Estimate the Performance of Cloudera Decision Support Queries

https://doi.org/10.3991/ijoe.v18i01.27877

Tahani M. Allam
Faculty of Engineering, Tanta University, Tanta, Egypt
tahany@f-eng.tanta.edu.eg

Abstract—Hive and Impala queries are used to process a big amount of data. The overwriting amount of information requires an efficient data processing system. When we deal with a long-term batch query and analysis Hive will be more suitable for this query. Impala is the most powerful system suitable for real-time interactive Structured Query Language (SQL) query which are added a massive parallel processing to Hadoop distributed cluster. The data growth makes a problem with SQL Cluster because the execution processing time is increased. In this paper, a comparison is demonstrated between the performance time of Hive, Impala and SQL on two different data models with different queries chosen to test the performance. The results demonstrate that Impala outperforms Hive and SQL cluster when it comes to analyze data and processing tasks. Using two benchmark datasets, TPC-H and statistical computing, we compare the performance of Hive, Impala, and SQL clusters 2009 Statistical Graphics Data Expo.

Keywords—hadoop, impala, hive, massive parallel processing, big data, TPC-H, graphics data expo 2009

1 Introduction

A massive quantity of data is produced daily from every part of the world [1]. This Inflation in the data comes from the advances of technology like the arise of cloud computing, internet of things and smart and sending devices which exist nowadays in [2]. The source of data is heterogenous and diversified. It could be means of communication like social media sites, sensor networks, health care reports, security camera, hospitals, governments and so on [3]. Big data is used to define large amount of complex data that can't be processed or analyzed by traditional means [4].

One of the many techniques to analyze a big amount of data is Hadoop framework [5-7]. Hadoop is an Apache framework that can be used to process big data [8]. Not only Hadoop can analyze data but also is a distributed storage for big data as well relying on distributed clusters of commodity machines [9]. An easy way to query and analyze data in Hadoop is by using Hive or Impala. In the next two sections we are explained the Hive and Impala frameworks.

By integrating big data, studies try to enhance the platform's usage as a platform for processing such large data [10]. To address the issue, Apache Hadoop is the best option [11]. However, for complicated businesses, relying on only one platform is insufficient. Big data processing with sophisticated queries, such as Cloudera [12] and Hortonworks [13], will impact execution time if using the data Platform, which is a solution that unifies the features and capabilities of several apps and utilities. Both platforms offer a variety of query processing scenarios for cutting down on execution time.

The rest of this article is organized as follows: Section 2 explains the Apache Hive Architecture. Section 3 briefly discusses the Cloudera Impala framework. Section 4 depicts the proposed frameworks. In Section 5 shows the experiment and result. Section 6 is a conclusion.

2 Related work

Hive is a data warehouse which offers HiveQL (HQL) language which is similar to SQL for easy query data [14]. Apache Hive supports different file formats such as text files, Avro, sequence file…etc. Also, it supports various programming languages like java, python which means Hive client applications can be implemented with many languages [15]. Figure 1 shows the Hive Architecture.

Hive services consist of three main components which are Compiler, Driver and Metastore. When a SQL command is written for execution, it gets parsed by the help of the compiler to check syntax and convert it into MapReduce input. Processing and resources managements can be MapReduce or YARN applications. The Metadata generated from executing Hive queries is storied in Metastore by the driver. The generated Metadata contains data for tables such as its schema and location [16]. Hadoop distributed file systems represent a distributed storage.

