Distributed Parallel and Cloud Computing: A Review

Amit Debnath
Department of Electrical and Computer Engineering
Lamar University
Beaumont, Texas, TX 77710, USA

Rahmanul Hoque
Department of Computer Science
North Dakota State University
Fargo, North Dakota, ND 58105, USA

Masum Billah
Department of Electrical and Computer Engineering
Lamar University
Beaumont, Texas, TX 77710, USA

Numair Bin Sharif
Department of Electrical and Computer Engineering
Lamar University
Beaumont, Texas, TX 77710, USA

Mahmudul Hoque
Department of Computer Science
Morgan State University,
Baltimore, Maryland 21251, USA

ABSTRACT
In this paper, a short review is presented on two prominent topics in this field, specifically distributed parallel processing and distributed cloud computing. The review paper examines various aspects, including the investigation of whether these topics have been addressed simultaneously in previous works. Additionally, the paper reviews the algorithms utilized in both distributed parallel computing and distributed cloud computing. The objective is to efficiently process tasks across resources and optimize calculation distribution among servers to enhance system performance. Throughout the review, articles are presented that discuss the design of applications in distributed cloud computing, as well as the concept of reducing response time in distributed parallel computing.

Keywords
Distributed systems, parallel computing, cloud computing, load balancing, virtualization.

1. INTRODUCTION
Cloud computing, which is a recently growing technology, incorporates scheduling mechanism as an integral component, enabling the processing of large volumes of data [1]. The field of high energy physics (HEP) increasingly embraces cloud computing, prompting us to investigate the reasons behind its integration in HEP applications and the gradual rise in its popularity [2].

Cloud computing is a broad term that generally refers to hosted services and involves the virtualization of physical hardware, with data being organized in designated centers. However, despite its numerous advantages as a new technology, it is not without challenges, with load balancing being a major issue [3].

This article proposes the adoption of robot SLAM architecture to meet the real-time requirement of practical robot systems, which is crucial. The robot SLAM employs two parallel threads for processing in order to fulfill this role. The computational complexity is primarily determined by the number of particles used, and two distributed threads with varying particle sizes are executed simultaneously [4].

Virtual clusters in cloud computing services enable the dynamic allocation of virtualized resources. The utilization of cloud environment by IT and business companies commonly involves user management and virtual storage. The effectiveness of the entire cloud services heavily relies on the proper organization and arrangement of job inputs. The selection of suitable properties for job implementation serves as the mechanism for executing the jobs [5].

The processing of large-scale industrial big data presents challenges in monitoring and modeling. To address this, a distributed and parallel designed principal component analysis approach is proposed. The initial step involves decomposing the large-scale process into distributed blocks based on prior knowledge. This approach effectively resolves the issue of high-dimensional process variables [6].

In a cloud system, some nodes may encounter heavy loads while others experience light loads [7], resulting in suboptimal performance. Load balancing, which plays a crucial role in cloud computing, aims to distribute the workload evenly among the nodes. This becomes a prominent issue in cloud environment [8]. Achieving even load distribution in the cloud system enhances resource utilization and is highly desirable [7].

To ensure good overall performance based on a specific performance metric, a load balancing algorithm [9] [10] [11] transparently transfers the workload from heavily loaded nodes to lightly loaded nodes. When considering performance in terms of the metric, the response time of the processes becomes the determining factor. However, from a resource perspective, the metric is the total system throughput [12]. Throughput is responsible for ensuring fair treatment of all users and their progress [9] [10] [12] [11] [8] [7], in contrast to response time [10].

2. DISTRIBUTED CLOUD COMPUTING
Scientific applications that make use of cloud computing have become an increasingly popular area of interest for researchers. At the same time, many large corporations are considering the possibility of transitioning to hybrid clouds, which combine the benefits of both public and private clouds. One of the key requirements for effectively executing complex applications in a cloud environment is parallel processing. By dividing tasks into smaller units that can be executed simultaneously, parallel processing allows for more efficient use of CPU resources. This is achieved through the presence of communication and synchronization mechanisms, which enable different processes to coordinate their actions and share information. By

maintaining the level of responsiveness of parallel jobs while also maximizing the utilization of computing nodes, data centers can ensure optimal performance and resource allocation.

