Indonesian Journal of Electrical Engineering and Computer Science
Vol. 31, No. 2, August 2023, pp. 945~956
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v31.i2.pp945-956

Predictive analytics on COVID-19 data using Hive based on Hadoop cluster

Ali Abbood Khaleel, Ali Noori Kareem, Laith Hikmet Mahdi
Department of Computer Engineering Techniques, Bilad Alrafidain University College, Diyala, Iraq

Article Info

Article history:

Received Jan 3, 2023
Revised Mar 7, 2023
Accepted Mar 12, 2023

Keywords:

Big data
COVID-19
Hadoop framework
Hive
MapReduce

ABSTRACT

COVID-19 pandemic has received a serious attention from academia, industry and governments to stop the huge number of deaths and economic disruptions around the world. Many techniques have been used to control the spread of the pandemic by understanding its characteristics and behavior. However, because of the large amounts and complex characteristics of COVID-19 data, the querying and analysis of such data using conventional tools have become a challenging task. As a result, powerful and distributed tools are highly required for querying and analyzing this data effectively. In this paper, distributed system using Hive based on Hadoop cluster is used to query and analyze COVID-19 data to obtain meaningful information. Hadoop is employed as a scalable and reliable framework to accommodate such large amounts of data. Hive is used as a data warehouse that run on Hadoop cluster to perform querying and predictive analytics on huge COVID-19 datasets. Several experiments are performed to evaluate the performance of proposed system. Experiments show that the proposed system outperforms relational database management system (RDBMS) in terms of query processing time. Experiments also show that the proposed system has a better efficiency in terms of data load, I/O operation, reading and writing data.

Corresponding Author:

Ali Abbood Khaleel
Department of Computer Engineering Techniques, Bilad Alrafidain University College
Diyala, 32001, Iraq
Email: alikh861@gmail.com, draliak@bauc14.edu.iq

1. INTRODUCTION

The wide usage of smartphones, tablets, sensors and wearable devices led to arise the internet of things (IoT) technology which has a considerable effect on data producing. Furthermore, data can be produced by different sources like education, banking, retails, transportation, healthcare and many more. Healthcare system can generate enormous amounts of data and such data can be complex because of its charachteristics such as, variety, volume, velocity and veracity. One of the most produced data in the healthcare system is COVID-19 data due to the high percentage of daily cases around the world. The analysis and query of this data is very important to predict outcomes and gain insights that help physicians to take required decisions. However, processing such data using conventional tools faces some challengings and consume considerable time. As a result, many orgnizations and companies have adopted some durable solutions. Hadoop is one of the most important solutions to deal with large and complex data due to its brilliant features. Hadoop framework is mainly designed to process and store massive unstructured data [1]. Although Hadoop is specifically designed to manage unstructured data, it is also used to handle structured and semi-structured data using some important software tools. One of the most effective tools for quering and analysing structured and unstructured data is Hive [2]. Hive is distributed data warehouse software that developed to be run on Hadoop cluster and designed to provide query and analysis on large datasets. COVID-19 data can be classified as a big data beacuase it is generated in large quantities at a high pace. As aforementioned, the processing and analysis of such data using traditional tools like relational database management is a challenging task beacuase it requires high capabilities of computing power. To cope with this issue, Hadoop MapReduce framework is employed. Hadoop MapReduce is a powerful platform that distributes the storage and processing of large datasets due to its reliability, fault tolerance and scalablity. Hadoop is an open source, which is provided by Apache to support the parallel processing of large datasets using MapReduce algorithm [3]. Furthermore, massive datasets are distributed on multiple Hadoop machines for better utilization of computing resources in trems of storage, hard disk I/O and processing. Different approaches and techniques have also been developed to analyse, query and process COVID-19 data in effective way. In this paper, predictive analytics for COVID-19 cases using Hive and Hadoop has been proposed. The contribution of this paper is shown below: i) Build a distributed system based on Hive and Hadoop to perform predictive analytics on COVID-19 data to gain meaningful information. ii) Tuning and enabling the most important parameters of Hive to improve its performence and reduce the execution time of COVID-19 queries.

The rest of paper is presented as follows. Section 2, presents some of the research and related work. Section 3 provides overview of Hive and Hadoop framework. While in section 4, the proposed system is explained. Results and discussions are presented in section 5 and finally, the paper is concluded in section 6.

2. RELATED WORK

Many research and studies have been proposed to analyse, manage, predict and prevent COVID-19 cases. One of them is Fbprophet Model that was proposed as a prediction tool for coronavirus detection and deaths [4]. The proposed work took into consideration two affected countries which are India and Japan in their approach. They perform prediction on total cases number, new cases, the number of total deaths and new deaths. Alsunaidi *et al.* [5], several studies regrding COVID-19-based big data analtics have been discussed. These studies are used as a taxonomy to control and manage the pandemic. Furthermore, several challenges of COVID-19 data analysis were also discussed. AutoRegressive integrated moving average (ARIMA) and Prophet time series forecasting models were employed by [6] to predict COVID-19 in most affected coutries such as, US, Italy, France, Russia, Germany, Spain, India, Turkey, and United Kingdom. Their results show that ARIMA model is more effective for COVID-19 prediction. Mahalle *et al.* [7], a study on different analytics techniques and various models for COVID-19 pandemic were discussed. The study show that Prophet algorithm is faster than other models for COVID-19 prediction. Temporal data science algorithm in [8] is presented to analyze big COVID-19 data using ubiquitous computing. Gupta *et al.* [9] detects COVID-19 cases including confirmed, deaths and cured cases in India. They use different models such as, random forest, linear model, support vector machine, decision tree, and neural network. Random forest is employed for prediction and analysis of all the results as it outperforms other models. They perform K-fold cross-validation to measure model consistency. A new federated learning (FL) solution is proposed to allow local generative adversarial networks (GANs) to collaborate and exchange learned parameters with a cloud server [10]. The proposed solution aims to improve the global GAN model for producing real images of COVID-19 instead of sharing actual data.