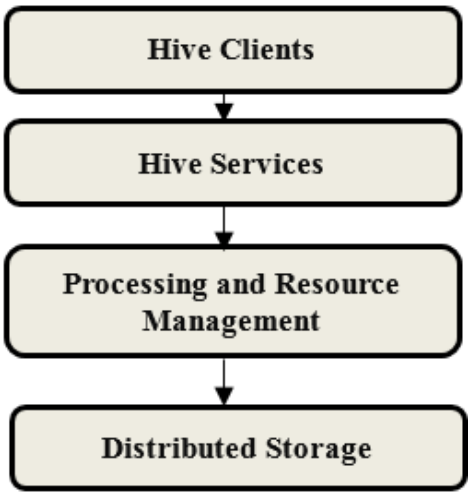

Fig. 1. The apache Hive architecture

3 Cloudera IMPALA

On the other hand, Apache Impala is considered the newest engine integrated into the Hadoop Distributed System [16]. Unlike Hive, Impala doesn't rely on MapReduce architecture, instead, Impala relied on a massively parallel process (MPP) architecture, so it performs faster than Hive [17]. Thus, Impala is considered a real time query engine which uses Hive metastore of the table information which already exists. Analysts prefer impala to perform real time analytics with lower cost, complexity and high speed [18].

Impala's architecture consists of three daemons which are catalog, statestored and impalad. Impalad deamons execute on every node of the distributed cluster as it considered as the main part of the Impala. Impalad also accepts queries and distribute it to other nodes in the cluster. While, Statestored deamon keeps track of the state of the nodes to inform impalad deamons of the healthy nodes which will accept and execute the queries. Finally, the catalogue deamon transfers the changes done to the metadata to all the nodes in the cluster [16] [19]. Figure 2 depicts how the Impala architecture looks from the front.

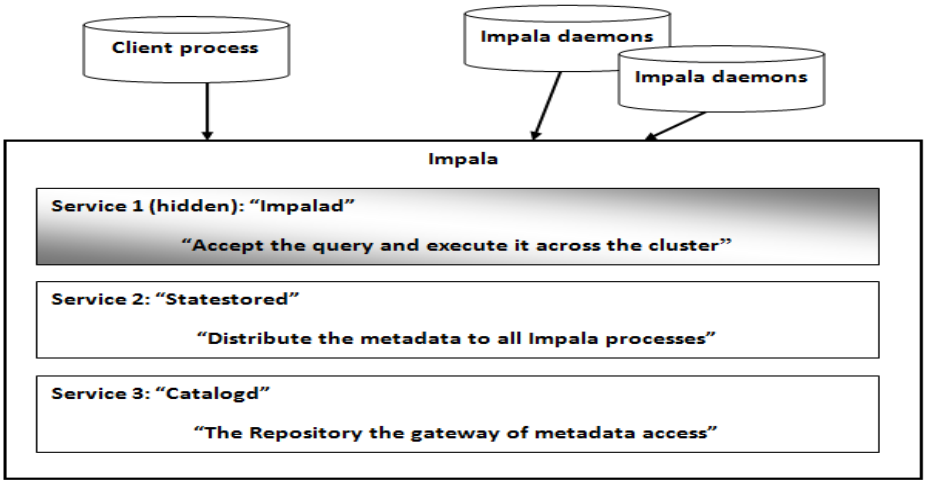

Fig. 2. The architecture of Impala

Even though Impala is based on Hadoop, it doesn't use it. All your nodes have daemons running that cache some of the HDFS data. The MapReduce technique does not need to be used because these Impala daemons can swiftly deliver data. Impala is not a replacement for Hive; rather, it excels in situations where Hive fails. Data scientists and business analysts who only want to look at and study some data without constructing comprehensive workflows may find the Impala to be an excellent choice. As a side note, Impala isn't all that mature. When the amount of data is greater than the available memory, it can crash. Figure 3 illustrates the query processing procedure in Impala [19] using Figure 3.

