International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Distributed Algorithms for Large-Scale Computing in Cloud Environments: A Review of Parallel and Distributed Processing

Hilmi Salih Abdullah[1,2*], Subhi R. M. Zeebaree[3]

Submitted: 13/12/2023 **Revised**: 27/01/2024 **Accepted**: 02/02/2024

Abstract: The popularity of cloud computing and large-scale distributed systems is rapidly increasing because of the variety of service models and advantages they offer as well as the necessity of individuals and organizations to access their resources easily and efficiently, in addition to the need for more reliable and robust systems. For these reasons, many distributed algorithms have been designed to facilitate the coordination and interconnection among the distributed computational elements to work together in parallel to achieve a common goal. These algorithms are related to various aspects such as consensus, load balancing, scheduling, communication, leader selection and fault tolerance. Many researches have been carried out to investigate and improve the performance of these distributed algorithms. Therefore, this paper studies and compares a variety of research works that has been performed in distributed algorithms for large-scale cloud computing.

Keywords: *Cloud Computing, Distributed Algorithms, Large-Scale computing, Parallel Processing, Distributed Processing.*

1. Introduction

The digital landscape has transformed extensively in the recent decades especially with emergence of cloud computing. These changes have altered the way how individuals and organizations use, access and manage their computing resources. The necessity for large scale computing, parallel processing and the need for distributed systems spread over different locations as well as the widespread adoption of the internet in addition to the flourish of virtualization technology were important factors for the advent of cloud computing[1][2][3]. At first the cloud computing was offering Software as a Service (SaaS), then a new service model has been added which was Infrastructure as a Service (IaaS), which offers virtualized computing resources such storage, networking and servers. Later, Platform as a Service (PaaS) was offered which provided developers a platform to build and deploy their applications without the need to manage the underlying infrastructure[4][5][6]. Due to the variety of service model offered by cloud computing, the necessity for cloud computing increases. Moreover, the advantages that distributed systems and cloud computing have over traditional ones are an important factor for the importance of the cloud computing. Some of these advantages or characteristics of the cloud computing are scalability, reliability, cost-effectiveness and security[7][8]. Generally, for the large-scale distributed computing or cloud computing to work, there is a strong need for distributed algorithms in order for multiple computing nodes to interact, coordinate and communicate with each other[9][10][11]. Moreover, these distributed algorithms have various purposes or objectives. Some of them are specialized for fault tolerance, others for scheduling the tasks to be processed[12][13]. Additionally, there are algorithms for managing load balancing, others are used for selecting the head node in the distributed systems. Moreover, another type of algorithms is utilized for consensus and decision making as well as others are specialized in communication and synchronization[14][15][16]. This paper studies a variety of researches that have been carried out by different researchers on all the mentioned distributed algorithms for large-scale cloud computing. The findings of the researches have been summarized in a comparison table. The paper's organization is as follows: Section II provides and overview of cloud computing along with its service models and types, large-scale computing, parallel processing, distributed computing and distributed algorithms. Section III provides literature review of previous studies in distributed algorithms. Section IV presents the comparison and discussions of the reviewed researches.Then, section V presents the extracted statistics, later, recommendations are provided in section VI. Finally, section V provides the conclusion.

IT Dept., Technical College of Informatics-Akre, Akre University of pplied Sciences, Duhok, Iraq, hilmi.salih@auas.edu.krd
[2]*ITM Dept., Duhok Technical College, Duhok Polytechnic University, Duhok, Iraq, hilmi.salih@dpu.edu.krd*
[3]*Energy Eng. Dept., Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq, subhi.rafeeq@dpu.edu.krd*

2. Background Theory

2.1. Cloud Computing

Cloud computing means delivering IT resources over the internet. In other words, computing resources can be accessed online, instead of owning the physical infrastructure such as servers and networking equipment as well as software and intelligence[17][1]. The cloud computing can have many benefits such as saving costs, as organizations can save money on hardware, needed maintenance, IT staff and software. The second benefit is scalability as the customers can scale up and down the resources per their needs within a faster time. Third benefit is flexibility, since the resources are online, they can be accessed from anywhere as long as there is available internet connection[15][18]. Reliability is considered another benefit as the cloud providers uptime is high and offers reliability. The last important benefit from the cloud computing is security[19]. Managing the security of the resources is tedious task as it has a lot of complexity but with a variety of security features offered by cloud providers and with dedicated skilled teams, protection of the data and applications is better[20] [21][22][23].

