ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

Impact Factor: 5.621| | A Monthly Peer Reviewed & Referred Journal |

Website: www.ijareeie.com

Vol. 6, Issue 6, June 2017

Scalable and Efficient Approaches to Big Data Processing in Cloud Environments

MS. SONAL BORDIA JAIN

ASSOCIATE PROFESSOR, DEPT. OF COMPUTER SCIENCE, SS JAIN SUBODH PG COLLEGE, JAIPUR, RAJASTHAN, INDIA

ABSTRACT: In today's world, the amount of data generated by businesses and individuals is growing at an exponential rate. This has created a need for efficient and scalable big data processing algorithms that can handle massive volumes of data in a timely and cost-effective manner. One solution to this problem is to use cloud computing environments, which can provide the necessary computational resources and scalability to process large amounts of data. Developing scalable and efficient algorithms for big data processing in cloud environments is a complex task that requires a deep understanding of distributed systems, algorithms, and data structures. The goal is to develop algorithms that can efficiently process large amounts of data in a distributed environment, while minimizing the use of computational resources and minimizing the time required to complete the processing tasks.

KEYWORDS: scalable, big, data, cloud, environments, efficient, alogrithms

I. INTRODUCTION

Developing scalable and efficient algorithms for big data processing in cloud environments In today's world, the amount of data generated by businesses and individuals is growing at an exponential rate. This has created a need for efficient and scalable big data processing algorithms that can handle massive volumes of data in a timely and cost-effective manner. One solution to this problem is to use cloud computing environments, which can provide the necessary computational resources and scalability to process large amounts of data. Developing scalable and efficient algorithms for big data processing in cloud environments is a complex task that requires a deep understanding of distributed systems, algorithms, and data structures. The goal is to develop algorithms that can efficiently process large amounts of data in a distributed environment, while minimizing the use of computational resources and minimizing the time required to complete the processing tasks.[1,2,3] One of the key challenges in developing such algorithms is to distribute the data processing tasks across multiple nodes in a cloud environment in an optimal manner. The optimal distribution of tasks depends on the characteristics of the data, the available computational resources, and the network bandwidth. In addition, the distribution of tasks should be load-balanced to ensure that no single node is overloaded, which can lead to a slowdown in the overall processing performance. Another challenge is to optimize the data storage and retrieval operations in a cloud environment. Cloud environments typically use distributed file systems, such as Hadoop Distributed File System (HDFS) or Amazon S3, to store and manage large volumes of data. Efficient algorithms for data storage and retrieval should minimize the number of network hops required to access the data, and should take advantage of the parallel processing capabilities of the distributed file system. To develop efficient and scalable big data processing algorithms for cloud environments, researchers and practitioners are exploring a range of techniques and approaches. These include: MapReduce: MapReduce is a programming model and algorithm for processing large data sets in a distributed environment. The MapReduce model divides the processing tasks into two stages: the map stage and the reduce stage. The map stage processes the input data and produces a set of key-value pairs, and the reduce stage processes the output of the map stage and produces a final set of key-value pairs. MapReduce has been used in many big data processing systems, such as Apache Hadoop and Amazon Elastic MapReduce (EMR). Distributed Machine Learning: Machine learning algorithms can be computationally expensive and can require large amounts of data for training. In a cloud environment, distributed machine learning algorithms can be used to distribute the training tasks across multiple nodes, reducing the time required to train the models. Distributed

DOI:10.15662/IJAREEIE.2017.0606152

machine learning algorithms can also be used to perform real-time analysis of streaming data, such as sensor data or social media data.

II. DISCUSSION

Graph Processing: Graph processing algorithms are used to analyze large graphs, such as social networks or web graphs. Graph processing algorithms can be computationally expensive and can require large amounts of memory. In a cloud environment, distributed graph processing algorithms can be used to distribute the processing tasks across multiple nodes, reducing the time required to process the graphs. Developing scalable and efficient algorithms for big data processing in cloud environments is a critical area of research in computer science. With the growth of big data, there is an increasing need for algorithms that can process large amounts of data in a timely and cost-effective manner. By leveraging the power of cloud computing environments, researchers and practitioners can develop algorithms that can scale to handle massive amounts of data, while minimizing the use of computational resources and minimizing the time required to complete the processing tasks