Cloud computing offers clients the flexibility to request various types of information, shared resources, software, and services according to their specific needs and preferences. This on-demand nature of cloud computing has become increasingly prevalent across the internet, where the entire network can be seen as a vast cloud. Additionally, utilizing cloud services can lead to a reduction in both capital and operational costs for organizations. However, one of the main challenges in cloud computing is load balancing, ensuring that the workload is evenly distributed across the available resources. As the cloud environment is complex and its components are distributed across multiple locations, achieving efficient load balancing can be a daunting task. It is not practical or cost-effective to manually assign jobs to individual servers and clients, and maintaining idle services is not a viable solution. Moreover, there is always some level of uncertainty when it comes to job assignment in a dynamic cloud environment.

To address these challenges, a protocol has been proposed and designed with the aim of enhancing server throughput, performance, resource utilization, and switching time. This protocol focuses on improving the scheduling of jobs within the cloud environment and overcoming the limitations of existing protocols. In order to minimize waiting and switching time, priority is given to jobs that offer better performance to the computer. This approach requires significant effort in terms of managing job scheduling and optimizing server throughput and efficiency. By addressing the drawbacks of existing protocols and refining the scheduling process, the proposed protocol aims to enhance the overall performance of cloud-based applications and maximize the utilization of resources in a data center.

Scientific applications that employ systems built on cloud computing have the ability to achieve high throughput computing (HTC). In the domain of particle physics, these applications rely on systems that possess a unified infrastructure and make use of multiple distinct IaaS clouds. The foundation of our cloud computing system is based on several criteria. For the specific applications of embarrassingly parallel single HEP, it is necessary for them to execute within a batch environment that exists within the system. In this context, the need for inter-node or inter-process communications is non-existent. However, to mitigate the potential memory footprint that arises from shared process memory, the utilization of multi-process jobs serves to reduce this burden [2].

CIM parallel topological processing enables the realization of topology island formation and fast network topology processing, which serves as the foundation for the power network discussed in this paper. The authors have designed high throughput, high reliability, and high availability storages for the cloud storage platform. Additionally, they have introduced the design concept of the MySQL-CIM model and have successfully implemented an efficient tube MySQL-CIM model in data through Ogma development. The development of the power network topology processing (NTP) application has been completed [13].

Cloud computing, a term encompassing various hosted services, involves the virtualization of physical hardware and the organization of data in specific centers. Despite its numerous advantages, this new technology comes with its own set of issues. Among these challenges, load balancing stands out as a major concern and a crucial topic in cloud computing, which has been the focus of extensive research and investigation. Given the complex structure of data centers involving multiple systems, load balancing becomes a difficult task, particularly in the context of cloud computing. While the majority of studies on load balancing have been conducted in distributed environments, the exploration of semi-distributed load balancing has received little attention. Embracing a semi-distributed approach to load balancing could pave the way for the development of novel algorithms tailored for cloud computing [3].

The merits of decentralized computing away from data centers have been thoroughly discussed in the paper, along with the consideration of utilizing infrastructure obtained from multiple providers and modifying the cloud infrastructure. These emerging trends necessitate the development of a novel architecture for computing, which should be effectively implemented by the future cloud infrastructure. The domains that are expected to undergo the most significant transformations as a result of these new architectures encompass self-learning systems, service space, data intensive computing, and the interconnections between individuals and their devices. In order to fully unlock the potential of the forthcoming generation of cloud systems, a comprehensive roadmap outlining the various challenges that need to be taken into account has been put forth. This roadmap serves as a valuable guide to navigate the complexities and intricacies inherent in the establishment of a robust and efficient cloud infrastructure. By addressing these challenges, the cloud computing field can make significant advancements and propel itself towards a more sustainable and scalable future.

In this paper, the discussion revolves around creating ad hoc clouds and harnessing computing for online applications and mobile at the edge of the network. A computing model is presented to replace the traditional notion of paying for a cloud VM even if the server executing on the VM is idle. The uprising computing model of cloud integrates resilience and is software defined. Newly forming computing architecture and changing cloud infrastructure will influence certain areas. The connectivity between people and devices will be further enhanced by the internet-of-Things paradigm, and new architectures will play a crucial role. Data intensive computing poses a challenge due to the volume of data, requiring novel techniques to address it. There will also be rising interest in new services such as acceleration, containers, and function. The convergence of search areas with cloud systems will lead to the realization of self-learning systems. While the academia and industry are driving these changes, many challenges still need to be addressed in the future. The paper also discusses the development of next generation cloud computing, with a focus on sustainable systems, efficient management, expressive applications, and improved security.