2.2. Cloud Computing Services

Generally, Cloud computing has three popular cloud service models:

1) *Infrastructure as a Service (IaaS):* Within this service model, virtualized computing resources are provided over the internet. Instead of buying and owning the hardware, the users can access and manage thier resouces online. Example of these resources are virutal machines, storage and networking. The users can scale up and down these resouces per their needs. However, the IaaS doesn't come with disadvantages as latency issue can be considered a challenge as well as limited customization and control[24], [25][26].

2) *Platform as a Service (PaaS):* This type of service offers the software platform to the users. Typically, developers use PaaS to develop and manage their apps without the need to care about the infrastructure and the platform. Usually, the organizations benefit from this service to shorten the application development life cycle and increase the productivity as well as the availability of a variety of developmenet tools facilitate the develepment process. On the other side one of the disadvantages might be having less control over the infrastructure as well as compatibility issues[27][4][28][29].

3) *Software as a Service(SaaS):* This service model enables the users and enterprises to subscribe to software applications online without the need to install or maintain them. These software applications are hosted and managed by third-party provides. One of the charactersitics of SaaS is accessibility, which means that the applications are accessible via a web browser over the internet without the need to install and maintain the application on local devices. Another advantage of this model is that the SaaS provided manages the updates ensuring that the user always have the most updated and secure version of the application. However, this model comes with drawbacks too like speed of delivery as the performance might be reduced with slow internet connections[25] [30][31].

2.3. Cloud Computing Types

Generally, there are three main types of cloud computing:

1) *Public Cloud:* It is a type of cloud computer which offers the services over the internet. The services might be Virtual machines or storage, PaaS or SaaS. Usually this type of cloud is owned and operated by enterprises and offers the services commercially to the customers. An example of this type could be Amazon Web Services (AWS) or Microsoft Azure[32].

2) *Private Cloud:* With this type, the cloud computing enviroment is private or dedicated to one organization, meaning that the resources or controls are used by a sepecific organization only. The private cloud can be on-permises or hosted by a cloud provider. Generally this type of cloud has the charactersitics of customization and flexibility as it can customized and configured in accordance with the specific requirements[5].

3) *Hybrid Cloud:* This type can be describes as the combination between private and public cloud. It takes the advantages of both types to meet the specific business requirements. On of the important charecteristics of this type of cloud is it enhances security and control. This is because the organization has more control and maintain a higher security over sensitive data and applications[4][33].

2.4. Large-Scale Computing

Large-scale computing is using extensive computational resources in processing and analyzing large amounts of data and performing complex computations. This type of computing is needed when the traditional computing systems are unable to achieve these large and complex computations[2][34]. Usually large-scale computing involves high processing speed, massive data sets or big data, distributed and parallel processing, simulation and modeling, machine learning and artificial intelligence as well as cloud computing[3][35].

2.5. Parallel Processing

Parallel processing is a computing technique which breaks down the a specific task into a number of smaller tasks then processes these tasks simultaneously. The main purpose of parallel processing is to enhance the computational speed and efficiency by dividing the load among multiple processing units. Parallel processing is considered as a key concept in modern computing and it has many applications in areas that have intensive computations requirements such as weather forecasting and financial modeling[9][36][37]. However, parallel processing have some challenges like data dependency which requires the subtasks to be independent so that they can be executed simultaneously. Another challenge is load balancing, which requires the tasks to be distributed evenly among the processing elements to avoid overloading and idling of some processing elements[38][39][40].