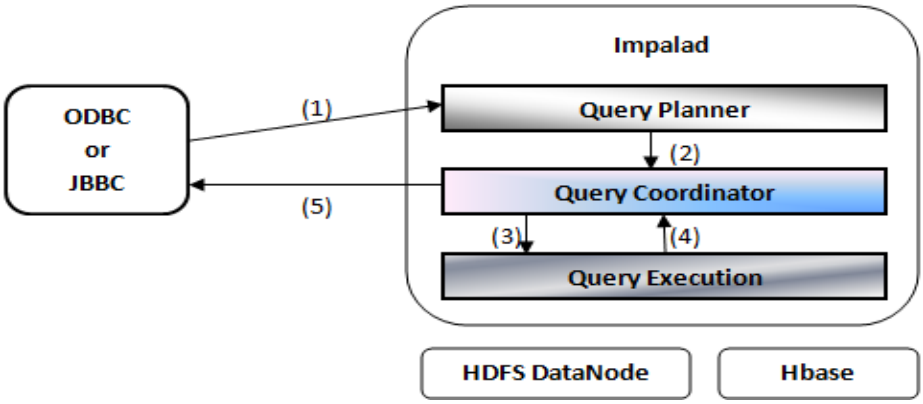

Fig. 3. The Impala query processing

4 Proposed frameworks

We have two different frameworks: Hive and Impala on Hadoop. There are two different datasets are used: the TPC-H dataset and Statistical Graphics Data expo '09 dataset. Performance parameters including data set file size, query statements, and query average time all have an impact on the final output.

4.1 Experimental result of TPC-H dataset

TPC-H dataset contains 22 queries of decision support queries from Q1 to Q22 which asses the performance of different decision support systems by examining large volumes of data and executing complex queries [20]. We examine only the queries which contains multiple joins between tables to show the difference between each framework. Queries "Q1, Q3, Q5, Q7, Q8, Q9, Q10, Q11, Q12, Q18, Q21" are shown the joint between tables. Each TPC-H query asks a business question and includes the corresponding query to answer the question. We make a comparison between Hadoop and SQL which discussed in TPC BenchmarkTM H Standard Specification Revision 3.0.0. The comparison depends on the execution time of each query in different approaches.

Experiments are carried out using Cloudera VM on Oracle VM VirtualBox. The queries were performed on Hue. Four processors were used with 2 cores per each processor and 16GB memory. According to TPC-H Rev. 2.18.0, the total database size is 1000 GB. We applied the same database and the same queries to explain the execution time for each query on Hadoop frameworks and SQL.

4.2 Experimental result of data expo '09 dataset

Statistical analysis and modeling graphs that display statistics data originally created from RITA, the Data Expo 2009 dataset which contains large records of flight

departure and arrival that could reach more than 120 million records [21]. Each data record contains 23 attributes and 70-80 lakh rows. The data size is up to 1.6 gigabytes when it is compressed and 12 gigabytes when it is uncompressed. This data is providing the important features of the dataset graphically. For each dataset, we have a year's worth of airline data saved in the D1 dataset, and 2- and 3-years' worth of airline data in the D2, D3 datasets [22]. We make a comparison with a previous work but on a different experiments' environment. Experiments are carried out using Cloudera VM on Oracle VM VirtualBox. The queries were performed on Hue. Four processors were used with 2 cores per each processor and 16 GB memory.

5 Results and discussion

We have two different frameworks: Hive and Impala on Hadoop. There are two different datasets are used: the TPC-H dataset and Statistical Graphics Data expo '09 dataset. Performance parameters including data set file size, query statements, and query average time all have an impact on the final output.

5.1 TPC-H dataset results

The performance of Hadoop frameworks "Hive and Impala" were tested by two benchmark databases named TPC-H and Graphics Data expo '09. In TPC-H Dataset, we tested the queries which contained the most joins in specific query to test the performance of Hive and Impala when multiple tables exist compared with SQL query. The database load time is 02:34:12 on SQL framework [20]. In Hadoop frameworks, the database sored on HDFS directly which save this time in the execution phase.

As shown in Table 1, Impala showed lower execution time while using TPC-H dataset. Hence, joining tables from multiple nodes in the cluster didn't affect the performance of Impala unlike Hive like shown in figure 4. In SQL and Hive queries, there are a big difference in the execution time as compared with Impala framework.