Oruche *et al.* [11] proposes KnowCOVID-19 as an “evidence-based” recommender system. The proposed system uses edge computing service to integrate recommender modules for data analytics using end-user thin-clients. A large-scale study based on data mined from Twitter is presented in Mourad *et al.* [12]. The authors have performed substantial analysis on around one million COVID-19 related tweets gathered during a period of two months. The results provided potential solutions and social networks management strategies during crisis periods. Senti-COVID19 is presented as an interactive visual analytic system to effectively analyze sentiment from social media text [13]. It allows users to explore extracted data and discover insights from the gathered tweets. Dash *et al.* [14], intelligent computing model is developed to predict the outbreak of COVID-19. Autoregressive integrated moving average (ARIMA) models are employed for forecasting the daily-confirmed cases for around 90 days future values of six most affected countries in the world and six Indian states that have the high incidence of infection. Health monitoring framework is presented by [15] to analyse and predict COVID-19. The proposed framework integrates both of big data analytics and IoT technology. Descriptive, diagnostic, predictive, and prescriptive analyses have been performed based on big data analytics. They focus on different pandemic symptoms by employing a novel disease real data set. A data science solution based on data ming and visualization is presented to obtain useful information and knowledge from COVID-19 data [16]. Another data science solution for healthcare system is presented to perform analysis on COVID-19 data. The proposed system predicts and classifies the cases of COVID-19 using real-life COVID-19 data including routine blood test results from Brazilian COVID-19 patients [17].

3. OVERVIEW OF HADOOP FRAMEWORK AND ITS ECOSYSTEM

Hadoop is a framework that facilitates the distributed storage and parallel processing through a cluster of computing nodes [18], [19]. Hadoop can be divided into one namenode and many datanodes. It supports master-slave architecture, the namenode is considered as a master while the other datanodes are working as slaves. The master node contains the information related to metadata and file system namespace, while the data nodes store the actual data and execute map and reduce tasks [20], [21]. Hadoop is widely used by many organizations and companies such as social media and Youtube due to its cost effectivity. Hadoop has some prominent features such as extensibility and availability. Hadoop can adapt the continuous increasing of the data size by using large number of nodes. It can be extended from one single node to hundreds of nodes to utilize the computing resources for processing and analysing large datasets effectively. It can store enormous unstructured data using Hadoop distributed file system (HDFS) [22]. As mentioned before, Hadoop is a fault tolerant tool, thus any hardware and software failure can be solved efficiently and therefore avoid the stop of the cluster. Hadoop uses MapReduce programming model to process and store massive data in a distributed manner through a cluster of servers and commodity devices [23].

3.1. Hadoop architecture

The main architecture of Hadoop framework includes MapReduce and HDFS [18]. Hadoop is illustrrated in Figure 1. The master/slave architecture of Hadoop is clearly seen. It consists of one node called masternode and multiple nodes called slaves. The duty of master one is to manage and monitor the progress of map and reduce tasks through job tracker. While the slave ones are used to execute both of map and reduce tasks by task trackers.

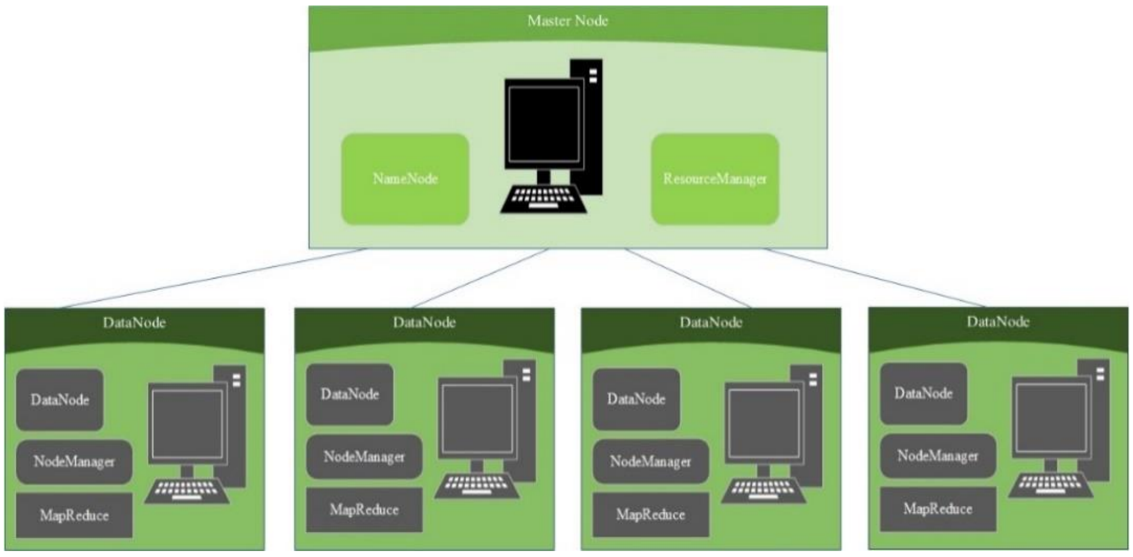


Figure 1. The architecture of Hadoop framework

3.2. Hadoop distributed file system (HDFS)

HDFS is a distributed file system, which built in java as an efficient storage for massive large datasets over a cluster of computing machines. It is designed to be similar to google file system (GFS), where the data is stored across several nodes [18]. HDFS can provide high throughput access to data applications because of its availability and extensibility [24]. It supports batch processing of job execution in Hadoop cluster. All input data in HDFS are broken into several blocks and multiple copies of them are created and distributed across several datanodes. Each data block is assigned with a specific size by Hadoop user during the configuration of the Hadoop cluster. Similarly, the replication factor is also can be configured by the user through the settings of Hadoop cluster.

Data block size is the same and it is configured by default as 64 MB, while the replication factor is 3. As aforementioned, each data block has multiple copies and each copy is stored in a different datanode to avoid the data losing in case of hardware failure or software crash. In HDFS, the file system namespace is managed by namenode. Moreover, creating the files and directories and some file operations are also performed by the namenode in the master node. Secondary namenode is available as assistant of the name node to obtain a better performance. The namenode sends the instructions to datanodes that store the data on the salve nodes. Data blocks on the slave machines are monitored by the datanodes in terms of creation, deletion and replication upon receiving the instructions from the namenode [24]. The Hadoop distributed file system is illustrated in Figure 2.