The last decade has been characterised by an exponential growth of digital data production. This trend is particularly strong in scientific computing. For example, in the biological, medical, astronomic and earth science fields, very large data sets are produced every day from the observation or simulation of complex phenomena. At the same time, new massive sources of digital data have emerged. These include social media platforms such as Facebook, Instagram, and Twitter which are credited among the most important sources of data production in Internet. This Big Data is hard to process on conventional computing technologies and demands for parallel and distributed processing, which can be effectively provided by Cloud computing systems and services. This special issue focuses on the use and modelling of Clouds as scalable platforms for addressing the computational and data storage needs of the Big Data applications that are being developed nowadays.[4,5,6]

In the first paper [Citation1], Belcastro et al. address the main issues in the area of programming models and systems for Big Data analysis, which are extensively used in Cloud environments. As a first contribution, the most popular programming models for Big Data analysis (MapReduce, Directed Acyclic Graph, Message Passing, Bulk Synchronous Parallel, Workflow and SQL-like) are presented and discussed. Then, the paper analyses and compares the features of the main systems implementing these models, with the aim of helping developers identifying and selecting the best solution according to their skills, hardware availability, and application needs. Specifically, the systems are compared according to four criteria: (i) level of abstraction, which refers the programming capabilities of hiding low-level details of a system; (ii) type of parallelism, which describes the way in which a system allows to express parallel operations; (iii) infrastructure scale, which refers to the capability of a system to efficiently execute applications taking advantage from the infrastr ucture size; and (iv) classes of applications, which describes the most common application domain of a system.

The second paper [Citation2], by Ristov et al., focuses on the accurate scalability modelling of Cloud elastic services. The speedup and efficiency parameters provide important information about performance of a computer system with scaled resources compared with a computer system with a single processor. However, as Cloud elastic services' load is variable, it is also vital to analyse the load in order to determine which system is more effective and efficient. The paper argues that both the speedup and efficiency are not sufficient enough for proper modelling of Cloud elastic services, as the assumptions for both the speedup and efficiency are that the system's resources are scaled, while the load is constant. Accordingly, the paper defines two additional scaled systems by (i) scaling the load and (ii) scaling both the load and resources. A model is introduced to determine the efficiency for each scaled system, which can be used to compare the efficiencies of all scaled systems, regardless if they are scaled in terms of load or resources. An evaluation of the model by using Microsoft Azure is presented to confirm experimentally the theoretical analysis.

In the third paper [Citation3], Altomare et al. present a data mining approach to improve consolidation of virtual machines in Cloud systems. Consolidation of virtual machines is one of the most used and well-studied strategies to reduce the energy consumption in large data centres. It has the goal of allocating virtual machines on as few physical

DOI:10.15662/IJAREEIE.2017.0606152

servers as possible, while satisfying the Service Level Agreement established with users. Nevertheless, the effectiveness of a consolidation strategy strongly depends on forecasting the resource needs of virtual machines, which can be made using predictive data mining models. According to this approach, the paper presents the design and development of a system for energy-aware allocation of virtual machines, driven by predictive data mining models. In particular, migrations are driven by the forecast of the future computational needs (CPU, RAM) of each virtual machine, in order to efficiently allocate those on the available servers. An experimental evaluation, based on real-world Cloud data traces, demonstrates the benefit deriving from the use of a predictive data mining approach in terms of energy saving.

The last paper [Citation4], by Bendechache et al., presents a parallel and distributed clustering approach to analyze spatial datasets, which is designed to run on Cloud platforms using the MapReduce model. The application of clustering techniques to very large spatial datasets presents numerous challenges such as high-dimensionality, heterogeneity, and high complexity of some algorithms. The paper describes the design and implementation of a Dynamic Parallel and Distributed Clustering (DPDC) approach that can analyse Big Data within a reasonable response time and produce accurate results, by using high-performance computing and storage infrastructure, such as that provided by Cloud systems. The DPDC approach consists of two phases: a fully parallel phase that generates local clusters, and a phase that aggregates the local results to obtain global clusters. The aggregation phase is designed in such a way that the final clusters are compact and accurate while the overall process is efficient in time and memory allocation. DPDC was thoroughly tested and compared to existing clustering algorithms. The experiments show that the approach produces high-quality results and scales up very well by taking advantage of the MapReduce paradigm.

