World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Review Article)

Comparative analysis of MapReduce and Apache Tez Performance in Multinode clusters with data compression

Sifat Ibtisum [1], S M Atikur Rahman [2, *] and S. M. Saokat Hossain [3]

[1] *Department of Computer Science, Missouri University of Science & Technology, Rolla, Missouri, MO 65409, USA.*

[2] *Department of Industrial, Manufacturing and Systems Engineering, University of Texas at El Paso, TX 79968, USA.*

[3] *Department of Computer Science, Jahangirnagar University, Savar, Dhaka-1342, Bangladesh.*

World Journal of Advanced Research and Reviews, 2023, 20(03), 519–526

Publication history: Received on 26 October 2023; revised on 04 December 2023; accepted on 06 December 2023

Article DOI: https://doi.org/10.30574/wjarr.2023.20.3.2486

Abstract

This article conducts a thorough comparative analysis of Apache Tez and MapReduce in the context of big data processing. It focuses on key performance metrics, scalability, and ease of use. The analysis begins with an overview of the architectural distinctions between the two frameworks, emphasizing their fundamental design principles. A detailed performance evaluation follows, considering factors such as execution time, resource utilization, and throughput across diverse workloads. The study explores scalability by examining how Apache Tez and MapReduce respond to increasing data volumes and computational demands. Cluster size effects, resource allocation strategies, and adaptability to dynamic workloads are scrutinized. Additionally, the article evaluates the frameworks' ease of use for developers and administrators, incorporating aspects like programming model simplicity, debugging capabilities, and system configurability. User experiences are gathered through surveys and practical use cases. The conclusions drawn from this analysis offer valuable insights for organizations and practitioners seeking suitable distributed computing frameworks. By addressing both performance and user experience, the article aims to provide a comprehensive perspective on the strengths and weaknesses of Apache Tez and MapReduce, assisting decision-makers in making informed choices for their big data processing requirements.

Keywords*:* Apache Tez; Spark Core; Compression; Cluster size; MapReduce.

1. Introduction

Hadoop represents the most effective tool for handling big data in contemporary research. It can be deployed on either a single or multiple clusters. An illustration of real-world big data analytics involves the utilization of social information to forecast and align individuals' lifestyles on Facebook. The term "business intelligence" is employed to describe the analysis of extensive datasets encompassing both social media and business information, which are highly intricate in terms of predicting the dynamic trends in customer demands for various products. Hadoop serves as the ultimate solution for managing big data, incorporating components such as HBase, Hive, R connectors, Mahout, Pig, and Oozie [1- 6]. These components operate on the Hadoop Distributed File System (HDFS) and the second version of MapReduce, known as "YARN". HDFS represents a logical disk structure across the physical directories within each data node of the Hadoop cluster. It communicates via the TCP protocol port 22, such as Secure Shell (SSH), on each node in the cluster. The HDFS disk is highly fault-tolerant since it has multiple replicas in the HDFS configuration. YARN exclusively facilitates access to and processing of the data. The number of replicas directly impacts the storage capacity of HDFS. Hortonworks and Cloudera, as organizations providing the Hadoop platform, have proposed a data compression algorithm within Hadoop, which can effectively reduce disk storage and network bandwidth between each node in the cluster. The Hadoop compression suite comprises DEFLATE, GZIP, Bzip2, LZ4, and Snappy. YARN encompasses two

* Corresponding author: S M Atikur Rahman.

frameworks for processing data: the MapReduce framework and the Tez framework. The MapReduce framework offers batch processing and serves as the default framework for the Hadoop cluster. In contrast, Tez supports interactive processing, but it is intricate to install and configure using the binary file obtained from the Apache Tez website. R. Singh et al. [6] explored the application of the Tez framework with Pig scripts, suggesting that this framework was more suitable for pre-structuring and processing data compared to MapReduce.