Table 1. Execution time in sec. between Hive, Impala and SQL using TPC-H database

Query	HIVE	IMPALA	SQL [17]
Q1	219.40	17.71	270.00
Q2	798.40	22.55	30.00
Q3	268.40	22.72	50.00
Q5	425.20	21.36	70.00
Q7	545.60	20.16	60.00
Q8	148.20	23.01	65.00
Q9	861.00	27.00	266.00
Q10	302.20	11.50	40.00
Q18	405.20	29.80	460.00
Q21	672.60	12.50	320.00

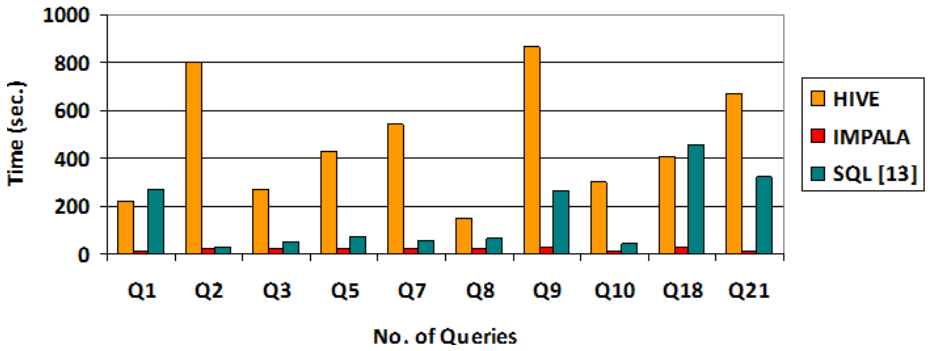

Fig. 4. Execution time (sec.) between Hive, Impala and SQL frameworks using TPC-H Dataset

5.2 Statistical graphics data expo '09 dataset results

While in the other dataset, performance is measured regarding the size of data in one table and the efficiency of Hive and Impala is tested on table containing the data of one-year D1, while D2 dataset contained the data of two years and D3 dataset contained data of three accumulated years. As stated in [22], size of the data being processed affects the performance and the query execution time, so the data of the statistical computing statistical graphics dataset is divided into three datasets. We test the query in the Hive and Impala three periods and takes these three outcomes, on average. Table 2 and figure 5 shown the mean result in a Hive query on D1datasets and the compared results. Data processing is scalable and straight forward in Hive cluster. Our Hive cluster has a less execution time than a compared cluster [22].

Table 2. The mean execution time (sec.) in Hive query on dataset D1

Query	HIVE	Compared Hive [19]
Q1	60.21	69.28
Q2	41.30	46
Q3	38.56	48.28
Q4	42.10	45.57
Q5	26.28	33.76
Q6	18.09	23.32
Q7	19.20	21.24
Q8	31.55	38.19
Q9	23.63	27.86
Q10	27.37	31.90

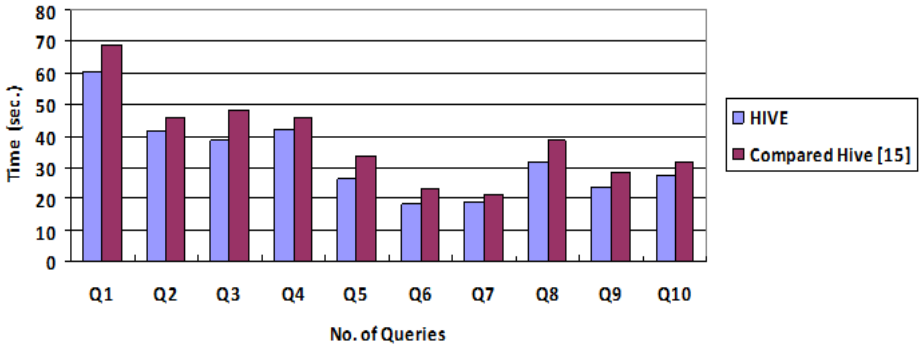

Fig. 5. The mean execution time (sec.) in Hive Query on dataset D1

Table 3 and Figure 6 shown the mean result in Impala cluster on datasets D1 and the compared Impala results [19]. Impala cluster on Hadoop has a less execution time than Hive cluster.