III. RESULTS

The large amount of data produced by satellites and airborne remote sensing instruments has posed important challenges to efficient and scalable processing of remotely sensed data in the context of various applications. In this paper, we propose a new big data framework for processing massive amounts of remote sensing images on cloud computing platforms.[7,8,9] In addition to taking advantage of the parallel processing abilities of cloud computing to cope with large-scale remote sensing data, this framework incorporates task scheduling strategy to further exploit the parallelism during the distributed processing stage. Using a computation- and data-intensive pan-sharpening method as a study case, the proposed approach starts by profiling a remote sensing application and characterizing it into a directed acyclic graph (DAG). With the obtained DAG representing the application, we further develop an optimization framework that incorporates the distributed computing mechanism and task scheduling strategy to minimize the total execution time. By determining an optimized solution of task partitioning and task assignments, high utilization of cloud computing resources and accordingly a significant speedup can be achieved for remote sensing data processing. Experimental results demonstrate that the proposed framework achieves promising results in terms of execution time as compared with the traditional (serial) processing approach. Our results also show that the proposed approach is scalable with regard to the increasing scale of remote sensing data.[10,11,12]

IV. CONCLUSION

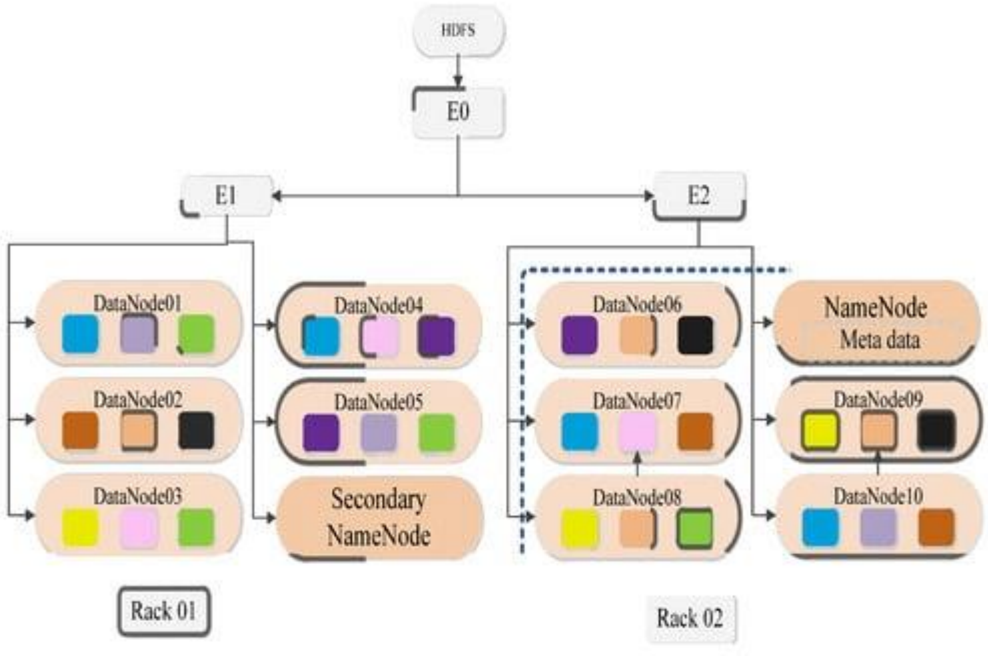
A smart grid is a heterogeneous and complex environment containing different kinds of devices, networks, systems, and data. IEC 61970 and IEC 61850 were discovered via a study on the principal Smart Grid standards and are open-source platforms based on cloud computing. IEC 61970 specifies the application program interface (API), while IEC 61850 specifies the abstract communication services interface (ACSI) .