Cloud computing encompasses grid computing, distributed computing, and parallel computing developments. The paper introduces two traditional parallel programming models and explains the concept of cloud computing. It further analyzes and studies the principles, advantages, and disadvantages of OpenMP, MPI, and Map Reduce. Finally, the paper discusses and compares MPI, OpenMP models, and Map Reduce from the perspective of cloud computing, aiming to provide a reference for the development of parallel computing.

This research paper places significant importance on the field of distributed computing, which involves the study and analysis of various aspects related to this domain. Specifically, it delves into the disparities between distributed and parallel computing, exploring their dissimilarities in terms of functionalities, characteristics, and underlying principles. Additionally, the paper investigates and elucidates the pertinent terminologies associated with distributed computing, providing a comprehensive understanding of key concepts such as task allocation and performance parameters. Furthermore, it comprehensively discusses the myriad advantages and extensive scope of distributed computing, shedding light on its potential applications and highlighting the various algorithm models employed in this context, including parallel distributed algorithm models, which play a pivotal role in achieving efficient and effective distributed computing systems.

3. DISTRIBUTED PARALLEL PROCESSING

The cloud system may experience a discrepancy in the load distribution across its nodes, leading to a considerable decrease in performance. Within a cloud environment, the responsibility of load balancing falls upon the task of distributing the workload among the nodes, which has become a prominent issue in the realm of cloud computing [7]. The quality of service provided by the system should adhere to certain standards to ensure the satisfaction of virtual users. To achieve this, various measures are implemented to optimize resource allocation and job arrangement, such as reducing source expenses, minimizing make span, and maintaining error acceptance alongside quality of service.

To evaluate the effectiveness of the proposed algorithms, the Cloud Sim toolkit has been utilized in conjunction with existing scheduling policies. Initial findings from experiments have demonstrated superior outcomes in terms of user response, execution time, cost, and time across a range of cloud workloads when compared to the preexisting algorithms. This paper delves into the discussion of different virtual machine scheduling algorithms and their performance, which are assessed based on a variety of quality metrics. The approach considers factors such as CPU performance, network consumption, scheduling success rate, and standard implementation time, among others.

Furthermore, this study has shed light on the utilization of caching within the network itself to process user requests. By employing parallel processing, a novel caching strategy has been introduced and thoroughly evaluated in terms of its effectiveness in reducing redundant traffic and data access delay in various caching scenarios. Moreover, the researchers have examined the cost implications of implementing the aforementioned caching network. The enhanced performance achieved through decreased delay comes at the expense of increased cost, necessitating a careful consideration of the trade-off when implementing the proposed parallel processing strategy. The simulation results have demonstrated the notable efficacy of this new strategy [17].

In an effort to enhance the real-time performance of the PF-based robot simultaneous localization and mapping (SLAM), this paper proposes an efficient parallel implementation. The discussed distributed parallel idea allows for an acceleration of the overall SLAM algorithm, as it leverages a large number of particles only when a keyframe laser is captured. Through our experimental endeavors, we have obtained results that unequivocally demonstrate the effective reduction of temporal cost through the utilization of this distributed architecture [4].

In their work, P. Srinivasa Rao et al. [18] emphasized the importance of adopting a well-balanced approach to achieving effective load balancing in a network. They argued that the information nodes of all other nodes must be taken into consideration in order to achieve this balance. When a node receives a job, it is required to query the status of the other nodes to determine which one has a lower usage and can therefore forward the work. However, when all the nodes engage in this query phenomenon, it can lead to an overload in the system. Furthermore, broadcasting a statement to inform all nodes about their status can also result in a significant load on the network.

