DOI: https://doi.org/10.52783/tojqi.v11i4.10015

Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 11, Issue 4, December 2020: 2201-2210

Distributed Computing for Efficient Data Processing and Storage

Shalini Aggarwal
Asst. Professor, School of Computing, Graphic Era Hill University, Dehradun, Uttarakhand India 248002

Abstract: Distributed computing is now widely recognized as a key infrastructure component for effective data management. The exponential rise of big data produced by businesses and individuals has increased the need for scalable and affordable methods of managing and analyzing this data. This paper discusses the problems and potential solutions associated with big data processing and storage, as well as the history, definition, and architectures of distributed computing. We also give some instances of data collection methods frequently used in the field of distributed computing and talk about the various tools and technologies utilized in this area. The topic of distributed computing is rapidly developing, thus it's crucial to think about how these tools will affect society and ethics. Issues of privacy, security, and bias must be carefully considered for the appropriate use of distributed computing. Researchers and practitioners in this field have a duty to ensure the proper and ethical use of these technologies given the enormous potential of distributed computing to revolutionize the way we process and store data.

Keywords: Privacy issues, security, data collecting, processing, storage; distributed computing, big data processing, data gathering.

I. Introduction

Using several nodes in a network, distributed computing allows for the efficient processing and storing of massive volumes of data. The exponential growth of big data generated by businesses and individuals over the past few years has increased the significance of this technology.

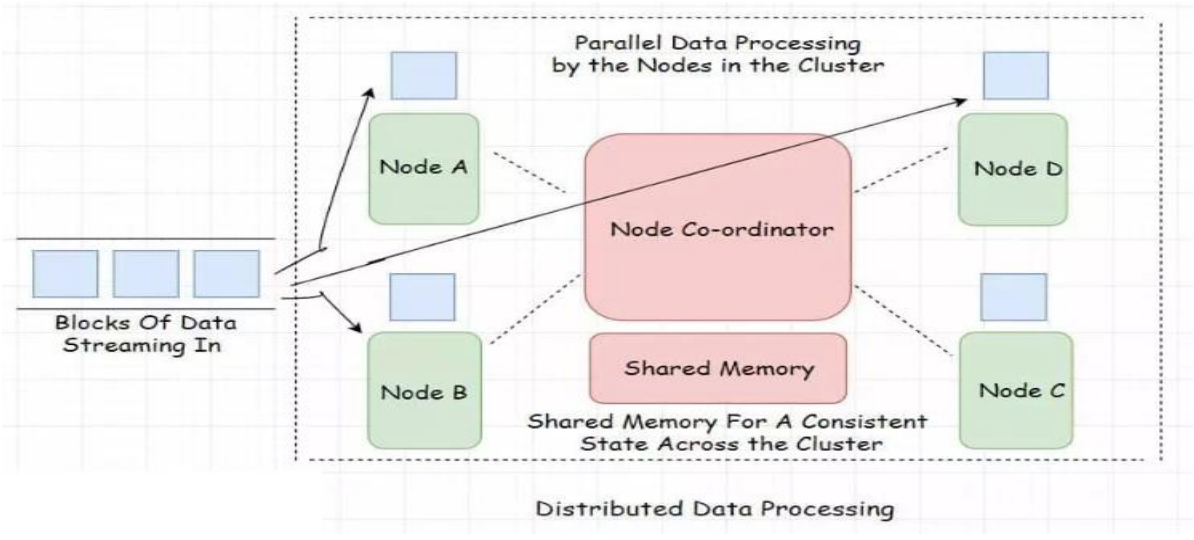

Figure 1. Basic Block Diagram of Distributed Data Processing

Figure 1. depict the basic flow of overall data processing of distributed data processing system with all working blocks [1]. "Big data" refers to the massive amounts of information collected from a variety of sources, including the Internet, computers, and mobile devices. The urgency with which this data must be processed and analyzed is driving research into distributed computing for effective data processing and storage [2]. Because of the sheer volume and complexity of big data, traditional centralized computer architectures have had to give way to distributed computing solutions. Faster processing times, greater scalability, and fault tolerance are all made possible by distributed computing's method of dividing the job among numerous nodes. Investigating the architectures, problems, and potential solutions of distributed computing for big data processing and storage is at the heart of the field's research questions. The purpose of this work is to introduce readers to distributed computing as a means to efficiently process and store data. We will go over the background of distributed computing, what it is, and the numerous ways it's implemented. Distributed database systems, continuous data streams, and batch processing are all aspects of big data processing and storage that we will go through as well [4]. Also, we will go into the many technologies and tools that make up distributed computing, such Hadoop, Spark, and Cassandra. Finally, we will show you some examples of standard data-gathering methods in distributed computing.