2. APACHE SPARK

Rattanaopas et. al. [1] focused on the performance evaluation of compression methods that are available on Hadoop cluster. They study to evaluate a comparison with those of frameworks Mapreduce and Tez. Tez can reduce the process of data stored in HDFS. It is significance for the research hypothesis. In the results of this research, they show an execution time from Hadoop's benchmark (e.g. word count, terasort) and the best methods to use compression in Hadoop cluster with big data. They purpose an alternative method for improving execution time which is the performance indicator of Hadoop cluster. R. Singh et. al. [6] emphasis on both theoretical empirical parameters and try to analyze that how these two frameworks react when particular job is submitted to multimode cluster installed on amazon cloud. Chandrabhan et. al. [7] delves into MapReduce and Apache Tez data compression techniques that efficiently compresses and decompresses a large amount of data. Kannan et. al. [8] delves into Hadoop and MapReduce architecture and its shortcomings and examines alternatives such as Apache Tez and Apache Spark for their suitability in iterative and interactive workloads.

3. HADOOP

In the realm of big data solutions, Hadoop stands as a linchpin, playing a pivotal role in efficiently managing vast datasets. Widely embraced by major commercial websites like Google and Yahoo, this portable software has become indispensable for organizations navigating the challenges of processing and analysing massive amounts of data. Hadoop's versatility is evident in its compatibility with the Java JDK, making it accessible across different platforms. At its core, Hadoop introduces the Hadoop Distributed File System (HDFS), a specialized file system designed for distributed storage. This file system ensures seamless data accessibility and is compatible with a range of operating systems, including Windows, Linux, and Mac OS X. The framework of Hadoop encompasses essential components such as MapReduce, Tez, and Spark. MapReduce, a programming model, facilitates the distributed processing of large datasets across clusters of computers. Tez and Spark enhance Hadoop's capabilities, offering alternative processing engines that cater to specific use cases and scenarios One of Hadoop's notable strengths is its compatibility across different operating systems. Whether on Windows, Linux, or Mac OS X, organizations can leverage Hadoop's capabilities without worrying about platform-specific limitations. This adaptability makes it a versatile solution for enterprises with diverse IT infrastructures. Hadoop's architecture is designed for scalability, allowing it to seamlessly grow from single servers to large clusters. Figure 1 describes the Apache Hadoop ecosystem.

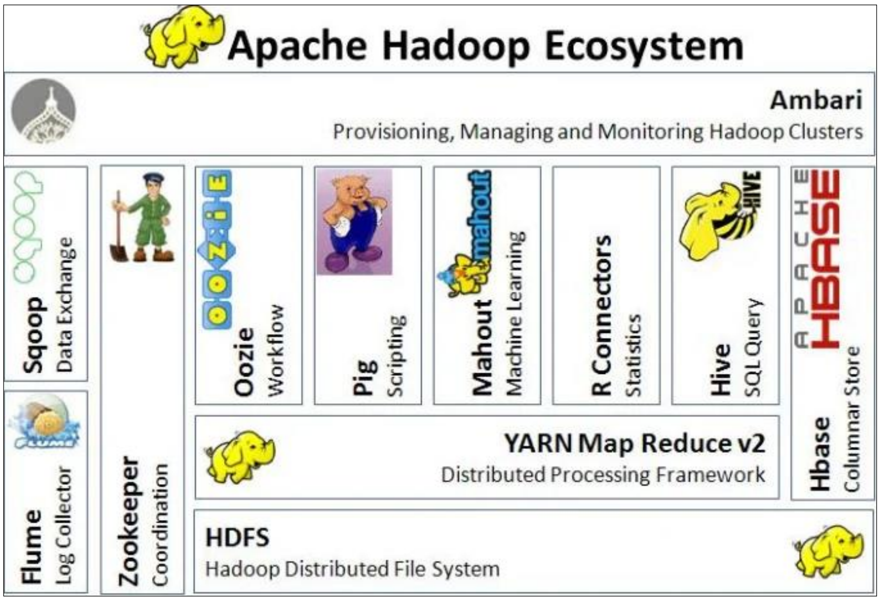


Figure 1 Illustration of Hadoop ecosystem [2]

This scalability is crucial in addressing the evolving needs of organizations dealing with expanding data volumes. Additionally, Hadoop's open-source nature contributes to its accessibility, fostering a collaborative environment where organizations can harness the collective expertise of the community.