2.6. Disributed Computing

Distributed computing means when multiple interconnected computing elements work together to perform a specific task. Unlike centralized systems the workload is divided among several nodes that are connected by a network. Distributed computing has several advantages such as resilience and redundancy which means that there is no single point of failure, when one node fails the others can continue the work[21][41][42]. Another advantage is scalability, when the load increases, more nodes can be added. Popular examples of distributed processing are cloud computing, blockchain and grid computing[20][43][44][45].

2.7. Disributed Algorithms

When multiple commuting elements are interconnected with each other, there is a need for algorithms which helps the nodes to work together to perform a common goal. Distributed algorithms can be divided to many types depending on the aspect they are dealing with, most commonly are the following algorithms[7]:

1) *Consensus Algorithms:* The purpose of these algorithms is to enable the interconnected nodes to take a unique decision or value which ensures the reliability of the distributed system. Most common examples of consensus algorithms are Raft and Paxos[46][47].

2) *Communication Algorithms:* The purpose of communication algorithm is to enable the nodes to exchange data among each *other* efficiently and reliably. Most common communication algorithms are Message Passing Interface (MPI), Vector Clock and Lamport Clock[48].

3) *Load Balancing Algorithms:* The purpose of these algorithms is to divide the workload evenly among all the computational *nodes*, to ensure the optimal performcance and resource unitlization in the distributed system. Some popular load balancig algorithms in distributed computing are Round Robin, HoneyBee, Throttled , Genetic algorithms[49][30].

4) *Fault Tolerance Algorithms:* Fault tolerance alogorithms enable the distributed systems to continue its operations *even* when faults or failures happen. The most popular fault tolerance algorithms in distributed computing are replication and Byzantine fault tolerance[50][51].

5) *Leader Selection Algorithms:* This type of alogorithms is needed in distributed computing to ensure that a particular *node* acts as the leader with special responsibilites to coordinate activities in a specific distributed system. Most popular leader selection algorithms are Ring and Bully algorithms[52].

6) *Scheduling Algorithms:* This type algorithms are mainly used for managing resource allocation along with the tasks *execution* across the interconnected nodes. The scheduling algorithms have an important role in ensuring high performance and optimal resource utilization in the distributed system. Some examples of such algorithms are MapReduce and Bulk Synchronous Parallel (BSP)[53][54].

3. Literature Review

In recent years, many researchers have studied the distributed algorithms of large-scale cloud computing environment and offered solutions and improvements to the distributed algorithms. This section will review the previous studies that focused on a variety of distributed algorithms used for distributed and parallel processing such as scheduling, load balancing, consensus, fault tolerance, communication and leader selection algorithms. [55] used vector clocks in their distributed system in order to detect the occurrence of concurrent updates and correct the order of the updates in case they are not concurrent. Their system guaranteed data consistency between various regions and ensured the synchronization of the replicas.[48] used a distributed vector clock for event ordering and synchronization in their distributed system. The method simulates a synchronous system on an asynchronous network based on the specified requirements such as accuracy and size of the network. [24] used the message passing interface (MPI) algorithm to test the performance of the SaaS applications on a private cloud. [46] proposed Proof Of Work (POW) algorithm which is widely used in blockchain systems such as Ethereum blockchain, to be used for solving consensus issues in the cloud computing. [56] proposed an improved algorithm for consensus known as KV-Raft which bypasses some of Raft's limitations on reading and writing procedures. The new algorithm KV-Raft is more productive and easier to implement with great potential to be adopted widely. [57]proposed and improved Paxos algorithm entitled PigPaxos to overcome the communication bottlenecks at strong consistency replication protocols. PigPaxos managed to successfully reduce the bottleneck of the leader's communication and improved the decision-making process which resulted in strong consistency in the distributed system. [47]Proposed a modified version of Raft algorithm that is mainly optimized for virtualized cloud environments named vRaft. The proposed algorithm accelerates the read and write requests processing resulting in better latency and throughput. [58] Have proposed an algorithm which uses a replication-based fault tolerance strategy on VMs. The result has shown better execution time and less delay time. [59] designed a scalable Byzantine fault tolerance method to improve fault tolerance on distributed system. [60] proposed an improved algorithm which is inspired by the genetic algorithm entitled Head Node Selection Algorithm (HNSA). With this optimized algorithm, the head node selection along with the candidate node is based on the available resources. [61] Proposed a ring algorithm, in which every node shares its ID with the other nodes and keeps the list of IDs of other nodes, then according to the priority, the algorithm selects the leader. [62] Proposed a bully algorithm in which the leader node is selected dynamically based on the ID of the node. [63] Proposed an algorithm to improve the round-robin algorithm by considering the priority of each task before assigning them to the virtual machines. In other words, the process of tasks assignment is done dynamically based on their priority. [49] Developed an enhanced component-based throttled load balancing algorithm which manages the workload over the cloud in a better way.[53] Presented a dynamic resource management for cloud distributed system using Bulk Synchronous Parallel (BSP) algorithm. The proposed work has shown improvement in performance and overcome limitations. [64] Proposed a simple MapReduce-based Apriori algorithm that has only one phase and doesn't use a combined with other features. The empirical results have shown that it outperforms traditional one.