Table 3. The mean execution time (sec.) in Impala query on dataset D1

Query	Impala	Compared Impala [19]
Q1	12.22	16.28
Q2	10.41	14.06
Q3	5.56	10.47
Q4	4.88	9.77
Q5	7.25	10.43
Q6	6.36	9.72
Q7	6.21	9.43
Q8	5.66	9.00
Q9	6.10	9.66
Q10	4.89	9.58

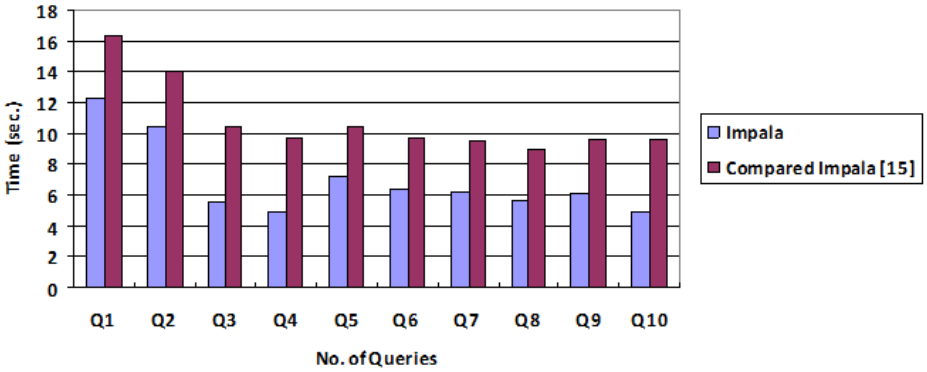

Fig. 6. The mean execution time (sec.) in Impala query on dataset D1

Table 4 and Figure 7 shown the mean result in a Hive query on D2 datasets and the compared results.

Table 4. The mean execution time (sec.) in Hive query on dataset D2

Query	HIVE	Compared Hive [19]
Q1	79.20	87.84
Q2	48.32	52.42
Q3	50.19	56.38
Q4	61.42	65.67
Q5	70.56	73.86
Q6	31.27	41.04
Q7	32.10	40.43
Q8	40.00	44.11
Q9	45.65	56.30
Q10	38.55	44.88

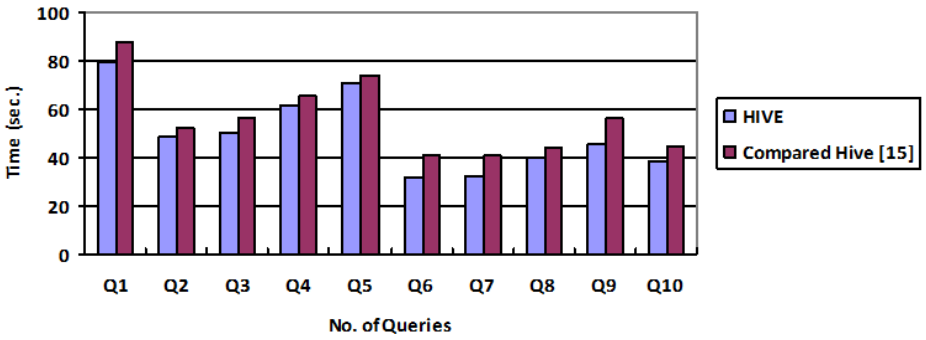

Fig. 7. The mean execution time (sec.) in Hive query on dataset D2

Table 5 and Figure 8 shown the mean result in Impala cluster on datasets D2 and the compared Impala results [22]. Impala cluster on Hadoop has a less execution time than Hive cluster.