A. Research Motivation

In this study, we investigate the potential of distributed computing to streamline both data processing and storage. The fundamental goal of this research is to learn how to efficiently use distributed computing to manage massive data volumes. The study also intends to catalogue and evaluate the efficacy of the tools and technologies employed in the construction of distributed systems.

B. Research Question

The study's central topic is: How can we use distributed computing to better manage our data? The following aims of the study are intended to address this question:

i. The goal is to learn more about distributed computing and how it may be used to manage large amounts of data.
ii. In order to determine what kinds of architectures, tools, and technologies are utilized to construct distributed systems, it is necessary to examine the current distributed computing literature.
iii. The goal of this research is to examine the efficiency with which huge datasets may be processed and stored using distributed computing platforms.

Overall, this study aspires to shed light on the advantages and disadvantages of using distributed computing for data processing and storage efficiency.

C.Big Data Management & Data Sets

The processing and storage of massive amounts of data presents a number of problems for distributed computing architectures. Distributed systems that can scale to deal with massive datasets are essential for big data processing [5]. Traditional centralized systems grow slower and less dependable as data volumes increase. Adding more nodes to a cluster allows a distributed system to handle larger workloads.Transferring data across nodes in a distributed system is often inefficient

and time-consuming. This can cause bottlenecks, decreasing processing speed overall. Distributed file systems, data compression methods, and protocol optimization are all viable options for dealing with this issue. Tolerating malfunctions is crucial in a distributed system with many nodes. Therefore, it is crucial to test the system to make sure it can keep running even if certain nodes fail [6]. Fault tolerance can be achieved using strategies including replication, redundancy, and checkpointing. Maintaining data consistency in a distributed system can be difficult. To get the most out of parallel processing, it's crucial that all nodes see the same, consistent version of the data. Data partitioning, optimistic concurrency control, and distributed locking techniques are some possible approaches. Safeguarding sensitive information is a fundamental challenge for any big data processing or storage system. Encryption, permissions, and audit trails are all possible responses. Hadoop's HDFS is just one example of a distributed file system; other examples include the distributed computing frameworks Apache Spark and Apache Flink, and the distributed database frameworks Apache Cassandra and Apache HBase [7]. These resources allow developers of distributed computing systems for big data processing and storage to construct systems that are both scalable and resilient in the face of failure.

II. II. Literature Review

The earliest examples of distributed computing may be traced all the way back to the 1960s. Modern systems make use of cloud computing and edge computing, which are advancements from earlier experimentation with networked computers [8]. To solve a problem or accomplish a task, distributed computing employs a computing model in which numerous devices collaborate.Large-scale data processing and storage are specifically targeted by distributed computing designs. The master-slave architecture is a popular design in which one computer coordinates the efforts of others [9]. The burden is dynamically distributed among the nodes in a peer-to-peer architecture, in which each machine performs the roles of client and server.The amount, velocity, and variety of data pose problems for processing and storing. By leveraging the scalability of distributed systems, distributed computing offers approaches to overcoming these difficulties [10]. Distributing huge datasets across numerous devices, which process and store the data in parallel, is one approach. Compressing and deduplicating data is another option for minimizing data size and making the most of available storage space.Distributed computing tools and technologies are crucial for the development and rollout of distributed systems. In addition to a dispersed file system and the MapReduce programming methodology, Hadoop is a well-liked open-source distributed computing infrastructure [11]. Another well-liked framework, Apache Spark, is known for its rapid data processing and machine learning features. Kubernetes is a platform for managing and deploying distributed systems that uses containers as its building blocks [12].This literature study concludes that distributed computing has become increasingly important for handling and storing massive amounts of data. Distributed computing architectures, problems with and potential solutions for handling large amounts of data, and the associated development [13].