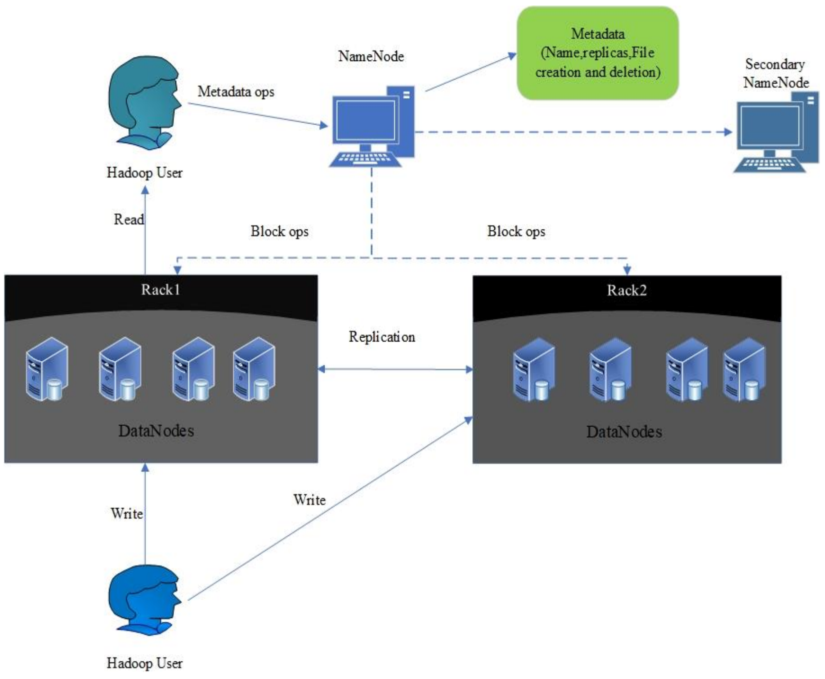


Figure 2. The architecture of HDFS

3.3. MapReduce

MapReduce is a robust model that mainly designed to handle large and complex datasets in parallel using large number of machines based on Hadoop framework [3]. Data is processed in the MapReduce programming model using Map and Reduce functions. Input data is divided into blocks by MapReduce job that process them at two different stages, the first one is Map stage that processes the input data and move the output to the reduce stage [25]. MapReduce is widely used for various applications such as healthcare, social media, indexing, and machine learning beacuase of its distributed and parallell capability of data processing. Since, large data processing has become a substantial issue due to the complex charcterstics of data, the MapReduce is adopted to be a solution by many organizations [26]. The architecture of MapReduce is shown in Figure 3.

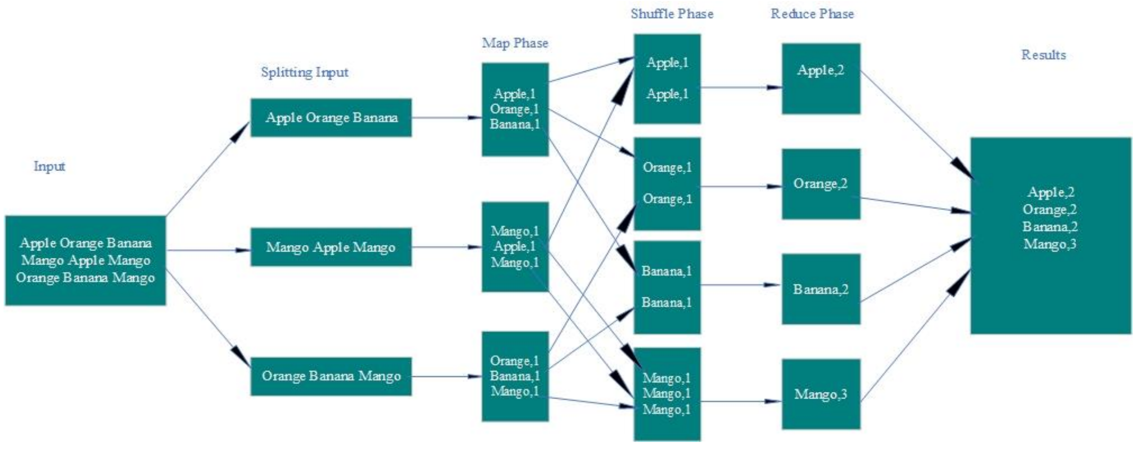


Figure 3. MapReduce architecture

3.4. Hive

Apache Hive is an open source distributed data warehouse developed by Facebook and Apache to read, write and manage massive datasets using SQL in distributed environment [2], [27]. It is designed to be run on Hadoop cluster using multiple commodity machines to analyse, summerize and query massive data that are stored in the Hadoop distributed file system. Hive is a data warehouse that performs different analytical features, such as windowing and partitioning [28], [29]. It is scalable system that can run queries using MapReduce programming model across several nodes of Hadoop cluster. It provides structured query language (SQL) dialect, namely Hive query language (HiveQL) and the results of HiveQL query are stored at different nodes of Hadoop cluster [30]. Furthermore, Hive supports API web interface and web browser to allow clients access the Hive database. Hive can be deployed in different production environments like GNU/Linux and Windows [27], [31]. It is adopted by many companies like Amazon that uses it in Amazon elastic MapReduce because of its brilient characteristics, such as scalability, speedup and extensibility. The main components of Hive are explained below as follows:

3.4.1 Hive client

Hive client is one of the main components of Hive. It is characterized into three types, which are thrift, JDBC and ODBC. The types of clients are shown in:
- Thrift client: It enables the Hive server to serve client requests based on Apache thrift [32], [33].
- JDBC client: It allows the connections between Java applications and Hive through thrift [32], [33].
- ODBC client: All applications based on ODBC protocol can be connected to Hive using ODBC driver. It also uses thrift to perform the communication between the Hive server and applications [33].

3.4.2 Hive services

Hive services are one of the major components of Hive. Hive provides different services like Hive server2, Beeline to perform queries. The following services of Hive are explained:
- Beeline: It is a JDBC client that allows users to submit its quiries and and commands via Beeline command shell supported by Hive server2. It can execute SQL quiries based on SQLLine CLI [34], [35].
- Hive User interface: the interaction between user and HDFS is provided via several interfaces, such as Hive Web UI, Hive command line, and Hive HD Insight [35].
- Hive server: It is known as Apache thrift server. It is responsible to receive requests from various clients and send them to Hive driver [35].
- Hive driver: All HiveQL statements submitted by the user are received by Hive driver via command shell. It handles all queries by creating sessions and send them to the compiler [35].
- Hive compiler: It uses the metadata stored in metastore to perform semantic analysis and type-checking on various query blocks and query expressions. It also creates directed acyclic graph (DAG) as execution plan that represents MapReduce jobs, metadata operation and HDFS operation [34].
- Optimizer: It is used to improve the efficiency and scalability by splitting the tasks and performing the transformation operations on the execution plan [34].
- Meta Store: Schema or metadata of tables, databases and other entities stored in relational database management system (RDBMS) datastore are quired by Metastore through application programming interfaces (APIs) [34].
- HiveQL Process Engine: It enables users to write queries for MapReduce jobs using HiveQL [34].
- Execution Engine: After the compilation and optimization steps, the executin engine will perform the execution process of quiries and obtain results using Hadoop MapReduce [34]. Hive architecture is illustrated in Figure 4 in *appendix*.