Table 5. The mean execution time (sec.) in Impala query on dataset D2

Query	Impala	Compared Impala [19]
Q1	29.03	32.06
Q2	22.84	29.91
Q3	23.02	31.08
Q4	25.56	30.90
Q5	25.66	30.98
Q6	28.24	32.02
Q7	26.12	30.01
Q8	26.30	30.50
Q9	26.00	30.00
Q10	27.73	31.90

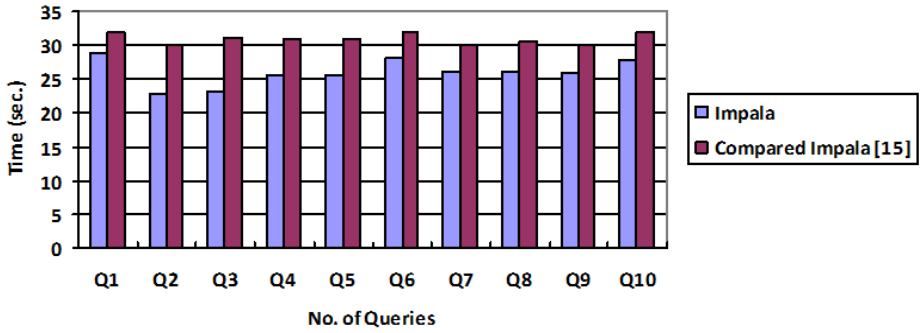

Fig. 8. The execution time (sec.) in Impala query on dataset D2

Lastly, a bigger dataset was made containing three years of data named D3. It tested on both Hive and Impala and compared with previous work [22-24]. Query mean time was higher than the last two experiments, but Impala's MPP architecture proved its efficiency as shown in Table 6, 7 and Figures 9, 10. Even though time increased but Impala still has better execution time than Hive as shown below.

Table 6. The mean execution time (sec.) in Hive query on dataset D3

Query	HIVE	Compared Hive [19]
Q1	98.30	111.27
Q2	100.25	116.53
Q3	91.60	108.67
Q4	92.22	108.32
Q5	110.40	123.15
Q6	95.35	107.48
Q7	89.50	99.01
Q8	41.58	50.63
Q9	62.56	73.68
Q10	61.77	72.98

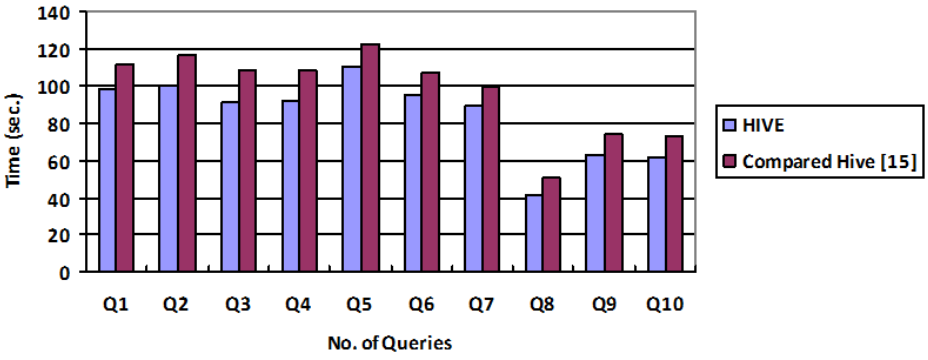

Fig. 9. The mean execution time (sec.) in Hive query on dataset D3

Table 7. The mean execution time (sec.) in Impala query on dataset D3

Query	Impala	Compared Impala [19]
Q1	40.20	48.65
Q2	38.88	46.73
Q3	38.20	46.33
Q4	41.43	48.73
Q5	37.36	45.85
Q6	39.02	46.15
Q7	39.25	46.56
Q8	40.56	47.65
Q9	40.32	47.42
Q10	35.52	44.85