Additionally, the time wasted at each node in performing the query for status information is another issue that needs to be addressed. Moreover, the current state of the network itself plays a crucial role in determining the performance load balancing. This is especially true in complex networks with multiple subnets, where configuring the network node to locate all other nodes becomes a complex task in itself. Consequently, the process of querying the status of nodes in the cloud can have a direct impact on the performance load balancing [18].

In a related study, reference [19] demonstrated that factors such as response time have a significant influence on the performance load balancing in cloud environments. The authors identified two key issues with previous algorithms: first, load balancing only occurs when the server is overloaded, which limits its effectiveness; and second, continuous information retrieval from available resources results in increased computational cost and bandwidth consumption. To address these issues, the authors proposed an algorithm that assigns the required decisions for servers based on the response time of the request. This approach effectively reduces the amount of query information on available resources, thereby minimizing communication and computation requirements on each server [19].

According to the prescribed algorithm, Min - Min [20] serves to minimize the time required to complete tasks at each network node. However, the algorithm fails to take into account the workload of each resource. To overcome this limitation, the authors propose the Load Balance Improved Min-Min (LBIMM) algorithm. Neglecting the workload of resources can result in some resources being overloaded while others remain idle. Consequently, the work performed on each resource becomes a determining factor in achieving load balancing on the cloud. The traditional Min - Min Algorithm serves as the basis for the current scheduling algorithm in cloud computing.

In [21], Kapur presents the LBRS (Load Balanced Resource Scheduling Algorithm) algorithm to address the significance of resource scheduling policies and load balancing for cloud resources. The primary objectives include maximizing CPU utilization, throughput, and fairness while minimizing response time, waiting time, and resource cost. QoS parameters such as throughput, response time, and waiting time are thoroughly examined through simulation and data analysis to assess their impact on load balancing in the cloud. The researchers discovered that the parameter of make span (runtime) holds great significance for the data center cloud. Therefore, the researchers focus on studying algorithms that effectively achieve load balancing to reduce the make span of virtual machines.

To ensure optimal resource utilization, load balancing is employed to distribute dynamic workloads across multiple resources, preventing any single resource from being underutilized or overwhelmed. This, however, poses a significant optimization problem. This paper proposes a load balancing strategy based on Simulated Annealing (SA), with a primary objective of balancing the load on the cloud infrastructure. The traditional Cloud Analyst simulator is modified to measure the effectiveness of this algorithm. When compared to existing approaches such as First Come First Serve (FCFS), Stochastic Hill Climbing (SHC), and Round Robin (RR), the proposed algorithm demonstrates superior overall performance [22].

Our focus is on maximizing resource utilization while ensuring the availability of working resources for future tasks and the reliability of cloud services. To achieve this, we introduce a new scheduling algorithm called the Dabbawala cloud scheduling Algorithm, inspired by the Mumbai Dabbawala delivery system. In this system, tasks are grouped based on the cost required to complete them within a cluster and its VM resources. The lowest cost cluster and its corresponding VM are identified for each task and grouped together, similar to the Hadoop Map Reduce model. The algorithm consists of four Dabbawalas assigned to each task for service. Several available scheduling algorithms are compared to this proposed algorithm, and significant improvements in time and resource utilization are achieved [23].

4. EXISTING LOAD BALANCING TECHNIQUE IN CLOUD

Load balancing is a crucial aspect of cloud computing infrastructure to ensure optimal resource utilization, maximize throughput, minimize response time, and avoid overloading any single server or resource

The VectorDot algorithm, proposed by A. Singh et al. [18], was an innovative approach for achieving load balancing. It leveraged a flexible data center that incorporated storage virtualization and integrated servers. This allowed it to effectively manage the multidimensional resource loads distributed across network switches, servers, and storages, as well as the hierarchical complexities of the data center. VectorDot not only helped alleviate overloads on storage nodes, switches, and servers but also distinguished nodes based on item requirements using dot product calculations.

J. Hu et al. [9] proposed a scheduling strategy for load balancing of VM resources that took into account the current state of the system and historical data. This strategy utilized a genetic algorithm to minimize dynamic migrations and achieve optimal load balancing. By addressing the challenges of migration costs and load imbalances, this approach enhanced resource utilization.