4. THE IMPLEMENTATION OF COVID-19 QUERING AND ANALYSIS USING HIVE BASED ON HADOOP CLUSTER

The query and predictive analytics of COVID-19 data is performed using Hive based on Hadoop cluster and MapReduce. In this paper, Hive is used to query and analyse vast number of COVID-19 datasets. These datasets are taken from Baqubah Teaching Hospital in Baqubah city, which contains the number of COVID-19 cases, number of deaths, severe cases, mild cases and cured cases. The proposed system can store COVID-19 data using HDFS, while the querying and analysis is performed by Hive using both of MapReduce programming model and Hadoop cluster. Datasets are converted into CSV formt, then loaded into HDFS in a local path using dfs -copyFromLocal. External table and ORC table are created to receive CSV data into Hive.Once datasets are loaded into Hive, it will perform the processing, quering and analysis using MapReduce algorithm. The query is submitted to the driver by the Hive client through UI. Once the query is received by the driver, it will submit it to the compiler that generates a query plan and transforms the SQL statement into

MapReduce tasks. After that, the compiler connects to the metastore that contains the data schem. All data definition language tasks, such as database creation, table creation and table modification are performed by connecting to the metastore. The job is then submitted by the execution engine to the Hadoop cluster and MapReduce for processing. Hive jobs processing is performed by the job tracker of Hadoop MapReduce. The COVID-19 datsets are automatically divided into several blocks. The size of each block is assigned to be 128 MB during the settings of configuration file in the Hadoop cluster through (hdfs-site.xml).

The computation process is performed on these blocks across severel nodes of the Hadoop cluster. The MapReduce process is responsible to process the Hive queries and send the intermediate results to the reducers to obtain the final results and store them on data nodes. Then, the results are sent to the execution engine, which is responsible to pass them to the driver. The transformation of input format to/from Hive row objects is done by Hadoop through Hive serializer/deserializer. Finally, the Hive client retrives the final results from the driver using Hive UI. The MapReduce process consists of several tasks and all these tasks are executed parallely to process the COVID-19 datasets. The number of Map tasks depends on Map slots number. The number of Map slots is set in Hadoop cluster by accessing (mapred-site.xml). In our proposed work, each Virtual Machine was assigned with one slot. As a consequence, eight slots have been assigned and, eight Map tasks have been carried out parallely to process COVID-19 datasets. The specifications of Hardware resources like physical memory and CPU Cores can play important role in assigning the number of Map slots to each virtual machine. The responsibility of Map tasks is to handle the data blocks, where every Map task is specified to process single data block. Once the Map process is completed, the intermediat output will be moved to the reduce tasks for processing. Same as Map tasks, reduce tasks number is also relied on the reduce slots number. Reduce slots number is set through the settings of Hadoop cluster. In this work, we assigned eight reduce slots and hence, eight reduce tasks are run to achieve the final output. As aforementioned, Hadoop offers remarkable features like reliability, scalability and fault tolerance. The blocks of datasets are duplicated across multiple nodes in the cluster to avoid any crash or failures. Furthermore, more machines can be easily attached to the cluster to improve the speed of computation process for our proposed system. Fault tolerance is also provided in the proposed system in case of any node failure by moving the running tasks from the failed node to another working node. The architecture of the proposed system is illustrated in Figure 5.

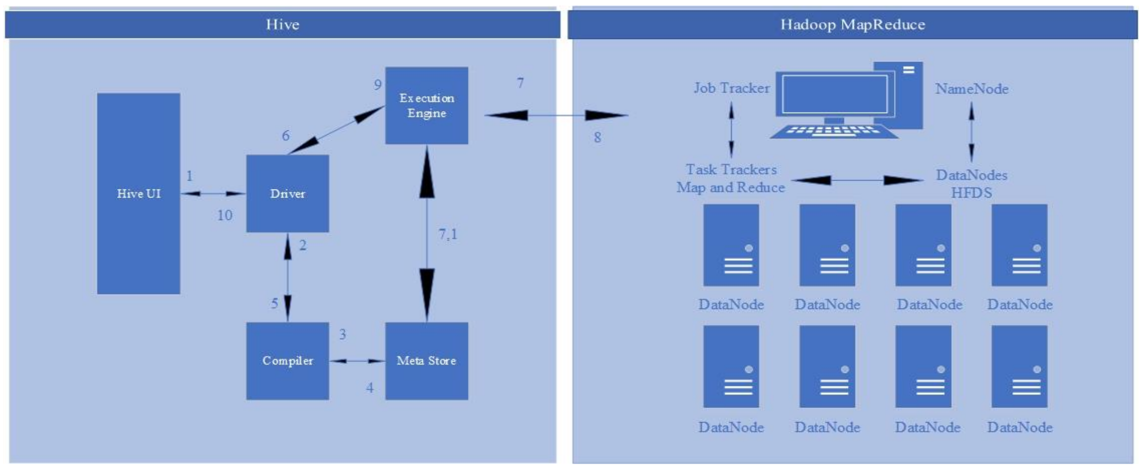


Figure 5. Proposed system

4.1. Proposed system evaluation

Evaluation on COVID-19 data quering and analysis by using Hive and Hadoop MapReduce framework has been performed. Comparison between relational database management system and our proposed system have been conducted. The performance of Hive using Hadoop cluster have been evaluaed in terms of data load, I/O operation, data transfer network and query execution time. The evaluation of proposed work has been performed on COVID-19 datasets provided by Baqubah teaching hospital. Due to the small size of these datasets, we duplicated them to be tens of Gigabytes to make big data scenario. The implementation of the proposed work was conducted using local Hadoop cluster consisting of eight nodes. These nodes have been created virtually by using oracle virtual box. Eight GigaByte of RAM and Four CPU cores are assigned to each virtual node with total 550 GB of storage. The setup of the cluster was performed using two high performance servers. Apache Hadoop 3.3.2 and Hive 3.1.3 have been installed on all virtual machines.