Compatibility with the IEC61970 and IEC61850 standards for cloud computing technologies such as Hadoop, Spark, and Storm involves careful analysis of existing systems, translation of data, adoption of standards-compliant data formats, integration of systems, and thorough testing to ensure that the systems are functioning as intended. Because these two standards defined data models and the interface separately, the models are not uniform, and seamless communication between the substation and the control center is not possible. Whereas IEC 61970 defines the power of an information model and is widely used in enterprise integration. IEC 61850 is restricted to data exchange within substation equipment. Research revealed that Hadoop might pool idle power system resources and provide “supercomputing capability" for the smart grid's data integration platform. The grid dispatch automation system's support platform and application software should be upgraded in accordance with the component interface specification (CIS) and Common Information Models (CIM) standards. The data is integrated using the IEC61970 standard and connected to the smart grid via a platform for data sharing . The main user of these standards will not be a person, but a computer and they have to be machine-readable. At the same time, they are very complex documents involving thousands of different items. The full series of IEC 61850 Standards is now available as a global package. They are issued with the available associated code components. The series includes no less than 35 documents, dealing with substation automation, DER integration or cyber security, to name but a few . With the development of cloud computing technology and the needs of big data processing, a number of new computing models and programming models have emerged, providing users with a basic platform for parallel programming in the cloud environment, and shielding users as much as possible from the bottom layer. The details are presented to the user through a higher-level abstract interface [13]

Hadoop Technology

Hadoop is a massive data processing system open sourced by the Apache Software Foundation. It includes many components for storage or processing such as the Hadoop Distributed File System (HDFS) and the MapReduce parallel computing model .

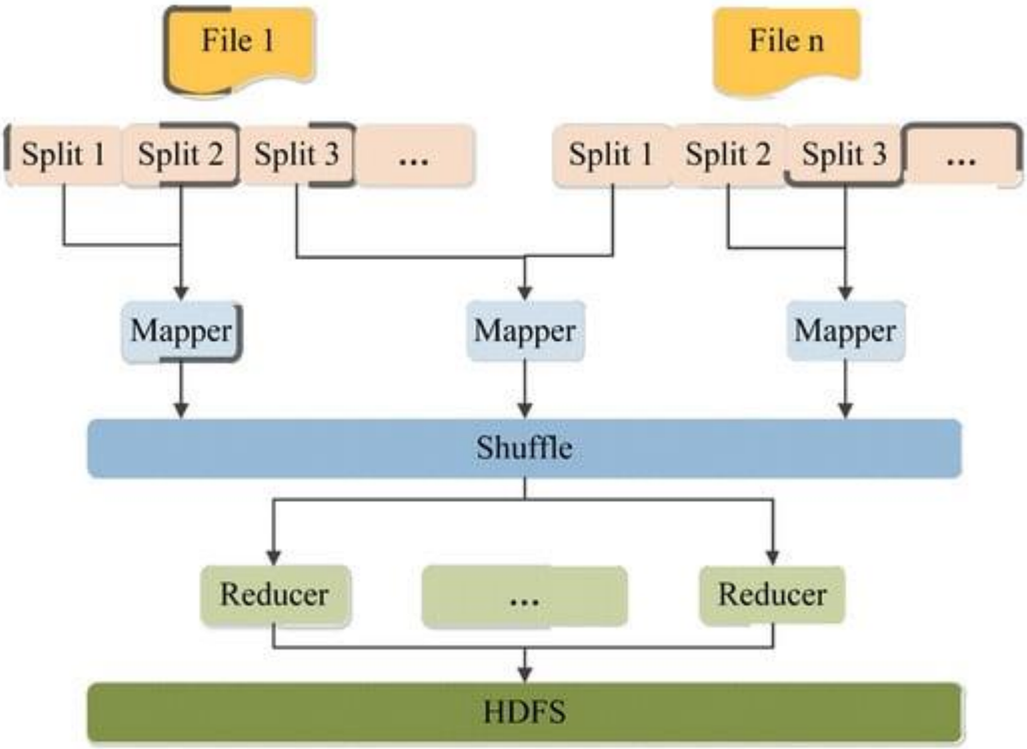
HDFS has the characteristics of distributed storage, high concurrent access, high fault tolerance, simple consistency and provides a reliable storage environment for parallel computing models .It adopts a master-slave architecture and is built on a physical cluster connected by multiple computers through a network. The bottom layer is the local file system of the operating system. Its architecture is shows HDFS includes a master node NameNode, a backup master node SecondaryNameNode and a set of DataNode slave nodes. The NameNode is responsible for managing the HDFS namespace, saving all metadata, and responding to client access requests. SecondaryNameNode is used to solve the single point of failure problem of Hadoop. The DataNode is responsible for the actual storage and management of redundant data blocks of files, and the data block files are actually stored on the local file system of each node



Architecture of HDFS.