Y. Fang et al. [11] discussed a two-level task scheduling mechanism based on load balancing to maximize resource utilization and meet dynamic user requirements. This mechanism involved mapping tasks to virtual machines and then to host resources, effectively achieving load balancing. The result was a cloud computing environment that exhibited improved resource utilization, enhanced task response time, and overall performance gains.

M. Randles et al. [9] investigated the concept of active clustering as a self-aggregating technique for load balancing. This technique optimized job assignments by using local re-wiring to connect similar services. By effectively utilizing resources in high-resource systems, it increased throughput and enhanced system performance. However, the effectiveness of active clustering diminished as system diversity increased.

The following are various metrics considered in existing load balancing techniques in cloud computing:

1. Throughput measures the number of executed tasks and indicates system performance. Higher throughput values are indicative of better performance.

2. Overhead, as measured by the involvement of overhead associates, assesses the impact of load balancing algorithms. Overhead includes inter-process, inter-processor, and task mobilization components. More efficient load balancing techniques involve less overhead.

3. Fault toleration is an important aspect of load balancing. A good load balancing technique should be able to achieve uniform load balancing despite link or arbitrary node failures.

4. Migration times should be minimized in high-performance systems. Migration time refers to the time required for resource or job migration between individual nodes, and lower migration times are desirable.

5. Response time is another critical parameter that affects system performance. Minimizing response time refers to reducing the time it takes for a load balancing algorithm to respond in a distributed system.

6. It is imperative to ensure the efficient distribution of resources in order to achieve effective load balancing. The process of optimization must be carried out to attain this objective.

7. The ability of an algorithm to perform load balancing for any given number of nodes in a system determines its scalability. The desire is to enhance this scalability.

8. The performance of a system serves as a measure of its effectiveness. However, it is essential to also consider the cost effectiveness and maintain it at a reasonable level. For instance, acceptable delays should be maintained while reducing task response times.

5. DISCUSSIONS

Distributed cloud computing is an emerging technology that facilitates the interconnection of data and applications hosted in diverse locations. In the realm of information technology, the term 'distributed' signifies the sharing of resources among multiple users or systems that are geographically dispersed. As delineated in Table I, employing distributed cloud computing offers several advantageous features, each of which exerts a distinct impact on the utilization of cloud technology.

Among the notable features that span across multiple resources is the multi-process job feature, which plays a pivotal role in cloud computing as it enables the simultaneous execution of multiple tasks across multiple servers located in disparate regions. When confronted with a voluminous amount of data to process, it is possible to partition the data into smaller segments, whereby each segment can be processed by a separate server. The objective of this approach is to alleviate CPU utilization, minimize switching time, reduce waiting time for data processing, enhance server throughput, and optimize the communication and computation of data.

Another noteworthy feature is the design of applications tailored for cloud infrastructure, which facilitates seamless communication among users situated in diverse locations.

Additionally, there is a feature that pertains to the reduction of memory usage. One of the prevailing issues in the past was excessive memory consumption; however, with the advent of cloud technology, users can now dynamically allocate memory resources according to their needs by engaging with the cloud application manager. Consequently, users rely solely on cloud memory, thereby reducing their own storage requirements.

The final crucial feature revolves around enhancing server performance, as effective communication performance is paramount. After thoroughly examining the references outlined in Table I, we have determined that authors in [1] represents the most comprehensive work on distributed cloud computing, as it encompasses a multitude of features that have been discussed in detail above.

Parallel processing is concerned with improving the performance of digital computers and augmenting other attributes such as cost-effectiveness and reliability through the deployment of various forms of concurrency, achieved via diverse algorithmic and architectural approaches. There are three primary types of parallel processing methodologies: distributed, shared, and hybrid memory systems. In this review, our focus has been on distributed parallel processing, and we have identified several significant features, as illustrated in Table 2. The enhancement of system performance through the utilization of load balancing technique is a prominent aspect that is discussed in multiple sources. By employing load balancing across servers, it becomes possible to distribute the workload and establish equilibrium between the servers, thereby optimizing the processing of tasks and bolstering the performance of the distributed system. Additionally, another noteworthy feature pertains to the reduction of resource costs. By allocating the load among servers, it becomes feasible to minimize the expenditure of resources, such as CPU, memory, and storage. The concept is implemented across all references by proposing an algorithm that leverages distributed parallel processing based on the response time of user requests. It is imperative to prioritize systems with minimal response time, as they are more favorable for addressing user requests.