Moreover, MySQL has also been installed for RDBMS. Hive was employed to query and analyse COVID-19 data using the Hadoop MapReduce on Eight virtual machines. The specifications of cluster resources are illustrated in (Table 1 and Table 2).

Table 1. Hardware specifications of Hadoop cluster

Servers	Cluster resources	amount
“Intel Xeon X5550 server 1”	“CPU”	“32 cores:
“and uxisvm04 server 2”	“Processor”	“2.27 GHz”
	“Storage”	“550 GB”
	“Connection”	“1 Gbit Ethernet”
	“Memory”	“64 GB”

Table 2. Software packages of the proposed work

Host operating system	Guest operating system	Software
Microsoft windows server 2012 R2	Ubuntu Linux 18.04 (x86, 64-bit)	Hive 3.1.3, Hadoop 3.3.2, MySQL 5.7.39

5. RESULTS AND DISCUSSION

Several experiments are performed to evalute the performance of our proposed work. Different data sizes ranging from 1 GB to 10 GB were applied to compare the proposed work with relational database management system (RDBMS). As mentioned before, MySQL is installed as an open-source relational database management system (RDBMS). It is clearly observed that the performance of the proposed system computation outperforms RDBMS. From Figure 6, we can see that the execution time of COVID-19 datasets processing is decreased using the proposed work. However, when using RDBMS, the execution time is increased especially for larg data sizes. The reason is the distributed nature of Hive that can distribute the processing across multiple nodes in Hadoop cluster and, hence reduce the execution time by utilizing more resources of cluster nodes. The performance of Hive queries in terms of data load, I/O operation and network data transfer has also been evaluated by enabling compression. It is noticed that the query response time of different data sizes in Hive is less using compression at different phases (intermediate and final) as shown in Figure 7. This is because the amount of disk space used by Hive queries can be reduced by enabling compression, resulting in less execution time of Hive queries. Compression in Hive is set according to [36]. Figures 7(a)-7(c) show the response time of different data sizes using compression.

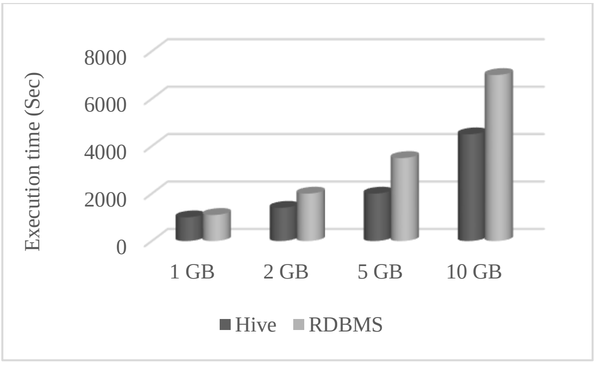


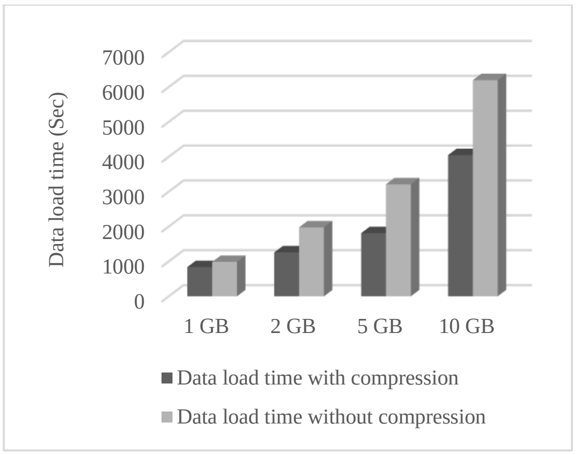
Figure 6. Hive execution time VS RDBMS execution time

The peformenace of Hive can also be improved by enabling the parallel execution of different stages. Each query in Hive is converted into one or more stages, which are MapReduce, sampling, merge and limit stage. Hive can execute these stages individually but a job may include stages which are not relying on one another and can be carried out simultaneously. This may allow the overall job to be completed quicker. It is observed from Figure 8 that the execution time of Hive queries is decreased by enabling the parallel execution.

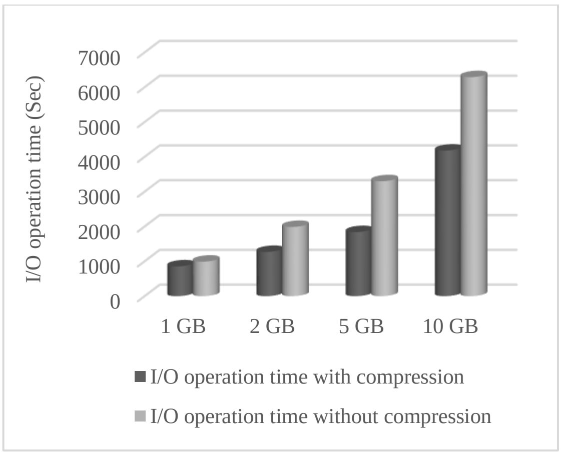
The strict mode of MapReduce can be enabled in configuration settings to improve the performance of Hive. Figure 9 shows the performance of Hive when enabling MapReduce strict mode. It is observed that the execution time of Hive queries is reduced by using strict mode. Strict mode is used to allow Hive avoiding the full scan of a table in a query especially for large tables. This improves the performance of Hive by avoiding unwanted delays of full table scan and consequently, reduce the execution time. This property is set to enable strict mode for MapReduce jobs based on [36].

As shown in Figure 10, Hive performance can aslo be enhanced by enabling the use of a single reducer task for multi group by operations. This helps to combine multiple GROUP BY operations into a single MapReduce job and consequently, reduce the execution time of Hive queries. Another parameter that can be tuned to improve the performance of Hive is the number of parallel reduce tasks. Figure 11 illustates the performance of Hive quires using the tuned parameter. It can be clearly seen that the execution time of Hive queries is decreased by using more reducers comparing to using less reducers. The execution time of Hive queries is reduced because multiple reducers are used parallelly to process them, therefore the large number of reducers is used the less execution time is achieved.