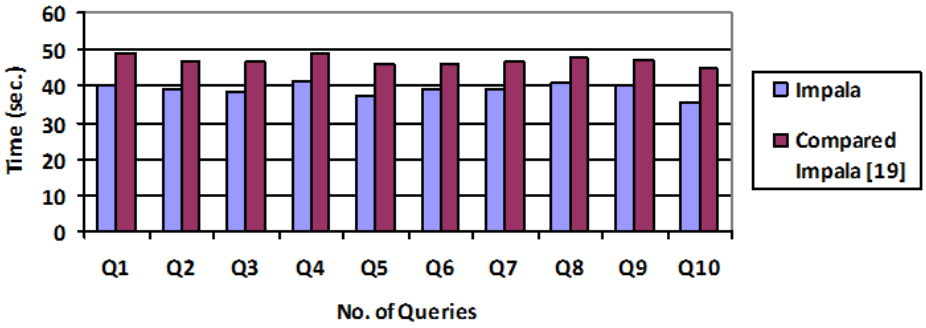

Fig. 10. The mean execution time (sec.) in Impala query on dataset D3

6 Conclusion

Even though Impala is the newest addition to the Hadoop distributed system, but it proved high efficiency in a big data. Two datasets were used in this paper to evaluate the outcomes of Hive and Impala when the size of data being processing is increased or when joining multiple tables which exists in multiple nodes occurs. According to experiments, Impala showed better performance and efficiency when it comes to query completion time due to relying on MPP which is designed to handle complex queries. Being a real time query system will open the doors to different applications which will need fast response and decisions depending on millions of data. In Hive, A good designed tables and query are improved the execution time which reduced the processing cost. Hive used MapReduce in parallel processing to execute one program while Impala provides MPP. Impala is ideal for real time queries, but it is not ideal for heavy joins data.

7 References

[1] Ma, J., Chen, L., Lv, M., Yang, Y., Zhao, Y., Wu, Y. and Wang, J. (2017). Logical query optimization for cloudera impala system. Journal of Systems and Software. 1(125): 35-46. https://doi.org/10.1016/j.jss.2016.11.038

[2] Paul, D. C., Neeraj, K. S. and Peter, T. Cloud Computing. In Essential Computer Science, Apress, Berkeley, First Online: 12 June 2021, pp 195-224.

[3] Oussous, A., Benjelloun, F.Z., Lahcen, A.A. and Belfkih, S. (2018). Big Data technologies: A survey. Journal of King Saud University-Computer and Information Sciences. 30(4):431-48. https://doi.org/10.1016/j.jksuci.2017.06.001

[4] Mahammad S. and Venkatesh Sharma. K. (2019). A Study on Big Data Advancement and Big Data Analytics. JASC: Journal of Applied Science and Computations, 6(1): 4099-4109.

[5] Mourlin, F. and Farinone, J.M., (2019). Cloud Mobile Storage for Mobile Applications. International Journal of Interactive Mobile Technologies (iJIM), 13(3): 13-28. https://doi.org/10.3991/ijim.v13i03.8086

[6] El-Seoud, S.A., El-Sofany, H.F., Abdelfattah, M. and Mohamed, R. (2017). Big Data and Cloud Computing: Trends and Challenges. International Journal of Interactive Mobile Technologies. International Journal of Interactive Mobile Technologies (iJIM), 11(2): 34-52. https://doi.org/10.3991/ijim.v11i2.6561

[7] Ouatik, F., Erritali, M., Ouatik, F. and Jourhmane, M. (2021). Students' Orientation Using Machine Learning and Big Data. International Journal of Online and Biomedical Engineering (iJOE), 17(1): 111-119. https://doi.org/10.3991/ijoe.v17i01.18037

[8] Amini, S., Gerostathopoulos, I. and Prehofer, C. Big data analytics architecture for real-time traffic control. 5th IEEE international conference on models and technologies for intelligent transportation systems (MT-ITS), IEEE, 2017 Jun 26, pp. 710-715. https://doi.org/10.1109/mtits.2017.8005605

[9] Huang, W., Wang, H., Zhang, Y. and Zhang, S. A novel cluster computing technique based on signal clustering and analytic hierarchy model using hadoop. Cluster Computing. 2019 Nov; 22(6):13077-84. https://doi.org/10.1007/s10586-017-1205-9

[10] Faridh, M. R., Ibnu, A., Sidik, P. and Fajar, A. Performance evaluation sql-on-hadoop: a case study of Hortonworks and Cloudera. 2nd International Conference on Data and Information Science, 2019. https://doi.org/10.1088/1742-6596/1192/1/012016

[11] Apache Hadoop. HDFS Architecture Guide" [Online], available on: https://hadoop.apache.org/docs/. (accessed on October 14, 2021).