4. Discussion and Comparison

This section provides a sufficient discussion for the previous works explained in the literature review section. Table 1 illustrates concentrated comparison among the algorithms of all these previous works. We studied various distributed algorithms for large-scale ccomputing in Cloud Environments related to parallel and distributed processing. The purpose and the algorithms along with the findings are summarized as shown in Table 1. The researchers used various algorithms for different distributed processing purposes in their researches, such as load balancing and fault tolerance, communication ,consensus, leader selection and scheduling. [46] [56] [57] [47] have used improved versions of consensus distributed algorithms such as vRaft, KV-Raft, PigPaxos and Proof of Work (POW). These improved algorithms have successfully solved some consensus problems. On the other hand, [60] [61] [62] have used leader selection algorithms , Ring, Bully and Head Node Selection Algorithm (HNSA) which improved the head node selection problem. Some of the researchers such as [24] [48] [55] investigated communication algorithms in the distributed systems such as vector clock and MPI which resulted in handling synchronization issues. Moreover, [49] and [63] used load balancing algorithms Round Robin and Throttled respectively which resulted in better task assignment and improved the management of the workload. Furthermore, [58] and [59] have worked on fault tolerance algorithms, replication and Byzantine fault tolerance respectively which improved the fault tolerance in distributed systems. Finally, [53] and [64] have used scheduling distributed algorithms Bulk Synchronous Parallel (BSP) and MapReduce respectively which have improved the performance and bypassed some scheduling limitations.

Table 1: Comparison Of Reviewed Researches

Reference	Algorithm	Large-Scale Purpose	Findings and Result
[53], 2023	Bulk Synchronous Parallel (BSP)	Scheduling	1.The performance of the distributed system has improved 2.Overcome limitations.
[49], 2022	Throttled	Load Balancing	1. Improved Managing the workload of the cloud computing distributed system more efficiently.
[6], [24], 2022	Message Passing Algorithm (MPI)	Communication	1. Better performance of SaaS applications in a private cloud.
[58], 2022	Replication	Fault Tolerance	1. Better execution time 2. less delay time in distributed systems.
[63], 2022	Round Robin	Load balancing	1. The process of tasks assignment in the distributed VMs is done dynamically. 2. Tasks assignment is based on their priority.
[48], 2021	Vector Clocks	Communication	1. Handling Synchronization issues in distributed cloud systems with vector clocks algorithm.
[56], 2021	KV-Raft	Consensus	1. Improved consensus in distributed systems. 2. Overcome limitation issues with Raft.
[57], 2021	PigPaxos	Consensus	1. Managed to successfully reduce the bottleneck of the leader’s communication. 2. Improved the decision-making process in distributed systems.