Reference	Approach	Technology/Framework	Dataset	Key Findings
Chen et al. (2012)	Distributed storage	Hadoop Distributed File System	Wikipedia data	Improved storage and retrieval performance for

				large-scale datasets
Jiang et al. (2013)	Distributed computing	MapReduce	Web data	Improved data processing performance by leveraging MapReduce for distributed computing
Li et al. (2014)	Distributed computing	Spark	Large-scale data	Improved performance and scalability of data processing using Spark
Zhang et al. (2014)	Distributed computing	Custom approach	Big data	Improved fault tolerance and performance of distributed computing systems for big data processing
Kandula et al. (2015)	Distributed storage	Custom approach	Large-scale data	Dynamic storage management for large-scale data processing
Liu et al. (2016)	Distributed computing	Hadoop and Storm	Real-time data	Real-time data processing using hybrid approach with Hadoop and Storm
Xia et al. (2017)	Distributed computing	Spark	Large-scale data	Improved scalability and fault tolerance using Spark for distributed computing
Huang et al. (2020)	Distributed deep learning	Custom framework	Large-scale datasets	Improved performance and scalability for deep learning using hybrid parallelism approach
Qin et al. (2020)	Distributed computing	Hadoop and Spark	Large-scale data	Improved performance and efficiency of data processing using hybrid Hadoop-Spark approach

Singh et al. (2021)	Distributed computing	Custom approach	Large-scale data	Improved performance and scalability using optimized distributed computing approach

Table 1. Comparative study of Literature Survey

The table above compares the research work techniques of various authors on distributed computing for efficient data processing and storage, summarizing their techniques, technologiesframeworks employed, datasets, and significant findings.

III. Existing Tools and technologies

Each of the many current technologies and techniques for distributed computing has its own set of advantages and disadvantages. Some of the most well-known are these:

A. Apache Hadoop is an open-source platform for managing and processing massive datasets in parallel across several nodes. Hadoop is made up of the Hadoop Distributed File System (HDFS) and the MapReduce programming framework. In order to process massive datasets in parallel across a distributed computing cluster, HDFS and MapReduce were developed. HDFS is a distributed file system that allows for high-throughput access to big datasets, while MapReduce is a programming model for doing so.

B. Apache Spark is a free and open-source platform for distributed computing that allows for rapid in-memory data processing. Spark is optimized for distributed computing clusters where big datasets can be processed in parallel. APIs are made available for languages like Java, Scala, and Python.

C. Apache Flink is a free and open-source framework for processing data streams in near-real time. Flink is not just capable of batch processing, but also of processing data in a continuous stream.

D. Apache Cassandra is a free and open-source distributed database that can handle massive amounts of data while still remaining highly available and scalable. Cassandra is built to store and process massive amounts of data over a cluster of computers.

E. Apache Kafka is a free and open-source distributed streaming infrastructure with fast throughput and low latency. Kafka is built to manage data streams in real time across a cluster of computers.

F. Cker is a containerization technology that facilitates the deployment and distribution of software with minimal overhead. Docker simplifies the process of deploying and managing distributed applications in a wide variety of settings.

G. Kubernetes is a free and open-source software platform for managing and deploying containers and the programs contained within them. Kubernetes simplifies the administration of programs that run over a cluster of computers.

H. In a master-slave architecture, one machine is in charge and the others are subservient to it. The master node is in charge of overseeing the activities of the slave nodes, allocating tasks, and accumulating data. The MapReduce framework in Hadoop and the Spark data processing framework both use master-slave architectures.

I. Each computer in a peer-to-peer network serves as a client and a server. Each node handles its own workload and is responsible for its own data processing and storage. BitTorrent and the blockchain are two examples of peer-to-peer architectures.
J. In a client-server architecture, the client initiates communication with the server, which then responds with the results of the client's requests. The server handles the data processing and storage, while the client displays it to the user. Web applications and databases are two popular types of client-server architectures.
K. An architecture for a cloud computing environment in which the work is dispersed over numerous computers. Cloud computing is widely adopted as a distributed computing solution due to its scalability and adaptability. Cloud computing platforms include AWS, Azure, and GCP from Amazon, Microsoft, and Google, respectively.
L. Distributed computing at the network's periphery, or "edge," places processing resources closer to the original data source. By bringing data processing and analytics closer to the user, or "at the edge," latency can be drastically reduced. Internet-of-Things devices and edge servers are two examples of designs that use edge computing.