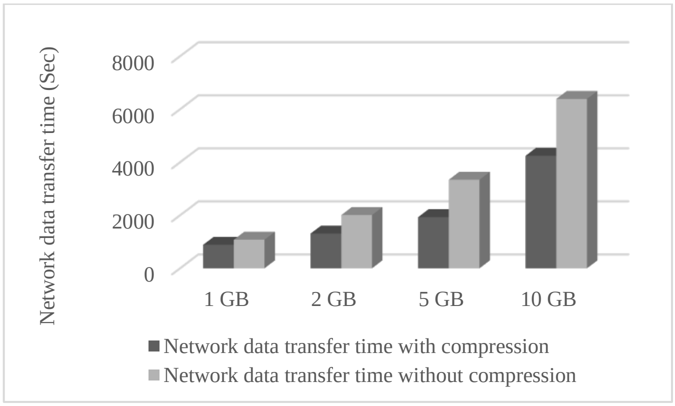
Cost-based optimization is a brilliant feature provided by Apache Hive. Further optimization based on query cost can be achieved by enabling this feature. This can result in different decisions about which joins to use, how to order the joins and the degree of parallelism when performing them. Cost based optimization is used to optimize and calculates different plans cost and examine the tables and queries. This helps in reducing th execution of queris and and cutting down the utilization of computing resources. Figure 12 shows that the execution time of Hive queries using cost-based optimization can be significantly reduced. Hive-site.xml is configured to enable cost-based optimization according to [36]. The performance of Hive queries can be significantly improved using optimized record columnar (ORC) file format. As mensioned before, ORC table are created to receive CSV data into Hive. "Hive table with ORC file format can be created by adding STORED AS ORC clause to CREATE TABLE command in Hive". Figure 13 shows the performance of Hive when using ORC compared to RCFile (Record Columnar File). It is clearly observed that reading and writing data to HDFS is faster using ORC as it provides a single file of each task output, which leads to reduce the load of NameNode.



(a)



(b)



(c)

Figure 7. Hive performance using compression; (a) data load time when using Compression, (b) I/O operation time when using compression, and (c) network data transfer operation time when using compression