4. MAPREDUCE

MapReduce, the original framework for distributed processing in Hadoop, is characterized by its batch-oriented nature. This method involves two distinct tasks, namely the Map and Reduce tasks. Following the Map task is the Sort/Shuffle task, and the Reduce task marks the conclusion of the entire process. One of the key advantages of MapReduce is its ability to distribute data in parallel across a multitude of machines. It is worth noting that the MapReduce processing model mandates a map phase preceding a reduce phase, and involves the temporary storage of data in the Hadoop Distributed File System (HDFS) after each map and reduce phase. The process of storing data in HDFS during processing may appear to be an inefficient use of MapReduce's time [2].

5. APACHE TEZ

Apache Tez is a data processing framework that was developed to improve the efficiency and performance of Hadoop MapReduce. It provides a more flexible and optimized execution engine for processing complex data workflows. Tez allows users to express their data processing tasks in the form of a directed acyclic graph (DAG), where each node represents a processing task, and edges represent the data flow between tasks. Unlike the traditional two-stage execution model of MapReduce, Tez enables multi-stage data processing with a more efficient and fine-grained control over the execution flow. Table 1 illustrates the comparison between MapReduce and Apache Tez [2].

6. Compression

The Linux operating system features a native compression algorithm, such as gzip, deflate, and bzip2. Both gzip and deflate utilize the zlib format for general compression. The concept of LZ777 was originally introduced by snappy, which employs Google's technology and design for the Hadoop Ecosystem. In contrast, bzip2 is the sole compression codec capable of splitting file formats. This study focuses on the utilization of a native compression tool within the Hadoop environment, specifically [1]:

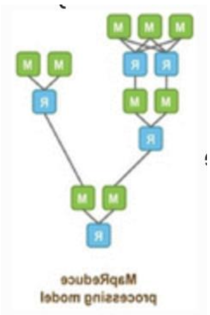
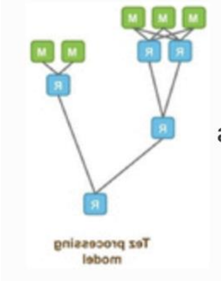
6.1. Bzip2

Bzip2 is an advanced data compression algorithm based on libbzip2, distributed under an open-source (BSD-style license) [9-13]. It employs a technique similar to the PPM family, offering compression performance twice as fast and decompression six times faster than the older version 1.0.6. It is beneficial for overfull disk drives, distribution CDs, backup tapes, and USB sticks, reducing download times over a network. In recovery mode, it can restore compressed data and decompress undamaged file parts, utilizing libbzip2 to directly read and write bz2 files with compressed data in memory [1].

6.2. Snappy

Google based on the LZ77 concept in 2011 developed Snappy, previously known as Zippy, in C++. Its primary goal is to maximize compression speed, resulting in the highest possible speed [14-17]. Benchmark tests for snappy utilize a Core i7 with a single core in 64-bit mode, achieving compression ratios 20-100% lower than gzip. In Hadoop clusters, snappy is the most widely used native compression codec, extensively employed in Cassandra, Hadoop, LevelDB, MongoDB, RocksDB, and Lucene [18].

Table 1 MapReduce vs. Apache Tez

Parameters	**MapReduce**	**Apache Tez**
Programming Model	MapReduce is well-suited for batch processing of large datasets.  **Figure 2(a)** MapReduce	Tez allows users to define a custom data flow model, which is more intuitive for certain types of computations.  **Figure 2(b)** Apache Tez
Execution Model	Mapreduce always requires a map phase before the reduce phase. [Figure 2(a)]	A single map phase and can have multiple reduce phase. [Figure 2(b)]
Optimization	Traditional MapReduce implementations have limitations in terms of optimization opportunities, and iterative algorithms can be less efficient.	Tez supports dynamic reconfiguration of the DAG, enabling more efficient execution of complex workflows.

7. Performance Benchmark

7.1. Benchmark tools

In [1], experiments were conducted to evaluate and compare the execution time between MapReduce and Tez. They ran two benchmarks to measure different solution of a dataset [1].