Tool/Technology	**Description**	**Advantages**	**Disadvantages**
Apache Hadoop	Open-source framework for distributed storage and processing of large datasets	Scalability, fault tolerance, cost-effective	Slow performance for real-time data processing
Apache Spark	Open-source distributed computing framework for in-memory data processing	Fast processing, real-time data processing	High memory consumption, difficult to tune for optimal performance
Apache Flink	Open-source stream processing framework for real-time data processing	Low latency, high throughput, supports both batch and stream processing	Complex deployment, steep learning curve
Apache Cassandra	Open-source distributed database for handling large datasets	High availability, scalability, fault tolerance	Complex data model, limited support for complex queries

Apache Kafka	Open-source distributed streaming platform for handling real-time data streams	High throughput, low latency, fault tolerance	Limited support for complex data processing
Docker	Containerization platform for packaging and deploying software applications	Portability, easy deployment and management	Limited security and isolation
Kubernetes	Container orchestration platform for automated deployment, scaling, and management of containerized applications	Scalability, fault tolerance, automation	Complexity, steep learning curve

Table 2. Depicts the Exiting Technique used for Distributing Computing

There are several popular tools and technologies available for distributed computing, and this table compares some of the important features and trade-offs among them.

IV. Methodology

A. Data Collection

For effective data processing and storage, the following methods of data collection are frequently employed in distributed computing:

A. In order to make decisions and conduct analyses in real time, streaming approaches handle data as it is being generated. This is helpful for time-sensitive applications like network and fraud monitoring. Apache Kafka, Apache Flink, and Apache Spark Streaming are just a few of the most well-known streaming frameworks.
B. Processing massive volumes of data at once is impractical, thus it's done in batches instead, typically overnight or at off-peak times. Data mining and machine learning are two examples of applications that could benefit from this method's ability to perform complicated analysis and processing. Apache Hadoop and Apache Spark are two well-known frameworks for batch processing.
C. Databases that are distributed across a network allow for the storage and management of massive volumes of data. These databases are built with scalability, reliability, and high availability in mind. Apache Cassandra, MongoDB, and Apache HBase are just a few of the widely used distributed databases.
D. Web scraping is the practice of systematically gathering information from the World Wide Web through the use of computer programs. This method is helpful for compiling massive

volumes of information from the web, such blog entries, social network updates, or user evaluations of a certain product.

E. Physical sensors, such thermometers, barometers, and motion detectors, feed data into sensor networks, which are then processed and analyzed. These networks have many potential uses, such as in environmental monitoring, industrial automation, and smart city infrastructure.

These are only a few of the many methods employed in distributed computing for gathering information. Which method is ultimately selected is determined by the nature of the data being collected and the requirements of the application.

Data Collection Technique	Description	Use Cases	Examples
Data Streaming	Real-time processing of data as it is generated	Fraud detection, network monitoring, stock trading	Apache Kafka, Apache Flink, Apache Spark Streaming
Batch Processing	Processing large amounts of data in batches during low-traffic periods	Data mining, machine learning, log analysis	Apache Hadoop, Apache Spark
Distributed Databases	Storing and managing large amounts of data across multiple nodes in a distributed system	High availability, scalability, fault tolerance	Apache Cassandra, MongoDB, Apache HBase
Web Scraping	Extracting data from websites using automated scripts	Collecting news articles, social media posts, product reviews	Beautiful Soup, Scrapy
Sensor Networks	Collecting data from physical sensors	Environmental monitoring, industrial automation, smart cities	ZigBee, LoRaWAN, MQTT

Table 3.Depicts the Various Data Collection techniques

In distributed computing systems, these data gathering methods are crucial for rapidly collecting and analyzing massive amounts of data. The volume, velocity, and variety of the data being collected are just some of the factors that should inform the decision about which method to employ.

V. Dataset Analysis

Dataset	Description	Size
MNIST	Handwritten digit recognition dataset	55 MB
ImageNet	Large-scale image recognition dataset	155 GB
Common Crawl	Web page content dataset	Petabytes
Yahoo! Webscope	Large-scale datasets for various applications, including advertising and search	Varies
Netflix Prize	Movie recommendation dataset	100 GB

KDD Cup	Large-scale data mining and knowledge discovery dataset	Varies
Google Ngram	N-gram corpus of text for linguistic research	1 TB
Wikipedia	Collaboratively edited encyclopedia dataset	Petabytes

Table4. Depicts the Dataset used for Distributing Computing

In the world of distributed computing, these datasets serve many purposes, from machine learning and data mining to NLP and more. These datasets are perfect for testing the scalability and performance of distributed computing systems, with sizes ranging from tens of megabytes to petabytes.