Cloud platforms can be used to develop and deploy sophisticated OEE analytics tools. These tools can analyze historical data, identify trends, and predict potential equipment issues before they occur. This allows for proactive maintenance and optimization strategies [26].

Distributed parallel and cloud computing technologies offer scalability, allowing healthcare systems to handle large volumes of data efficiently. With the increasing amount of medical data generated from various sources such as electronic health records (EHRs), medical imaging, and wearable devices, scalable computing infrastructure is essential for processing and analyzing this data in real-time [27-41].

Table 1: Distributed Cloud Computing Summary

Features	[1]	[2]	[3]	[13]	[14]	[15]	[16]
reduce CPU Utilization	✓						
Multi-process jobs		✓	✓	✓	✓	✓	
Minimizing switching time				✓			
Minimize waiting time				✓			
Improve resource utilization					✓		
Improve server performance			✓		✓		✓
Improve server throughput							✓
Load balancing						✓	
Cost efficiency							✓
Reduce memory usage		✓	✓				
Design an application for cloud structure			✓	✓	✓		
Improve connectivity between people			✓				
Enhancing security							✓
Managing efficiency and developing system						✓	
Expand the concept of cloud computing							✓
Advantages and disadvantages of MPI, OpenMPI, and MapReduce	✓						
Compare MPI, OpenMPI, and MapReduce							✓

Table 2: Distributed Cloud Computing Summary

Features	[4]	[7]	[17]	[18]	[19]	[20]	[21]	[22]	[23]
Improved Performance with Load Balancing	√	√	√	√	√	√	√	√	√
Querying Node Status	√								
Algorithm Based on Response Time			√	√	√				
Reduced Query Information on Available Resource				√	√				
Reduced Communication and Computation on Each Server					√				
Consideration of Workload for Each Resource					√				
Maximizing CPU Utilization						√			
Minimizing Throughput							√		
Minimizing Response Time								√	
Minimizing Waiting Time								√	
Minimizing Resource Cost						√	√	√	
Safeguarding Error Acceptance and QoS									√
Mapping Tasks and Reducing Mapped Tasks									√
Efficient Parallel Implementation									√
Methods to Enhance Job Arrangement									√
Enhancing Resource Allotment									√
Better Results in Execution Time									√

6. CONCLUSIONS

This comprehensive review has extensively delved into the various aspects of distributed cloud computing and distributed parallel computing. The discourse within this review paper has dedicated significant attention to the amalgamation of algorithms pertaining to these domains. The central objective of this scholarly work revolves around the efficient distribution of workloads across multiple servers, subsequently processing them through the master and slave nodes. Furthermore, the articles encompassed in this paper encompass a thorough exploration of the methodology involved in designing applications for distributed cloud computing, along with the introduction of a novel concept aimed at optimizing response times during the execution of user-generated images.

7. REFERENCES

[1] L. Tripathy and R. R. Patra "SCHEDULING IN CLOUD COMPUTING", in International Journal on Cloud Computing: Services and Architecture (IJCCSA), Volume: 4, No. 5, October 2014

[2] R. J. Sobie "Distributed Cloud Computing in High Energy Physics", in DCC '14 Proceedings of the 2014 ACM SIGCOMM workshop on Distributed cloud computing, Pages 17–22, 2014, Chicago, IL, USA, August 2014

[3] P. A. Pawade and Prof. V. T. Gaikwad "Semi-Distributed Cloud Computing System with Load Balancing Algorithm", 2014

[4] Xiuzhi Li, Songmin Jia, Ke Wang and Xiaolin Yin "DISTRIBUTED PARALLEL PROCESSING OF MOBILE ROBOT PF-SLAM", in International Conference on Automatic Control and Artificial Intelligence (ACAI 2012), Xiamen, China, April 2013.