7.1.1. Word count benchmark

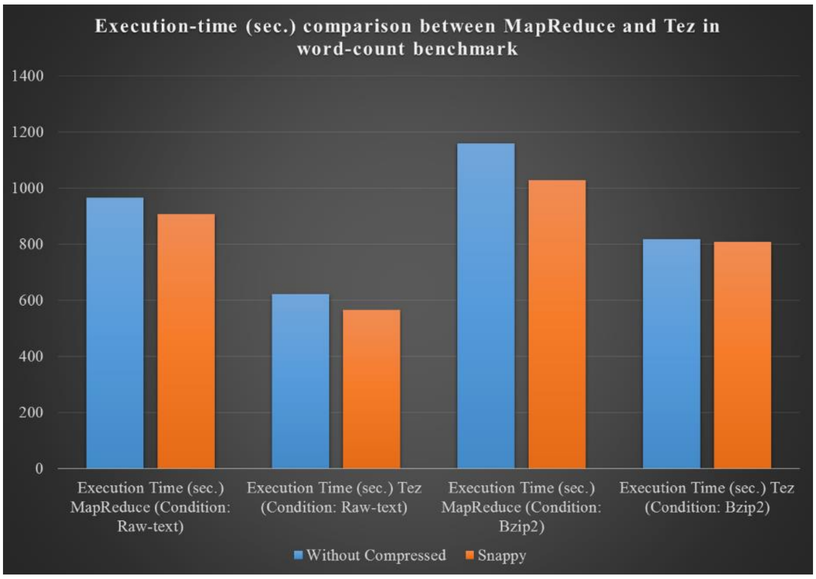


Figure 3 Execution Time comparison between MapReduce and tez in word-count benchmark [1]

The average execution time of word-count benchmark with those of the e-book files a raw (file size: 14.4 GB) and a Bzip2 compression (file size: 3.74 GB) on Hadoop cluster is shown in Figure 3. It compares between MapReduce and Tez frameworks. In Figure 3, snappy compression with Tez framework is the best execution time at 565.75 seconds as lower than others in a Raw-text. Tez framework can reduce an execution time more than ~37% of MapReduce framework in Hadoop cluster. In a compressed e-book text by Bzip2, snappy compression with Tez framework has a same better execution time at 807.25 seconds as lower than others. Bzip2 input file is around 1,093.33 seconds in MapReduce and 813.13 seconds in Tez. It increases an execution time up to 37% of Raw-text's execution-time trade-off with a disk space reducing more than 70%.

7.1.2. Terasort Benchmark

In [1], a 10 GB raw-text file was created by teragen command. An average execution time of terasort benchmark with snappy compression was evcaluated in map output. It shows a compression between MapReduce and Tez frameworks.