[12] Cloudera. Machine Learning, Analytics, Cloud - Cloudera [Online], available on: https://www.cloudera.com/. (accessed on October 14.2021).

[13] Hortonworks. Data Management Platform [Online], available on: https://hortonworks.com/. (accessed on October 14, 2021).

[14] Capriolo, E., Wampler, D. and Rutherglen, J. Programming Hive: Data warehouse and query language for Hadoop. O'Reilly Media, Inc., 2012 Sep 19.

[15] Wang, H., Niu, D. and Li, B. Dynamic and Decentralized Global Analytics via Machine Learning. In Proceedings of the ACM Symposium on Cloud Computing 2018 Oct 11, pp. 14-25. https://doi.org/10.1145/3267809.3267812

[16] Chang, B.R., Tsai, H.F. and Lee, Y.D. (2018). Integrated High-Performance Platform for Fast Query Response in Big Data with Hive, Impala, and SparkSQL. A Performance Evaluation. Applied Sciences. 8(9):1514. https://doi.org/10.3390/app8091514

[17] Chen, L., Zhao, Y., Yang, Y., Lv, M., Chen, D., Wu, Y. and Wang, J. (2018). A query execution scheduling scheme for Impala system. Concurrency and Computation, Practice and Experience. 30(8): e4392. https://doi.org/10.1002/cpe.4392
[18] Russell, J. Cloudera Impala. O'Reilly Media, Inc., 2013 Nov 25.
[19] Kornacker, M., Behm, A., Bittorf, V., Bobrovytsky, T., Ching, C., Choi, A., Erickson, J., Grund, M., Hecht, D. (2015). Jacobs M and Joshi I. Impala: A Modern, Open-Source SQL Engine for Hadoop. In Cidr Journal, 1: 1-34. https://doi.org/10.1007/978-3-658-11589-0_8
[20] Transaction Processing Performance Council. (2017). TPC-HBenchmark Specification.
[21] Data Expo 2009 - Airline on-time performance, [Online], available on: http://statcomputing.org/dataexpo/2009/. (accessed on October 14, 2021).
[22] Saeed, S., Shaikh, A., & Naqvi, S. M. R. (2018). Assurance using two erp: Microsoft dynamics ax and sap due to the usage of supply chain management module. Mehran University Research Journal of Engineering & Technology, 37(2), 337-350. https://doi.org/10.22581/muet1982.1802.10
[23] Tummalapalli, S. and Machavarapu, V. R. (2016). Managing mysql cluster data using cloudera impala. Procedia Computer Science, 85(5):463-74. https://doi.org/10.1016/j.procs.2016.05.193
[24] Naveed, Q. N., Qureshi, M. R. N., Tairan, N., Mohammad, A., Shaikh, A., Alsayed, A. O., ... & Alotaibi, F. M. (2020). Evaluating critical success factors in implementing E-learning system using multi-criteria decision-making. Plos one, 15(5), e0231465. https://doi.org/10.1371/journal.pone.0231465

8 Author

Tahani M. Allam is a lecturer at the Computers and Control Engineering Department, Faculty of Engineering, Tanta University, Egypt, from 2018 till now. In addition, Tahani works as an assistance Lecturer at the Computers and Control Engineering Department, Faculty of Engineering, Tanta University-Egypt from 2010 till 2018. She was born in Kuwait, her B.Sc., M.Sc., and Ph.D. degrees were taken from the Department of Computers and Control Engineering, Faculty of Engineering, Tanta University in 2002, 2010, and 2018, respectively. Tahani works as a Consultant Engineer, then a General Supervisor of Human Resources Management Program on Management Information Systems (MIS) Project- Tanta University since 2010 till now. She is the Director of the Job Performance Evaluation Unit in Tanta University since 2021. Her research interests include cloud computing, Semantic Web, machine learning, security, and the Internet of Things. Tahani has published about 6 articles in various refereed international journals and conferences.

Article submitted 2021-10-26. Resubmitted 2021-12-05. Final acceptance 2021-12-07. Final version published as submitted by the author.