[5] A. K.Indira, B. M. K. Devi "Effective Integrated Parallel Distributed Processing Approach in Optimized Multi-cloud computing Environment", in Sixth International Conference on Advanced Computing (lCoAC), pages 17-19, Chennai, India, Dec. 2014

[6] J. Zhu, Z. Ge and Z. Song "Distributed Parallel PCA for Modeling and Monitoring of Large-scale Plant-wide Processes with Big Data", in IEEE Transactions on Industrial Informatics, pages 1877 – 1885, Volume: 13, Issue: 4, Aug. 2017

[7] A. Khiyati, M. Zbakh, H. El Bakkali, D. El Kettani “Load Balancing Cloud Computing: State Of Art”, in 2012 National Days of Network Security and Systems, pages 20-21, Marrakech Morocco, April 2012

[8] M. A. Vouk, “Cloud Computing – Issues, Research and Implementations”, in ITI 2008 - 30th International Conference on Information Technology Interfaces, pages 23-26, Dubrovnik, Croatia, June 2008

[9] C. Lin, H. Chin, D. Deng, “Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System”, in IEEE Systems Journal , pages 225 – 234, Volume: 8, March 2014

[10] Y. Deng, Rynson W.H. Lau, “On Delay Adjustment for Dynamic Load Balancing in Distributed Virtual Environments”, in IEEE transaction on visualization and computer graphics, Volume: 18, NO. 4, April 2012

[11] L.D. D. Babua and P. V. Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments”, in Applied Soft Computing, Pages 2292-2303, Volume: 13, Issue 5, Amsterdam, The Netherlands, May 2013.

[12] D. Warneke and O. Kao, “Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud”, in IEEE transaction on parallel and distributed systems, Volume: 22, NO. 6, NJ, USA, JUNE 2011

[13] X. L. Xingong and X. Lv “Distributed Cloud Storage and Parallel Topology Processing of Power Network”, in Third International Conference on Trustworthy Systems and Their Applications, pages 18-22, Wuhan, China, Sept. 2016

[14] B. Varghese and R. Buyya “Next Generation Cloud Computing: New Trends and Research Directions1”, in Future Generation Computer Systems, 07 September 2017.

[15] Z. Peng, Q. Gong , Y. Duan and Y. Wang “The Research of the Parallel Computing Development from the Angle of Cloud Computing”, in IOP Conf. Series: Journal of Physics: Conf. Series 910, 2017

[16] Md. F. Ali and R. Zaman Khan “Distributed Computing: AnOverview”, in International Journal of Advanced Networking and Applications, Pages: 2630-2635, Volume: 07 Issue: 01, 2015

[17] Y. Sun, Z. Zhu and Z. Fan “Distributed Caching in Wireless Cellular Networks Incorporating Parallel Processing” , in IEEE Internet Computing, pages 52-61, Volume: 22, Issue: 1, Feb. 2018

[18] P. Srinivasa Rao, V.P.C Rao and A. Govardhan,“Dynamic Load Balancing With Central Monitoring of Distributed Job Processing System”, in International Journal of Computer Applications, Volume 65– No.21, March 2013.

[19] A. Sharma and S. K. Peddoju, “Response Time Based Load Balancing in Cloud Computing”, In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT). July 2014, Kanyakumari, India

[20] H. Chen, F. Wang, N. Helian, and G. Akanmu, “User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing”, in 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH), Bangalore, India, Feb 2013

[21] R. Kapur, “A Workload Balanced Approach for Resource Scheduling in Cloud Computing”, in 2015 Eighth International Conference on Contemporary Computing (IC3), page 20-22, Noida, India, Aug 2015

[22] B. Mondal and A. Choudhury “Simulated Annealing (SA) based Load Balancing Strategy for Cloud Computing”, in International Journal of Computer Science and Information Technologies, pages 3307-3312, Volume: 6, Kolkata-700053 India, 2015

[23] S.K.S. Kumar and P. Balasubramanie “Cloud Scheduling Using Mumbai Dabbawala”, in International Journal of Computer Science and Mobile Computing, Volume: 4, Issue. 10, October 2015

[24] N. J. Kansal and I. Chana “Existing Load balancing techniques in cloud computing: A SYSTEMATIC REVIEW”, in Journal of Information Systems and Communication, pages 87-91, Volume: 3, Issue 1, 2012.

[25] Molla, S., Bazgir, E., Mustaquim, S. M., Siddique, I. M., & Siddique, A. A. (2024). Uncovering COVID-19 conversations: Twitter insights and trends. World Journal of Advanced Research and Reviews, 21(1), 836-842.