DOI:10.15662/IJAREEIE.2017.0606152

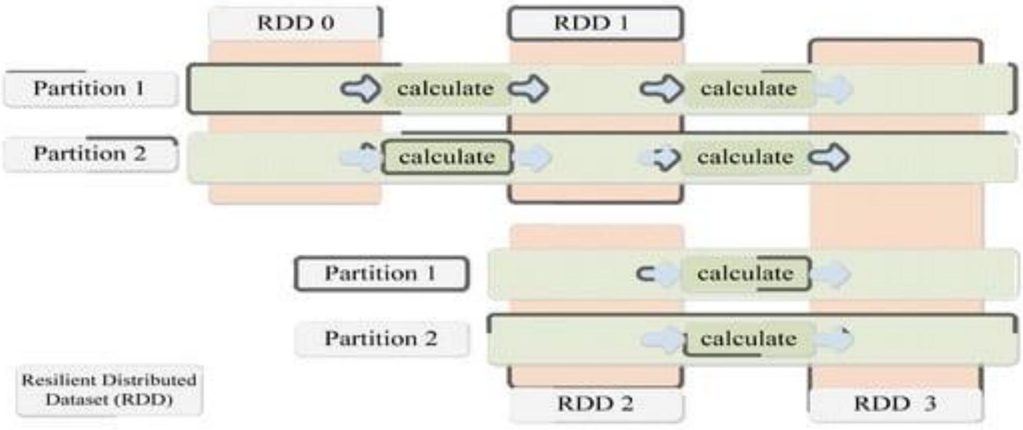
The basic idea of MapReduce is basically the same as that of traditional parallel computing models such as MPI, which is to "divide and conquer" a large amount of data. The system provides two simpler interfaces, Map and Reduce which automatically completes many underlying functions such as task division and scheduling, communication, load balancing, and failure recovery . shows the MapReduce programming mode.The input file is divided into several slices (InputSplit) according to a specific format, and converted into <key, value> key-value pairs, which are input to Mapper for calculation, and the intermediate results are key–value pairs. The collection is aggregated by the Reducer after a shuffle phase, and the result is saved to HDFS. Hadoop MapReduce is mainly oriented to the batch mode of massive static data, and its real-time performance is not high . The collection is aggregated by the Reducer after a shuffle phase and the result is saved to HDFS. Hadoop MapReduce is mainly oriented to the batch mode of massive static data, and its real-time performance is not high .Therefore, it too quickly processes and analyze the massive historical data accumulated in the condition monitoring of power equipment, hoping to discover valuable knowledge from it .



MapReduce programming model.

Spark Technology

Spark is a Hadoop MapReduce-like general-purpose parallel computing framework that appeared in 2012 [157]. The difference is that it puts data (including some intermediate data) in memory for calculation, avoiding a large amount of disk I/O caused by frequent reading and writing of HDFS during the calculation process. Therefore, spark is suitable for iterative and interactive computing scenarios, and even in general application scenarios which is more efficient than Hadoop MapReduce [158]. Spark's in-memory computing features from its core abstraction, Resilient Distributed Dataset (RDD) [159]. RDDs are read-only, fault-tolerant, distributed computing, partitionable, coarse-grained transformations, and in-memory storage. Each partition of the new RDD generated during Spark's calculation process has a dependency relationship with the partition of its parent RDD due to the calculation, which is called lineage. The lost RDD partition can be regenerated from the ancestor RDD by tracing the lineage, so as to implement fault tolerance. At the same time, Spark divides the entire computing process into multiple stages according to the different dependencies between RDDs, each stage generates a job, and creates tasks in units of RDD partitions and distributes them in multiple stages. Figure 11, shows the process in parallel on two computing nodes, and the Spark computing model [2]. The computing model is richer and more flexible than the single MapReduce model of Hadoop, and is compatible with various data sources such as HDFS, HBase, and Hive [4]. At present, Spark has been widely used in Internet companies such as Amazon, Yahoo, and Taobao [3], and it is still in the research stage in the power industry, and research still needs to be carried out in combination with typical application scenarios [163]. Based on Spark, is incapable of how the complex signal processing algorithm can perform fast calculation when the amount of data is large, which makes up for the application scenarios that Hadoop MapReduce [14]



The computing model based on resilient distributed dataset (RDD).