[47], 2021	vRaft	Consensus	1. Accelerating the read and write requests processing, 2. Resulting in better latency and throughput in cloud computing.
[59], 2021	Byzantine fault tolerance	Fault Tolerance	1. Designed a scalable Byzantine fault tolerance method to improve fault tolerance on distributed system.
[55], 2021	Vector Clocks	Communication	1. The distributed system proved consistency 2. Improved synchronization among the replicas in various regions.
[60], 2021	Head Node Selection Algorithm (HNSA)	Leader Selection	1. Improved a distributed algorithm which the head node selection 2. The head node selection along with the candidate node is based on the available resources.
[46], 2020	Proof Of Work (POW)	Consensus	1. POW was successful in solving consensus issues in the cloud computing distributed system.
[62], 2020	Bully	Leader Selection	1. The leader node is selected dynamically in the distributed system.
[64], 2019	MapReduce	Scheduling	1. The proposed algorithm was better and outperformed traditional ones.
[61], 2019	Ring	Leader Selection	1. The leader node of the distributed system is selected according to the priority.

5. Extracted Statistics

It can be noticed from the review studies that researchers have shed lights on various distributed algorithms types in order to check them, optimize or even suggest a better replacement for them. Figure 1 below summarized percentage of the various types of distributed algorithms that the researchers focused on which were cover by this review paper.

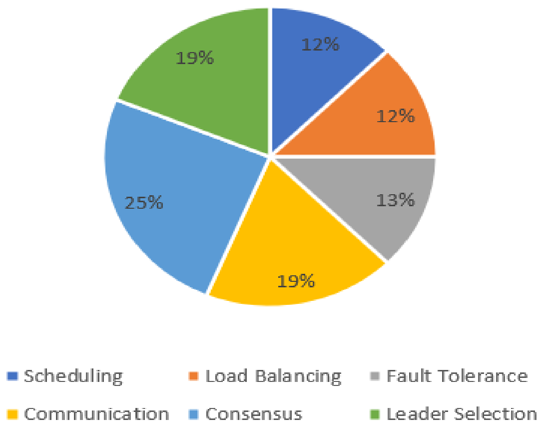

Fig 1. Percentage of distributed algorithms types that have be studied by researchers.

6. Recommendations

According to the reviewed papers and the finding of the experiments performed by the researchers in distributed algorithms. It is recommended to use KV-Raft algorithm to solve consensus issues because it surpasses other algorithms and overcomes limitations of the Raft algorithm. On, the other hand, is it recommended to used Bulk Synchronous Parallel (BSP) for managing scheduling in distributed systems due to its performance and overcoming limitations. In addition, for managing load balancing issues Throttled algorithm is recommended due its good workload management. Moreover, for fault tolerance in distributed systems, it is recommended to use Byzantine fault tolerance algorithm. Furthermore, for communication issues it is recommended to use Ring algorithm. Finally, it recommended to use Head Node Selection Algorithm for handling leader selection issues in Distributed systems. In addition to the above recommendations, more researches need to be done in this important area, which needs continuous improvement in the distributed algorithms in pace with the continuous advancement, complexity and importance of large-scale distributed systems and cloud computing.

7. Conclusion

With the increasing adoption of cloud computing and large-scale distributed and parallel processing applications, there is an insistent need to explore and study this field in more details. One of the important aspects of cloud computing and distributed processing is how various computing nodes communicate and work together via a variety of distributed algorithms. Shedding the light on these algorithms and understanding them well, will have a positive impact on improving them to perform better. This study has reviewed multiple most recent researches on the most popular distributed algorithms and summarized their purpose and findings in one comparison table. Finally, the study discussed the improvements made by the researchers and their positive impact of the distributed computing algorithms.

References

[1] B. Varghese and R. Buyya, “Next generation cloud computing: New trends and research directions,” *Future Generation Computer Systems*, vol. 79, pp. 849–861, 2018.

[2] J. Chinna and K. Kavitha, “A study on Large Scale Graph Processing Frameworks Performance for Cloud Operations in different nodes.”

[3] F. Li, “Execution Feature Extraction and Prediction for Large-Scale Graph Processing Applications,” in *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, 2019, pp. 84–89.

[4] M. Birje, P. S. Challagidad, M. Tapale, and R. Goudar, “Security issues and countermeasures in cloud computing,” *International Journal of Applied Engineering Research*, vol. 10, no. 86, pp. 71–5, 2015.