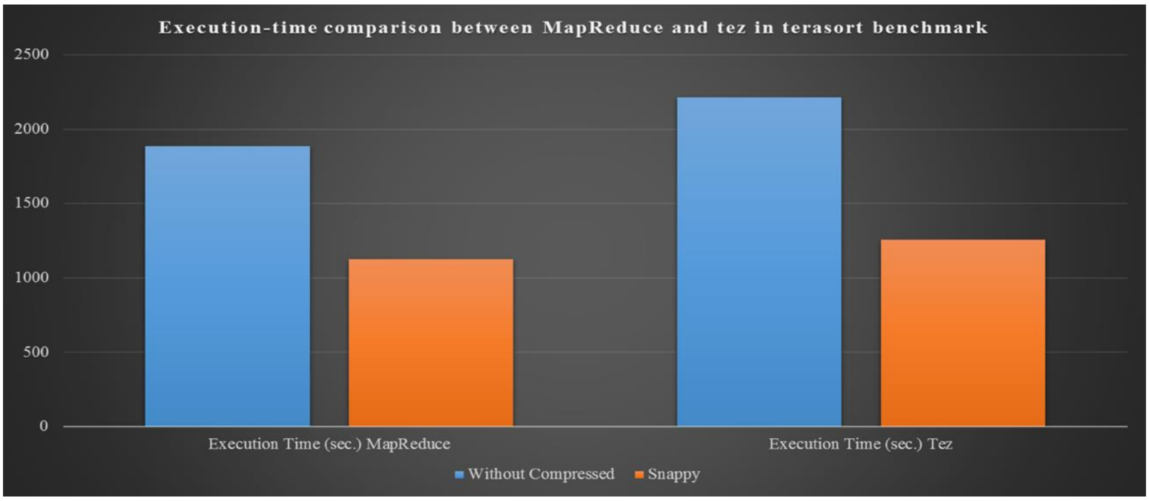


Figure 4 Execution-time comparison between MapReduce and Tez in Terasort Benchmark [1]

In Figure 4, MapReduce framework has execution time at 1,886.50 seconds which is better than Tez framework having execution-time at 2,212.00 seconds. In snappy compression, it has a same best performance in MapReduce framework that it is 1,150.25 second and increase performance up to 39% of a without compressed case. Tez framework is 1,256.00 second and increase performance up to 43.2% of a without compressed case. However, the compressed map output file with snappy compression that it can increase a computing performance more than 39% on both frameworks. On the other hand, Tez framework has a performance, which is lower than MapReduce framework around 13% in Hadoop cluster.

8. Result Comparison

The best performance found in a compressed map output by snappy with both of MapReduce and Tez frameworks in Figure 5 and 6. In word-count benchmark, Tez framework has a better performance because it does not need to temporary store data in HDFS during process. As the result, word-count output is lower than 100 MB. On the other hand, MapReduce always store data in HDFS that it has a direct effect of an execution time. In compressed input file by Bzip2 compression, it can reduce disk space but it must trade-off a performance around 37% with e-book text (14.4 GB). In terasort benchmark, MapReduce framework has a better performance than Tez framework for those of map output compressed and not compressed as shown in Figure 8. However, compressed map output by snappy compression keep a better performance than uncompressed. Output of terasort is a similar with input size 10GB. In our opinion, a very large data size has some effects on a performance of reduce phase in Tez. Tez must keep data in memory that it has a direct effect with a large output of reduce phase. Therefore, it can cause the lower performance of Tez comparing with MapReduce in this case. Rupinder Singh et al. [6] used the below dataset in order to run the experiment. Table 2 shows the details of the sample datasets used in the experiment.

Table 2 Sample datasets used in the experiment

Name	No of records	No of attributes
Geolocation	8013	10
Drivermilage	101	2

Different configuration of M3 instance are shown in the table 3, which is also used to run the experiment.

Table 3 Different type of instances used in our experiment

Model	vCPU	Mem (GiB)	SSD Storage (GB)
M3.medium	1	3.75	1*4
M3.large	2	7.5	1*32
M3.xlarge	4	15	2*40
M3.2xlarge	8	30	2*80

Here they have deployed multi node cluster with varying configuration. So firstly, they run apache pig script ten–ten times on both the frameworks at 3.75 GB, 1 CORE Machines of M3.medium type. Figure 5 shows the results of execution of script on both the frameworks. Vertical axis depicts the time in milliseconds and horizontal axis shows the no of runs of script. Fig clearly shows that Apache Tez has lesser execution time than MapReduce framework does.

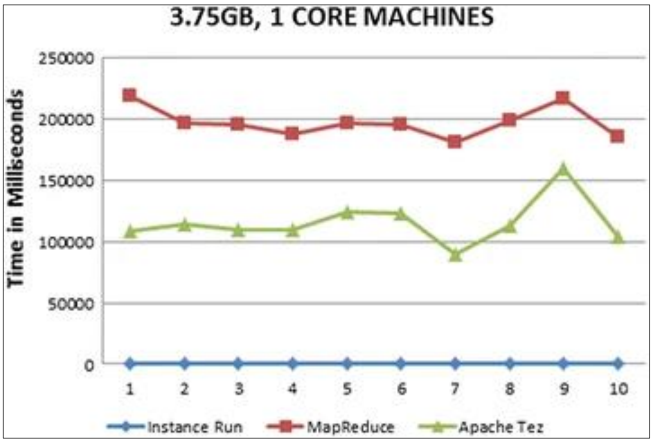


Figure 5 Results of execution on 3.75 GB, 1 core machines [6]

They extended their experiment by executing the same script on both the frameworks installed on machines having more no of cores and memory.