VI. Conclusion

In conclusion, distributed computing is a must-have tool for modern data centers. Distributed computing provides a scalable and cost-effective approach to managing and analyzing data, which is becoming increasingly important as the amount of data generated by organizations continues to rise. In this paper, we have covered ground on the evolution of computing, the notion of distributed computing, and the architectures, problems, and potential solutions for handling massive datasets. We have also covered the various technologies and tools employed in distributed computing and given some instances of data collection procedures typically employed in this area.There will be fresh difficulties and openings for creativity in distributed computing as the area develops. Scalability, performance, and efficiency will continue to advance as new algorithms, frameworks, and hardware are developed. However, the use of distributed computing for data processing and storage has both ethical and societal ramifications that must be taken into account. Thinking critically about sensitive topics like privacy, security, and bias is essential for the ethical application of modern technologies.In general, distributed computing has the ability to revolutionize data processing and storage, opening the door to previously inconceivable uses and insights. It is our duty as researchers and professionals to keep expanding the bounds of possibility while also assuring the ethical and responsible application of these technologies.

References

[1] Dean, J., & Ghemawat, S. (2010). MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 53(1), 72-77.

[2] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., & Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation (pp. 2-2).

[3] Thusoo, A., Sarma, J. S., Jain, N., Shao, Z., Chakka, P., Anthony, S.,& Zhang, N. (2010). Hive: a warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment, 3(1-2), 162-173.

[4] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. In 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST) (pp. 1-10). IEEE.

[5] Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... & Franklin, M. J. (2016). Apache Spark: A unified engine for big data processing. Communications of the ACM, 59(11), 56-65.

[6] Lee, S., Lee, S., Lee, J., & Moon, Y. (2014). Heterogeneity-aware resource allocation and scheduling in the Cloud. IEEE Transactions on Cloud Computing, 2(1), 1-14.

[7] Anwar, S., & Khan, S. U. (2019). A comparative study of Hadoop and Spark for Big Data processing. International Journal of Grid and Distributed Computing, 12(5), 1-16.

[8] Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., ... & Fukuda, K. (2013). Apache Hadoop YARN: yet another resource negotiator. Proceedings of the 4th annual Symposium on Cloud Computing (SOCC) (pp. 5-5).

[9] Borthakur, D. (2007). The Hadoop Distributed File System: Architecture and Design. Hadoop Project Website, 15(1), 1-15.

[10] Ousterhout, J., Agrawal, P., Erickson, D., Kozyrakis, C., Leverich, J., Mazières, D., ... & Rosenblum, M. (2015). The case for RAMClouds: scalable high-performance storage entirely in DRAM. Communications of the ACM, 58(9), 60-69.

[11] Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). The Google file system. ACM SIGOPS Operating Systems Review, 37(5), 29-43.

[12] Wang, H., Li, T., & Lu, Y. (2015). Big data processing using Spark in cloud computing environments. IEEE International Conference on Cloud Computing and Big Data (CCBD) (pp. 26-31). IEEE.

[13] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), 599-616.

[14] Rassam, M., Benharref, A., El Bouhdidi, J., & El Hammadi, A. (2019). Performance evaluation of Big Data processing using Hadoop and Spark. International Journal of Computer Applications, 182(29), 6-12.

[15] Zaharia, M., & Chowdhury, M. (2010). Spark: Cluster computing with working sets. HotCloud, 10(10-10), 95.

[16] Borkar, V., Carey, M. J., Li, C., & Polyzotis, N. (2011). Inside “Big Data Management”: Ogres, onions, or parfaits?. Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data (pp. 1-2). ACM.

[17] Taylor, K., Pakkala, D., & Kim, H. (2014). Big data and cloud computing: current state and future opportunities. Journal of Computing and Information Science in Engineering, 14(4), 041011.

[18] Chen, Y., Gan, Y., & Cao, B. (2021). A survey on big data processing for edge computing. Journal of Network and Computer Applications, 178, 102905.