[5] P. Kumari and P. Kaur, “A survey of fault tolerance in cloud computing,” *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 10, pp. 1159–1176, 2021.

[6] P. Y. Abdullah, S. Zeebaree, H. M. Shukur, and K. Jacksi, “HRM system using cloud computing for Small and Medium Enterprises (SMEs),”

Technology Reports of Kansai University, vol. 62, no. 04, p. 04, 2020.

[7] J. Zhang, H. Huang, and X. Wang, “Resource provision algorithms in cloud computing: A survey,” *Journal of network and computer applications*, vol. 64, pp. 23–42, 2016.

[8] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, “A load balancing algorithm for the data centres to optimize cloud computing applications,” *IEEE Access*, vol. 9, pp. 41731–41744, 2021.

[9] S. Heidari and R. Buyya, “A cost-efficient auto-scaling algorithm for large-scale graph processing in cloud environments with heterogeneous resources,” *IEEE Transactions on Software Engineering*, vol. 47, no. 8, pp. 1729–1741, 2019.

[10] S. R. Zeebaree, H. M. Shukur, L. M. Haji, R. R. Zebari, K. Jacksi, and S. M. Abas, “Characteristics and analysis of hadoop distributed systems,” *Technology Reports of Kansai University*, vol. 62, no. 4, pp. 1555–1564, 2020.

[11] P. Y. Abdullah, S. Zeebaree, K. Jacksi, and R. R. Zeabri, “An hrm system for small and medium enterprises (sme) s based on cloud computing technology,” *International Journal of Research-GRANTHAALAYAH*, vol. 8, no. 8, pp. 56–64, 2020.

[12] A. Rehman, R. L. Aguiar, and J. P. Barraca, “Fault-tolerance in the scope of cloud computing,” *IEEE Access*, vol. 10, pp. 63422–63441, 2022.

[13] M. M. A. Baig, “An Evaluation of Major Fault Tolerance Techniques Used on High Performance Computing (HPC) Applications,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 3s, pp. 320–328, 2023.

[14] S. Chand and Y. A. Liu, “Brief Announcement: What’s Live? Understanding Distributed Consensus,” in *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, 2021, pp. 565–568.

[15] S. S. Priya and T. Rajendran, “Improved round-robin rule learning for optimal load balancing in distributed cloud systems,” *International Journal of System of Systems Engineering*, vol. 13, no. 1, pp. 83–99, 2023.

[16] J. Nikoli’c, N. Jubatyrov, and E. Pournaras, “Self-healing dilemmas in distributed systems: Fault correction vs. fault tolerance,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 2728–2741, 2021.

[17] D. Tomar and P. Tomar, “Integration of cloud computing and big data technology for smart generation,” in *2018 8th International Conference on Cloud Computing, Data Science \& Engineering (Confluence)*, 2018, pp. 1–6.

[18] Z. N. Rashid, S. R. Zeebaree, M. A. Sadeeq, R. R. Zebari, H. M. Shukur, and A. Alkhayyat, “Cloud-based Parallel Computing System Via Single-Client Multi-Hash Single-Server Multi-Thread,” in *2021 International Conference on Advance of Sustainable Engineering and its Application (ICASEA)*, 2021, pp. 59–64.

[19] S. Zeebaree and others, “DES encryption and decryption algorithm implementation based on FPGA,” *Indones. J. Electr. Eng. Comput. Sci*, vol. 18, no. 2, pp. 774–781, 2020.

[20] Y. Afek, G. Giladi, and B. Patt-Shamir, “Distributed Computing with the Cloud,” in *Stabilization, Safety, and Security of Distributed Systems: 23rd International Symposium, SSS 2021, Virtual Event, November 17-20, 2021, Proceedings 23*, 2021, pp. 1–20.

[21] E. Jonas, Q. Pu, S. Venkataraman, I. Stoica, and B. Recht, “Occupy the cloud: Distributed computing for the 99\%,” in *Proceedings of the 2017 symposium on cloud computing*, 2017, pp. 445–451.