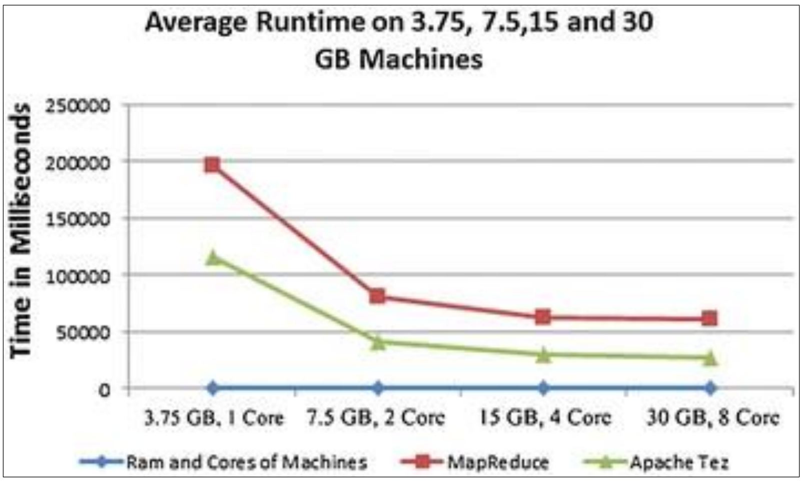


Figure 6 Average time for execution of script on different machines [6]

After execution of script on machines of different configurations, they calculated the average time of execution. Figure 6 shows that the execution time decreases as the no of available resources increased and when we move from 15 GB, 4

Core to 30 GB, 8 Core configurations there is slight decrease in slope. This shows that no of resources required for execution of script attains the peak value.

9. Discussion

This paper reviews both the frameworks used for execution of Pig Scripts. It tried to review both theoretical and empirical analysis based on some parameters. With the help of chosen parameters, it was possible to understand that how these frameworks differ from each other. Results show that Apache Tez is a better choice for execution of Apache Pig scripts, as MapReduce requires more resources in the form of time and storage.

10. Conclusion

This review article has thoroughly examined and contrasted the efficacy of Apache Tez and MapReduce, two important frameworks for processing large-scale data in the field of big data analytics. By conducting a thorough examination of several factors such as execution models, task scheduling, data location, and resource utilisation, we have obtained valuable understanding of the advantages and disadvantages of each framework.

Apache Tez, with its directed acyclic graph (DAG) execution paradigm, has demonstrated significant enhancements compared to conventional MapReduce. Tez's capability to optimise task scheduling and take advantage of data proximity has led to improved performance, making it very efficient for iterative and interactive processing workloads. Furthermore, Tez's ability to optimise dynamically, including features like parallelism at the vertex level and precise control over data flow, enhances its adaptability and leads to faster execution rates. However, MapReduce, a trailblazer in distributed data processing, continues to hold its importance, particularly for workloads involving batch processing. Although MapReduce may have limits in handling iterative algorithms and interactive workloads, it is nevertheless a reliable option for situations where fault tolerance and simplicity are of utmost importance. The evaluation of Apache Tez and MapReduce in terms of performance is intricate, and the selection between them ultimately relies on the precise demands and attributes of the data processing activities being performed. When choosing a framework for big data processing, organisations should thoroughly evaluate aspects such as the nature of the workload, data access patterns, and the overall system architecture. Apache Tez and MapReduce play crucial roles in the ever-changing landscape of big data, contributing considerably to the wide range of frameworks used for processing data. This review article offers valuable insights for researchers and practitioners, enabling them to make educated decisions that optimise performance and resource utilisation in their big data processing pipelines, according to the specific requirements of their applications.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] K. Rattanaopas, "A performance comparison of Apache Tez and MapReduce with data compression on Hadoop cluster," 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), NakhonSiThammarat, Thailand, 2017, pp. 1-5, doi: 10.1109/JCSSE.2017.8025950. \

[2] Ibtisum, S. (2020). A Comparative Study on Different Big Data Tools

[3] S M Atikur Rahman, Sifat Ibtisum, Ehsan Bazgir and Tumpa Barai. The Significance of Machine Learning in Clinical Disease Diagnosis: A Review. International Journal of Computer Applications 185(36):10-17, October 2023.