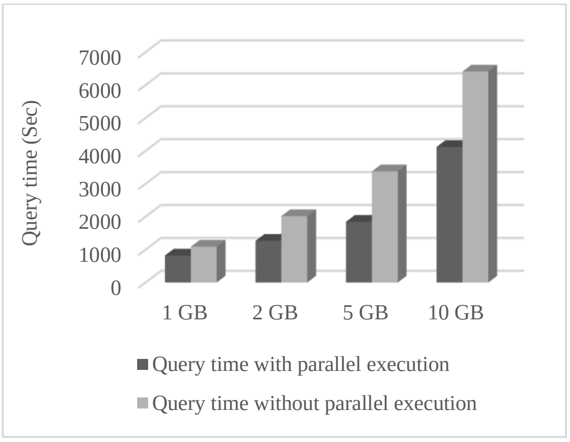


Figure 8. Hive query time when using parallel execution

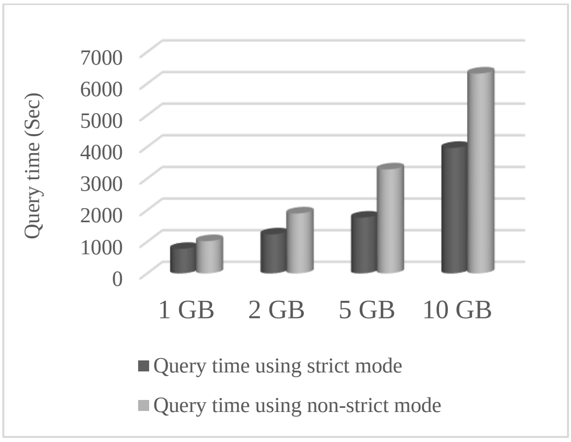


Figure 9. Hive query time when using strict mode

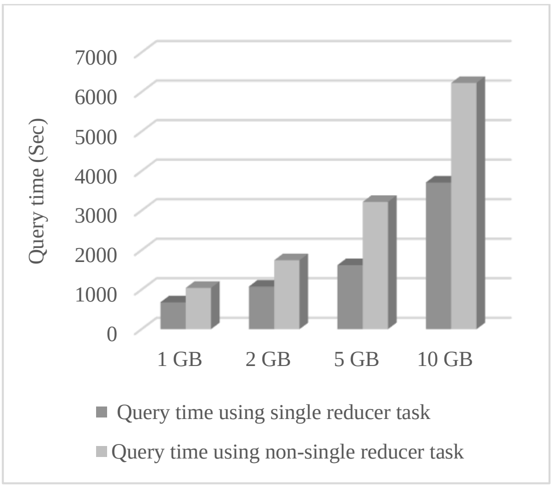


Figure 10. Hive query time using single reducer

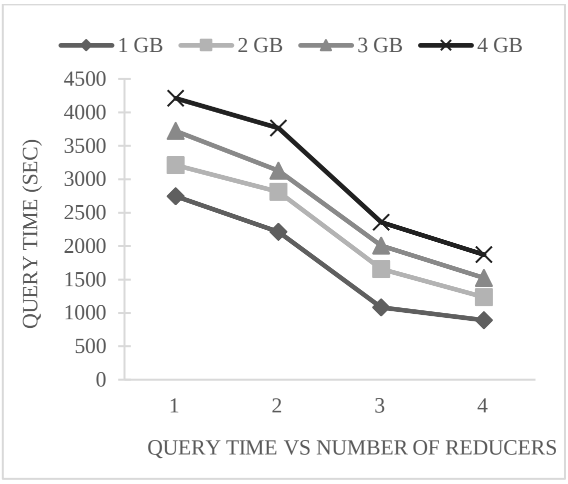


Figure 11. Hive query time using different reducers

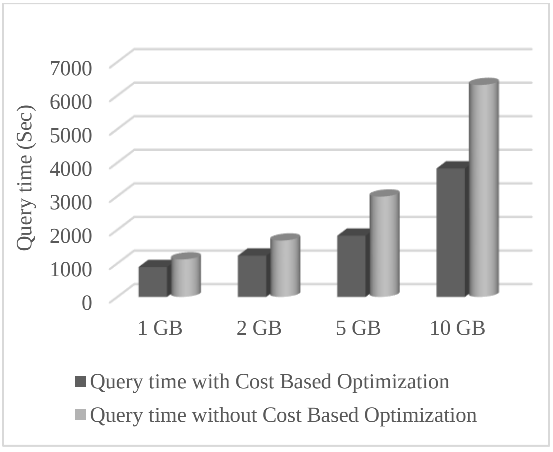


Figure 12. Hive query time using cost-based optimization

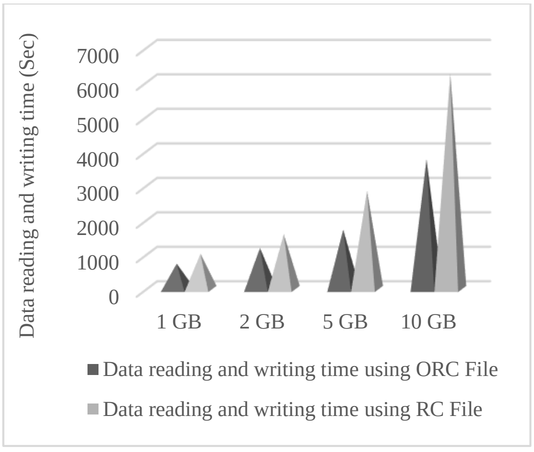


Figure 13. Data reading and writing time using ORC and RC file

6. CONCLUSION

In this paper, Hive has been employed on top of a Hadoop cluster to query and analyse COVID-19 datasets. Comparision between RDBMS and our proposed work was carried out. Some of Hive properties and parameters such as compression, MapReduce strict mode, parallel reduce tasks, cost-based optimization and optimized record columnar have been tuned and enabled to improve the proposed work. Several experiments have been run to evaluate the proposed work. The performance evaluation of the proposed work has been carried out by using a cluster of eight virtual machines. Results show that the proposed work outperformed RDBMS in terms of query processing time and scalability. The results also show that the performance of our proposed work in terms of data load, I/O operation, network data transfer, query execution time and data read and write were improved by tuning and enabling the important parameters of hive.

APPENDIX

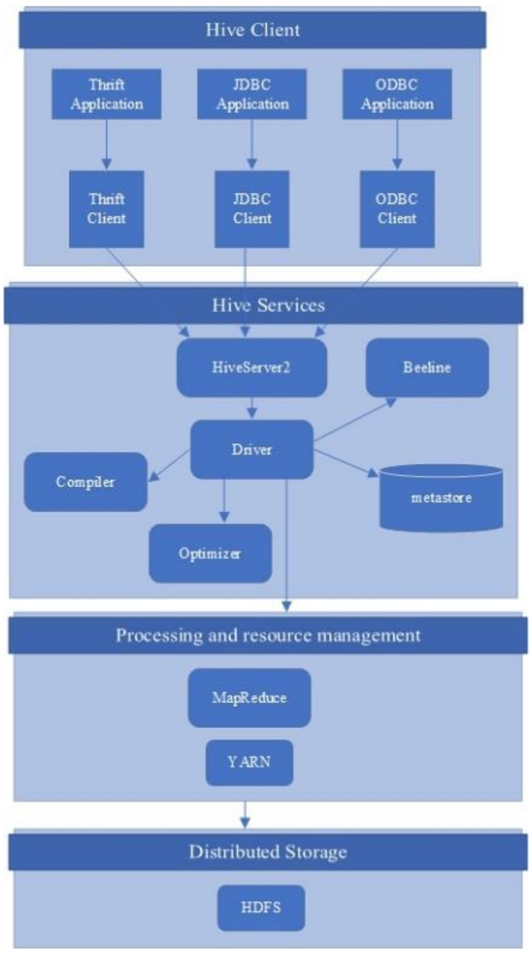


Figure 4. Hive architecture

REFERENCES

[1] "Apache Hadoop," *Apache Hadoop*. https://hadoop.apache.org/ (accessed Jul. 29, 2022).
[2] D. Vohra, "Apache Hive," *Practical Hadoop Ecosystem*, 2016.
[3] The Apache Software Foundation, "MapReduce Tutorial," *Source*, 2008. https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html (accessed Aug. 31, 2022).
[4] M. M. R. Rana, F. Rahman, J. Al Faysal, and M. A. Rahman, "An Effective Prediction on COVID-19 Prevalence for India and Japan using Fbprophet Model," *Asian Journal of Research in Computer Science*, pp. 16–28, Aug. 2021, doi: 10.9734/ajrcos/2021/v11i230258.
[5] S. J. Alsunaidi *et al.*, "Applications of big data analytics to control covid-19 pandemic," *Sensors*, vol. 21, no. 7, p. 2282, Mar. 2021, doi: 10.3390/s21072282.

[6] N. Kumar and S. Susan, "COVID-19 Pandemic Prediction using Time Series Forecasting Models," in *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*, Jul. 2020, pp. 1–7, doi: 10.1109/ICCCNT49239.2020.9225319.

[7] P. N. Mahalle, N. P. Sable, N. P. Mahalle, and G. R. Shinde, "Data Analytics: COVID-19 Prediction Using Multimodal Data," *SpringerBriefs in Applied Sciences and Technology*, pp. 1–10, 2020, doi: 10.1007/978-981-15-6572-4_1.

[8] Y. Chen, C. K. Leung, S. Shang, and Q. Wen, "Temporal data analytics on COVID-19 data with ubiquitous computing," *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2020, pp. 958–965, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00146.

[9] V. K. Gupta, A. Gupta, D. Kumar, and A. Sardana, "Prediction of COVID-19 confirmed, death, and cured cases in India using random forest model," *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 116–123, 2021, doi: 10.26599/BDMA.2020.9020016.

[10] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, and A. Y. Zomaya, "Federated Learning for COVID-19 Detection With Generative Adversarial Networks in Edge Cloud Computing," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 10257–10271, 2022, doi: 10.1109/JIOT.2021.3120998.

[11] R. Oruche *et al.*, "Evidence-Based Recommender System for a COVID-19 Publication Analytics Service," *IEEE Access*, vol. 9, pp. 79400–79415, 2021, doi: 10.1109/ACCESS.2021.3083583.

[12] A. Mourad, A. Srour, H. Harmanani, C. Jenainati, and M. Arafeh, "Critical impact of social networks infodemic on defeating coronavirus COVID-19 pandemic: Twitter-based study and research directions," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2145–2155, Dec. 2020, doi: 10.1109/TNSM.2020.3031034.