[22] S. R. Zeebaree, A. B. Sallow, B. K. Hussan, and S. M. Ali, “Design and simulation of high-speed parallel/sequential simplified DES code breaking based on FPGA,” in *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 2019, pp. 76–81.

[23] S. Zeebaree, R. R. Zebari, K. Jacksi, and D. A. Hasan, “Security approaches for integrated enterprise systems performance: A Review,” *Int. J. Sci. Technol. Res*, vol. 8, no. 12, pp. 2485–2489, 2019.

[24] O. Bystrov, A. Ka\vceniauskas, and R. Pacevi\vc, “Cost and Performance Analysis of MPI-Based SaaS on the Private Cloud Infrastructure,” in *International Conference on Parallel Processing and Applied Mathematics*, 2022, pp. 171–182.

[25] M. Masdari, F. Salehi, M. Jalali, and M. Bidaki, “A survey of PSO-based scheduling algorithms in cloud computing,” *Journal of Network and Systems Management*, vol. 25, no. 1, pp. 122–158, 2017.

[26] Z. S. Ageed and S. R. Zeebaree, “Distributed Systems Meet Cloud Computing: A Review of Convergence and Integration,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 11s, pp. 469–490, 2024.

[27] S. Basu, A. Bardhan, K. Gupta, P. Saha, M. Pal, M. Bose, K. Basu, S. Chaudhury, and P. Sarkar, “Cloud computing security challenges \& solutions-A survey,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 347–356.

[28] H. Malallah, S. R. Zeebaree, R. R. Zebari, M. A. Sadeeq, Z. S. Ageed, I. M. Ibrahim, H. M. Yasin, and K. J. Merceedi, “A comprehensive study of

kernel (issues and concepts) in different operating systems," *Asian Journal of Research in Computer Science*, vol. 8, no. 3, pp. 16–31, 2021.

[29] H. Shukur, S. Zeebaree, R. Zebari, O. Ahmed, L. Haji, and D. Abdulqader, "Cache coherence protocols in distributed systems," *Journal of Applied Science and Technology Trends*, vol. 1, no. 3, pp. 92–97, 2020.

[30] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 3910–3933, 2022.

[31] D. A. Hasan, B. K. Hussan, S. R. Zeebaree, D. M. Ahmed, O. S. Kareem, and M. A. Sadeeq, "The impact of test case generation methods on the software performance: A review," *International Journal of Science and Business*, vol. 5, no. 6, pp. 33–44, 2021.

[32] M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 12, p. e4123, 2017.

[33] H. M. Zangana and S. R. Zeebaree, "Distributed Systems for Artificial Intelligence in Cloud Computing: A Review of AI-Powered Applications and Services," *International Journal of Informatics, Information System and Computer Engineering (INJIISCOM)*, vol. 5, no. 1, pp. 1–20, 2024.

[34] S. R. Zeebaree, L. M. Haji, I. Rashid, R. R. Zebari, O. M. Ahmed, K. Jacksi, and H. M. Shukur, "Multicomputer multicore system influence on maximum multi-processes execution time," *TEST Engineering \& Management*, vol. 83, no. 03, pp. 14921–14931, 2020.

[35] D. M. ABDULQADER, S. R. ZEEBAREE, R. R. ZEBARI, S. A. SALEH, Z. N. RASHID, and M. A. SADEEQ, "SINGLE-THREADING BASED DISTRIBUTED-MULTIPROCESSOR-MACHINES AFFECTING BY DISTRIBUTED-PARALLEL-COMPUTING TECHNOLOGY," *Journal of Duhok University*, vol. 26, no. 2, pp. 416–426, 2023.

[36] L. M. Haji, S. R. Zeebaree, O. M. Ahmed, M. A. Sadeeq, H. M. Shukur, and A. Alkhavvat, "Performance Monitoring for Processes and Threads Execution-Controlling," in *2021 International Conference on Communication \& Information Technology (ICICT)*, 2021, pp. 161–166.

