and Spark which shows that memory usage in spark is much more than that in Hadoop.

Figure 5 shows the productivity of processors in the entire cluster which is better in Spark than Hadoop.

Given the structural features of Hadoop and Spark and the implementation process of KNN algorithm, the following results are found:

Like Higgs10, when the data set is small, Spark has a better performance than Hadoop so that the improvement of Spark's performance is 4.5 to 5 times than Hadoop.

On the other hand, by increasing the size of data set, like Hadoop, the advantage of Spark reduces compared to Hadoop. Hadoop is generally not suitable for small processes. Spark is run fast for processing such data set.

Spark's ability to maintain data in memory has caused Spark to be suitable for return algorithms. This leads to saving I/O time of intermediate results and this amount of time is a large part of wasting time in Hadoop.

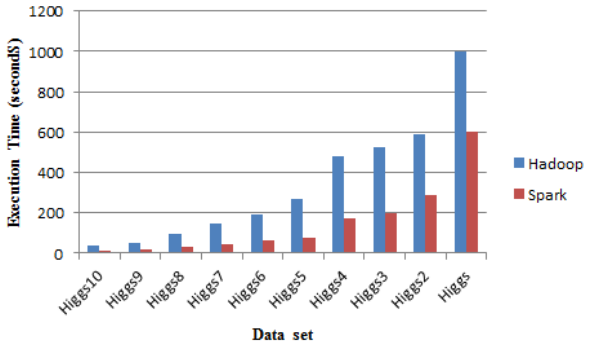

Figure 3: Execution time comparison of the cluster on a KNN algorithm.

Figure 4: Compare memory usage in the entire cluster on KNN algorithm.

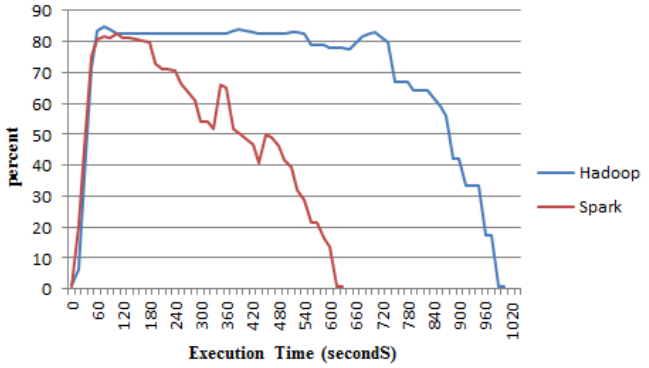

Figure 5: Compare CPU efficiency in the entire cluster on KNN algorithm.

5. Conclusions

In this study, an overview of both Spark and Hadoop frameworks was done and to evaluate their performance, different methods were compared using KNN algorithm. The results show the superiority of spark which is a very strong contender in the field of data processing using in-memory processes. In terms of memory usage, Hadoop consumes less ram compared to Spark and CPU is more suitable in Spark than Hadoop. Thus, our tests showed that although spark is generally faster than Hadoop, it significantly consumes in the memory usage. If speed is required

and there is no shortage in the amount of memory, spark is an appropriate choice; however, if there is enough disk space to store data and intermediate results and on the other hand, there is not enough memory, Hadoop can be a good alternative.

References

Chen, M., Mao, S., Liu, Y., Big data: A survey, *Mob. Netw. Appl.*, vol.19, pp.171–209, Apr. 2014.

Dean, J., Ghemawat, S., Mapreduce: Simplified data processing on large clusters, *Commun. ACM*, vol.51, pp.107–113, Jan. 2008.

Gu, L., Li, H., Memory or time: Performance evaluation for iterative operation on hadoop and spark, in *10th IEEE*, pp. 721-727, 2013.

Liang, F., Feng, C., Lu, X., Xu, Z., *Performance Benefits of DataMPI: A Case Study with BigDataBench*, pp.111–123, Cham: Springer International Publishing, 2014.

Massie, M., Li, B., Nicholes, B., Vuksan, V., Alexander, R., Buchbinder, J., Costa, F., Dean, A., Josephsen, D., Phaal, P., Pocock. D., *Monitoring with Ganglia*, O'Reilly Media, Inc, 1st ed., 2012.

Shamshirband, S., Mohammadi, K., Yee, L., Petković, D., Mostafaeipour, A., A comparative evaluation for identifying the suitability of extreme learning machine to predict horizontal global solar radiation, Renewable and sustainable energy reviews, vol. 51, pp. 1031-1042, 2015.

Shamshirband, S., Mohammadi, K., Tong, C.W., Petcock, D ., Porcu, E., Mostafaeipour, A., Ch, S., Sedaghat, A., Application of extreme learning machine for estimation of wind speed distribution, Climate Dynamics, vol. 46, no., (5-6), pp. 1893-1907, 2016. DOI 10.1007/s00382-015-2682-2

Wang, H., Wu, B., Yang, S., Wang, B., Liu, Y., Research of decision tree on yarn using mapreduce and spark, in *in Int. Conf. on Advances in Big Data Analytics*, (Las Vegas, USA), 2014.

White, T., *Hadoop: The Definitive Guide*, O'Reilly Media, Inc., 1st ed., 2009.

"whiteson, D., higgs data set," https://archive.ics.uci.edu/ml/datasets/HIGGS, 2014. Accessed: 2016.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., Stoica, I., Spark: Cluster computing with working sets, in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, (Berkeley, CA, USA), pp.10–10, USENIX Association, 2010.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., Mc- Cauly, M., Franklin, M. J., Shenker, S., Stoica, I., Resilient distributed datasets: A fault-tolerant abstraction for inmemory cluster computing, in *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, (San Jose, CA), pp.15–28, USENIX, 2012.

Mohammad Ahmadi is the M.S. student of Computer Engineering Department from Yazd University of Iran.

Amir Jahangard Rafsanjani received the B.Sc. degree in Software Engineering from Shahid Beheshti University, Tehran, Iran in 2003, and the M.Sc. (2006) and Ph.D. (2014) degree also in Software Engineering from Sharif University of Technology, Tehran, Iran. He joined Yazd University in 2014, and is currently an Assistant Professor in the Department of Computer Engineering. His current research interests include model driven development, semi-structured data, internet of things, big data and database security.

Ali Mostafaeipour is an assistant professor of Industrial Engineering at Yazd University, Iran. He has been teaching at Yazd University since 1989. He studied at Winona State University (University of Minnesota) in state of Minnesota, USA; University of Wisconsin at Platteville, Wisconsin, USA; Alabama A&M, Alabama, USA; and Iran University of Science and Technology, Tehran, Iran. He has served as a committee member, guest speaker, and co-chairman of 145 international conferences. He has been reviewer of 17 international journals mainly Elsevier. He has presented 78 mostly International conferences throughout the world. He has undertaken and managed 18 research projects, and holds 3 patents. He has been editorial board of several professional journals. Finally, he has published 54 journal articles mostly at Elsevier (ISI), and he authored 4 books. He holds an award for excellence from Yazd University as the year 2013 distinguished researcher, also distinguished author of "Wind Energy" book (INTech publisher, 2012, Croatia) with more than 5000 downloads in six months. His research interest lies in renewable energies, wind energy, value engineering, economic evaluation, and feasibility study of project.

.