Storm Technology

Storm is primarily geared towards real-time analytics on large-scale, uninterrupted streams of data, unlike Hadoop which focuses on batch processing of massive amounts of data. Although Spark Streaming can also achieve the function of stream computing by decomposing batch jobs, its latency is longer than Storm . Storm also adopts a master-slave architecture and uses ZooKeeper to coordinate the entire cluster. Where the master node is called the control node and runs the Nimbus daemon, which is responsible for publishing topology programs, distributing tasks and monitoring cluster status .The slave node is called a worker node, running the Supervisor daemon, which is responsible for accepting the assigned tasks and starting the Java Virtual Machine (JVM) process worker to execute that the topology where Spout is the data entry of the topology, connecting to an external data source and converting the data into tuples that are sent to Bolt. The processing logic for tuples is encapsulated in Bolts, and after the processing is completed, tuples can be transmitted to subsequent Bolts The Spout and Bolt components are linked by a stream grouping strategy and can be configured as multiple instances to achieve parallel processing. Each instance will eventually form a task to be scheduled for execution

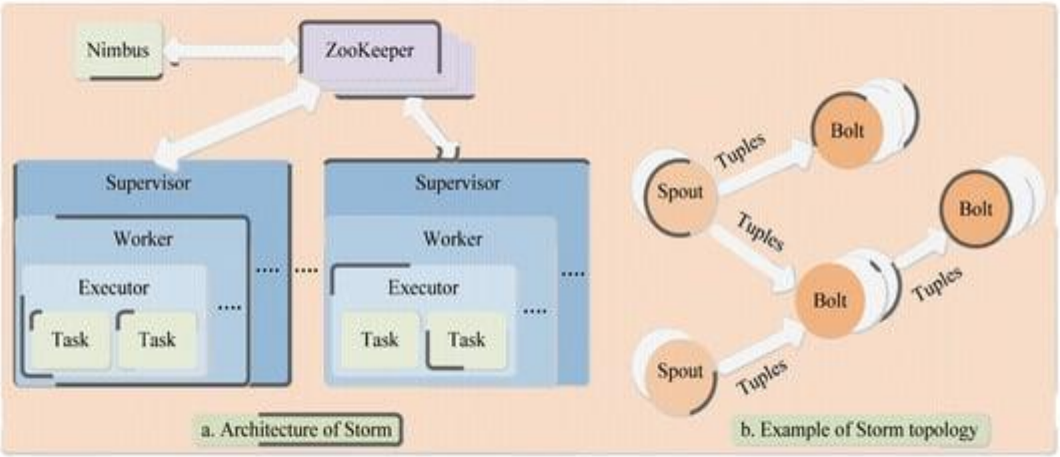


Figure 12. Architecture and topology of storm.

5. Comparison of Parallel Computing with Cloud Computing

Cloud computing is the fusion of many technologies, including parallel computing technology, so it is not equivalent to parallel computing, and its connotation richer [3]. The parallel computing usually refers to a specially designed parallel computer, while cloud computing combines multiple ordinary computers to achieve the purpose of improving computing performance [2]. In a broad sense, the computing technology it adopts also belongs to parallel computing. The technical point of parallel computing only focuses on computing and ignores data storage [3]. This is because traditional parallel programming models such as MPI are designed for high-performance computing, and small amount of data [1]. Cloud computing includes storage and computing, and the two cooperate with each other. For example, in the Hadoop system, data is stored in a distributed manner, and then the calculation is moved to the location of the data for execution, because mobile computing is more efficient than moving data [12]. From the perspective of applicable fields, parallel computing is suitable for the scientific computing field with high-performance computing requirements [2]. It is oriented to computing-intensive applications and requires users to have high professional quality in order to be able to deal with many low-level details [12]. The cloud computing provides services to users through three different levels, which is easier for users to use. The cloud computing is a key technology for big data processing and is suitable for data-intensive applications, but the large system management and maintenance costs make it not good at computing-intensive applications with a small amount of data [3]. To sum up, parallel computing and cloud computing technologies have a wide range of applications in various fields including the power industry [6]. The two are complementary rather than mutually exclusive, and each has different application scenarios [7].