[4] S M Atikur Rahman, Sifat Ibtisum, Priya Podder and S. M. Saokat Hossain. Progression and Challenges of IoT in Healthcare: A Short Review. International Journal of Computer Applications 185(37):9-15, October 2023.

[5] Ibtisum, S., Bazgir, E., Rahman, S. A., & Hossain, S. S. (2023). A comparative analysis of big data processing paradigms: Mapreduce vs. apache spark. World Journal of Advanced Research and Reviews, 20(1), 1089-1098.

[6] Singh, R., Kaur, P.J. Analyzing performance of Apache Tez and MapReduce with hadoop multinode cluster on Amazon cloud. J Big Data 3, 19 (2016).

[7] Chandrabhan S. Jadhao, Prof. Harish K. Barapatre, “MapReduce and Apache Tez Data Compression Techniques”, Int J S Res Sci. Tech. 2018 Mar-Apr;4(5) : 1018-1021.

[8] Kannan, P. (2014). Beyond Hadoop MapReduce Apache Tez and Apache Spark.

[9] P. M. Szecowka and T. Mandrysz, "Towards hardware implementation of bzip2 data compression algorithm," 2009 MIXDES-16th International Conference Mixed Design of Integrated Circuits & Systems, Lodz, Poland, 2009, pp. 337-340.

[10] Sarker, B., Sharif, N. B., Rahman, M. A. & Parvez, A. S. (2023). AI, IoMT and Blockchain in Healthcare. Journal of Trends in Computer Science and Smart Technology, 5(1), 30-50. doi:10.36548/jtcsst.2023.1.003.

[11] Sarker, B., Sarker, B., Podder, P., & Robel, M. R. A. (2020). Progression of Internet Banking System in Bangladesh and its Challenges. International Journal of Computer Applications, 177(29), 11-15.

[12] "Project Gutenberg", Free ebooks by Project Gutenberg, 2017, [online] Available: https://www.gutenberg.org/.

[13] "CVE-2010-0405", bzip2 and libbzip2, 2017, [online] Available: http://www.bzip.org/index.html.

[14] N. Sarker, P. Podder, M. R. H. Mondal, S. S. Shafin and J. Kamruzzaman, "Applications of Machine Learning and Deep Learning in Antenna Design, Optimization, and Selection: A Review," in IEEE Access, vol. 11, pp. 103890-103915, 2023, doi: 10.1109/ACCESS.2023.3317371.

[15] S M Atikur Rahman, Iqtiar Md Siddique, Eric D Smith, “Analyzing bitcoin's decentralization: Coefficient of variation approach and 21 million divisibility”, Advancement of IoT in Blockchain Technology and its Applications, Vol. 2, Issue: 3, pp. 8-17, 2023.

[16] Datta A, Ng KF, Balakrishnan D, Ding M, Chee SW, Ban Y, Shi J, Loh ND. A data reduction and compression description for high throughput time-resolved electron microscopy. Nat Commun. 2021 Jan 28;12(1):664

[17] Ahmmed, S.; Podder, P.; Mondal, M.R.H.; Rahman, S.M.A.; Kannan, S.; Hasan, M.J.; Rohan, A.; Prosvirin, A.E. Enhancing Brain Tumor Classification with Transfer Learning across Multiple Classes: An In-Depth Analysis. BioMedInformatics 2023, 3, 1124-1144. https://doi.org/10.3390/biomedinformatics3040068

[18] Ehsan Bazgir, Ehteshamul Haque, Numair Bin Sharif, Md. Faysal Ahmed, “Security Aspects in IoT Based Cloud Computing”, World Journal of Advanced Research and Reviews, 2023.