[26] Md Rahamat Ullah, Selim Molla, Iqtiar Md Siddique, Anamika Ahmed Siddique, Md. Minhajul Abedin, “Optimizing Performance: A Deep Dive into Overall Equipment Effectiveness (OEE) for Operational Excellence”, Journal of Industrial Mechanics, Vol. 8, No. 3, 2023.

[27] Rahman, S. M., Ibtisum, S., Bazgir, E., & Barai, T. (2023). The significance of machine learning in clinical disease diagnosis: A review. arXiv preprint arXiv:2310.16978.

[28] Rahman, M. A., Bazgir, E., Hossain, S. S., & Maniruzzaman, M. (2024). Skin cancer classification using NASNet. International Journal of Science and Research Archive, 11(1), 775-785.

[29] Ehsan Bazgir, Ehteshamul Haque, Md. Maniruzzaman and Rahmanul Hoque, “Skin cancer classification using Inception Network”, World Journal of Advanced Research and Reviews, 2024, 21(02), 839–849.

[30] 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.

[31] Bharati, S., Rahman, M. A., & Podder, P. (2018, September). Breast cancer prediction applying different classification algorithm with comparative analysis using WEKA. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT) (pp. 581-584). IEEE.

[32] Rahmanul Hoque, Suman Das, Mahmudul Hoque and Ehteshamul Haque, “ Breast Cancer Classification using XGBoost”, World Journal of Advanced Research and Reviews, 2024, 21(02), 1985–1994

[33] Ibtisum, S., Rahman, S. A., & Hossain, S. S. (2023). Comparative analysis of MapReduce and Apache Tez Performance in Multinode clusters with data compression.

World Journal of Advanced Research and Reviews, 20(3), 519-526.

[34] Rahman, S. M., Ibtisum, S., Podder, P., & Hossain, S. M. (2023). Progression and challenges of IoT in healthcare: A short review. arXiv preprint arXiv:2311.12869.

[35] Bharati, S., Rahman, M. A., Podder, P., Robel, M. R. A., & Gandhi, N. (2021). Comparative performance analysis of neural network base training algorithm and neuro-fuzzy system with SOM for the purpose of prediction of the features of superconductors. In Intelligent Systems Design and Applications: 19th International Conference on Intelligent Systems Design and Applications (ISDA 2019) held December 3-5, 2019 19 (pp. 69-79). Springer International Publishing.

[36] Podder, P., Bharati, S., Mondal, M. R. H., & Khamparia, A. (2022). Rethinking the transfer learning architecture for respiratory diseases and COVID-19 diagnosis. In Biomedical data analysis and processing using explainable (XAI) and responsive artificial intelligence (RAI) (pp. 105-121). Singapore: Springer Singapore.

[37] Begum, A. M., Mondal, M. R. H., Podder, P., & Bharati, S. (2021, December). Detecting Spinal Abnormalities using Multilayer Perceptron Algorithm. In International Conference on Innovations in Bio-Inspired Computing and Applications (pp. 654-664). Cham: Springer International Publishing.

[38] Rahmanul Hoque, Md. Maniruzzaman, Daniel Lucky Michael and Mahmudul Hoque, “Empowering blockchain with SmartNIC: Enhancing performance, security, and scalability”, World Journal of Advanced Research and Reviews, 2024, 22(01), 151–162.

[39] Rahmanul Hoque, Masum Billah, Amit Debnath, S. M. Saokat Hossain and Numair Bin Sharif, “Heart Disease Prediction using SVM”, International Journal of Science and Research Archive, 2024, 11(02), 412–420.

[40] Mohammad Fokhrul Islam Buian, Ramisha Anan Arde, Md Masum Billah, Amit Debnath and Iqtiar Md Siddique, “Advanced analytics for predicting traffic collision severity assessment”, World Journal of Advanced Research and Reviews, 2024, 21(02), 2007–2018.

[41] Alam, Fatema Binte, Prajoy Podder, and M. Rubaiyat Hossain Mondal. "RVCNet: A hybrid deep neural network framework for the diagnosis of lung diseases." Plos one 18.12 (2023): e0293125.