Distributed Cloud Computing and Parallel Computing

A new method to connect data and applications supplied from several places is distributed cloud computing. A shared resource geographically dispersed among several users or systems is referred to as distributed. The ability to execute several jobs simultaneously is a key characteristic of cloud computing, which also aims to reduce CPU consumption, cut down on switching times, reduce waiting times for data processing, increase server throughput, and enhance data processing and communication speed. Another feature makes using any cloud application to communicate with users in various places simple for users. The final crucial component is enhancing server performance since communication performance is crucial.

The three different methods of parallel processing are distributed, shared, and hybrid memory systems. The important feature mentioned in more than one resource is improving performance using the load balancing technique through distributing the process and making a balance between servers for processing the jobs and improving the performance of our distributed system. Another feature is minimizing resource costs because when we divide the load among servers, we can minimize the resource cost such as CPU, memory, and storage. Since it is preferable to employ a system for handling user requests with a low response time, all of the references apply this principle by recommending a method for distributed parallel computing based on that factor. After looking through the references in this work, it was determined that was superior since it provides a wide range of characteristics, such as load balancing, enhancing system performance, and lowering reaction time and resource cost.[15]

REFERENCES

1. Ray, Partha Pratim (2016). "An Introduction to Dew Computing: Definition, Concept and Implications - IEEE Journals & Magazine". IEEE Access. 6: 723–737. doi:10.1109/ACCESS.2016.2775042. S2CID 3324933.
2. ^ Montazerolghaem, Ahmadreza; Yaghmaee, Mohammad Hossein; Leon-Garcia, Alberto (September 2016). "Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation". IEEE Transactions on Green Communications and Networking. 4 (3): 873–889. doi:10.1109/TGCN.2016.2982821. ISSN 2473-2400. S2CID 216188024. Archived from the original on 2016-12-09. Retrieved 2016-12-06.
3. ^ Wray, Jared (2014-02-27). "Where's The Rub: Cloud Computing's Hidden Costs". Forbes. Archived from the original on 2014-07-14. Retrieved 2014-07-14.
4. ^ Mell, Peter; Timothy Grance (September 2011). The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce. doi:10.6028/NIST.SP.800-145. Special publication 800-145.
5. ^ White, J. E. (1971). "Network Specifications for Remote Job Entry and Remote Job Output Retrieval at UCSB". tools.ietf.org. doi:10.17487/RFC0105. Archived from the original on 2016-03-30. Retrieved 2016-03-21.
6. ^ Levy, Steven (April 1994). "Bill and Andy's Excellent Adventure II" Archived 2015-10-02 at the Wayback Machine. Wired.
7. ^ Mosco, Vincent (2015). To the Cloud: Big Data in a Turbulent World. Taylor & Francis. p. 15. ISBN 9781317250388.
8. ^ "Announcing Amazon Elastic Compute Cloud (Amazon EC2) – beta". 24 August 2006. Archived from the original on 13 August 2014. Retrieved 31 May 2014.
9. ^ Qian, Ling; Lou, Zhigou; Du, Yujian; Gou, Leitao. "Cloud Computing: An Overview". Retrieved 19 April 2015.
10. ^ "Windows Azure General Availability". The Official Microsoft Blog. Microsoft. 2010-02-01. Archived from the original on 2014-05-11. Retrieved 2015-05-03.
11. ^ "Announcing General Availability of AWS Outposts". Amazon Web Services, Inc. Archived from the original on 2015-01-21. Retrieved 2015-02-04.
12. ^ "Remote work helps Zoom grow 169% in one year, posting $328.2M in Q1 revenue". TechCrunch. Archived from the original on 2015-01-17. Retrieved 2015-04-27.
13. ^ "What is Cloud Computing?". Amazon Web Services. 2013-03-19. Archived from the original on 2013-03-22. Retrieved 2013-03-20.
14. ^ Baburajan, Rajani (2011-08-24). "The Rising Cloud Storage Market Opportunity Strengthens Vendors". It.tmcnet.com. Archived from the original on 2012-06-17. Retrieved 2011-12-02.
15. ^ Oestreich, Ken (2010-11-15). "Converged Infrastructure". CTO Forum. Thectoforum.com. Archived from the original on 2012-01-13. Retrieved 2011-12-02.