[13] X. Yu, M. D. Ferreira, and F. V. Paulovich, "Senti-COVID19: An interactive visual analytics system for detecting public sentiment and insights regarding COVID-19 from social media," *IEEE Access*, vol. 9, pp. 126684–126697, 2021, doi: 10.1109/ACCESS.2021.3111833.

[14] S. Dash, C. Chakraborty, S. K. Giri, S. K. Pani, and J. Frnda, "BIFM: Big-Data Driven Intelligent Forecasting Model for COVID-19," *IEEE Access*, vol. 9, pp. 97505–97517, 2021, doi: 10.1109/ACCESS.2021.3094658.

[15] I. Ahmed, M. Ahmad, G. Jeon, and F. Piccialli, "A framework for pandemic prediction using big data analytics," *Big Data Research*, vol. 25, p. 100190, Jul. 2021, doi: 10.1016/j.bdr.2021.100190.

[16] C. K. Leung, T. N. Kaufmann, Y. Wen, C. Zhao, and H. Zheng, "Revealing COVID-19 Data by Data Mining and Visualization," *Lecture Notes in Networks and Systems*, vol. 312, pp. 70–83, 2022, doi: 10.1007/978-3-030-84910-8_8.

[17] C. K. Leung, D. L. X. Fung, and C. S. H. Hoi, "Health analytics on COVID-19 data with few-shot learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12925 LNCS, 2021, pp. 67–80.

[18] T. White, *Hadoop: The Definitive Guide*. O'Reilly Media, 2009.

[19] A. Hong, W. Xiao, and J. L. Ge, "Big data analysis system based on cloudera distribution hadoop," in *2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, May 2021, pp. 169–173, doi: 10.1109/BigDataSecurityHPSCIDS52275.2021.00040.

[20] C. Lam, *Hadoop in Action*. Simon and Schuster. 2010.

[21] J. Zhang and M. Lin, "A comprehensive bibliometric analysis of Apache Hadoop from 2008 to 2020," *International Journal of Intelligent Computing and Cybernetics*, vol. 16, no. 1, pp. 99–120, Mar. 2023, doi: 10.1108/IJICC-01-2022-0004.

[22] K. Singh and R. Kaur, "Hadoop: Addressing challenges of Big Data," in *2014 IEEE International Advance Computing Conference (IACC)*, Feb. 2014, pp. 686–689, doi: 10.1109/IAdCC.2014.6779407.

[23] S. Sakr and A. Y. Zomaya, *Encyclopedia of Big Data Technologies*. Cham: Springer International Publishing, 2019.

[24] D. Borthakur, HDFS architecture guide, (2008) Accessed: Aug. 31, 2022. [Online]. Available: https://archive.cloudera.com/cdh/3/hadoop-0.20.2-cdh3u6/hdfs_design.pdf

[25] D. Borthakur, "The hadoop distributed file system: Architecture and design," *Hadoop Project Website*, pp. 1–14, 2007, [Online]. Available: http://cloudcomputing.googlecode.com/svn/trunk/??/Hadoop_0.18_doc/hdfs_design.pdf.

[26] S. N. Khezr and N. J. Navimipour, "MapReduce and Its Applications, Challenges, and Architecture: a Comprehensive Review and Directions for Future Research," *Journal of Grid Computing*, vol. 15, no. 3, pp. 295–321, Sep. 2017, doi: 10.1007/s10723-017-9408-0.

[27] J. Camacho-Rodríguez *et al.*, "Apache Hive," in *Proceedings of the 2019 International Conference on Management of Data*, Jun. 2019, pp. 1773–1786, doi: 10.1145/3299869.3314045.

[28] R.-G. Stan, C. Negru, L. Bajenaru, and F. Pop, "Data and Systems Heterogeneity: Analysis on Data, Processing, Workload, and Infrastructure," Springer International Publishing, 2021, pp. 77–90.

[29] K. Bansal, P. Chawla, and P. Kurle, "Analyzing Performance of Apache Pig and Apache Hive with Hadoop," in *Lecture Notes in Electrical Engineering*, vol. 478, 2019, pp. 41–51.

[30] P. Bhat and P. Hegde, "Novel methodologies for processing structured big data using hadoop framework," in *Lecture Notes in Networks and Systems*, vol. 171, 2021, pp. 565–572.

[31] S. Shaw, A. F. Vermeulen, A. Gupta, and D. Kjerrumgaard, *Practical Hive: a guide to Hadoop's data warehouse system*, 1st ed. ed. Apress, 2016.

[32] D. Du, *Apache Hive Essentials*, vol. 53, no. 9. Packt Publishing Ltd, 2013.

[33] K. Srinivasa, G. Siddesh, and H. Srinidi, *Network Data Analytics: A Hands-On Approach for Application Development*. Springer, Cham, Switzerland, 2018.

[34] H. Bansal, S. Chauhan, and S. Mehrotra, *Apache Hive Cookbook*. Apache Hive Cookbook., 2016.

[35] E. Capriolo, D. Wampler, and J. Rutherglen, *Programming Hive*. O'Reilly Media, 2012.

[36] "Hive Performance Tuning-Hadoop Online Tutorials," *hadooptutorial.info*. http://hadooptutorial.info/hive-performance-tuning/ (accessed Mar. 19, 2023).

BIOGRAPHIES OF AUTHORS

Ali Abbood Khaleel recieved the Ph.D. degree in Electronic and Computer Engineering from Brunel University, London, U.K. in 2018. His research interests include big data analytics, cloud computing, high performance computing, parallel computing, distributed computing and storage and software-defined networks. He can be contacted at email: draliak@bauc14.edu.iq.

Ali Noori Kareem received his Ph.D. degree from Universiti Malaysia Perlis in 2019, his research interests include computer network and communication, cloud computing and big data. He can be contacted at email: dr.alinoori@bauc14.edu.iq.

Laith Hikmet Mahdi received his B.Sc. from Al- Yarmouk University College, Iraq, Computer Engineering Techniques Dep.in 2015 and Master Degree in Electronics and Nano Electronics from Mordovian State University, Russia in 2018. His research interests include high performance computing, parallel computing, distributed computing and storage. He can be contacted at email: laith@bauc14.edu.iq.