[37] L. M. Haji, S. R. Zeebaree, Z. S. Ageed, O. M. Ahmed, M. A. Sadeeq, and H. M. Shukur, "Performance Monitoring and Controlling of Multicore Shared-Memory Parallel Processing Systems," in *2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA)*, 2022, pp. 44–48.

[38] Y. Chen, B. Brock, S. Porumbescu, A. Buluc, K. Yelick, and J. Owens, "Atos: A task-parallel GPU scheduler for

[39] K. Zhang, Y. Fang, Y. Zheng, H. Zeng, L. Xu, and W. Wang, "Graphlib: A parallel graph mining library for joint cloud computing," in *2020 IEEE International Conference on Joint Cloud Computing*, 2020, pp. 9–12.

[40] S. Zeebaree and K. Jacksi, "Effects of processes forcing on CPU and total execution-time using multiprocessor shared memory system," *Int. J. Comput. Eng. Res. Trends*, vol. 2, no. 4, pp. 275–279, 2015.

[41] Z. N. Rashid, S. R. Zeebaree, R. R. Zebari, S. H. Ahmed, H. M. Shukur, and A. Alkhayyat, "Distributed and Parallel Computing System Using Single-Client Multi-Hash Multi-Server Multi-Thread," in *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, 2021, pp. 222–227.

[42] M. A. Sadeeq and S. R. Zeebaree, "Design and implementation of an energy management system based on distributed IoT," *Computers and Electrical Engineering*, vol. 109, p. 108775, 2023.

[43] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2017.

[44] M. A. Sadeeq and S. R. Zeebaree, "DPU-ALDOSKI dataset of Monitoring and Controlling distributed far distances energy consumed system based on Internet of Things," *Data in Brief*, vol. 49, p. 109455, 2023.

[45] Y. S. Jghef, S. R. Zeebaree, Z. S. Ageed, and H. M. Shukur, "Performance Measurement of Distributed Systems via Single-Host Parallel Requesting using (Single, Multi and Pool) Threads," in *2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA)*, 2022, pp. 38–43.

[46] X. Zhu, J. Shi, S. Huang, and B. Zhang, "Consensus-oriented cloud manufacturing based on blockchain technology: An exploratory study," *Pervasive and mobile computing*, vol. 62, p. 101113, 2020.

[47] Y. Wang and Y. Chai, "vRaft: accelerating the distributed consensus under virtualized environments," in *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11-14, 2021, Proceedings, Part I 26*, 2021, pp. 53–70.

[48] M. H. Cintuglu and D. Ishchenko, "Real-time asynchronous information processing in distributed power systems control," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 773–782, 2021.

[49] D. Mekonnen, A. Megersa, R. K. Sharma, D. P. Sharma, and others, “Designing a Component-Based Throttled Load Balancing Algorithm for Cloud Data Centers,” *Mathematical Problems in Engineering*, vol. 2022, 2022.

[50] W. Zhong, C. Yang, W. Liang, J. Cai, L. Chen, J. Liao, and N. Xiong, “Byzantine Fault-Tolerant Consensus Algorithms: A Survey,” *Electronics*, vol. 12, no. 18, p. 3801, 2023.

[51] R. Guo, Z. Guo, Z. Lin, and W. Jiang, “A hierarchical byzantine fault tolerance consensus protocol for the internet of things,” *High-Confidence Computing*, p. 100196, 2023.

[52] V. K. Madisetti and S. Panda, “A dynamic leader election algorithm for decentralized networks,” *Journal of Transportation Technologies*, vol. 11, no. 3, pp. 404–411, 2021.

[53] E. Wang, Y. Barve, A. Gokhale, and H. Sun, “Dynamic Resource Management for Cloud-native Bulk Synchronous Parallel Applications,” in *2023 IEEE 26th International Symposium on Real-Time Distributed Computing (ISORC)*, 2023, pp. 152–157.

[54] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, and J.-J. Li, “Distributed evolutionary algorithms and their models: A survey of the state-of-the-art,” *Applied Soft Computing*, vol. 34, pp. 286–300, 2015.

[55] D. House, H. Kuang, K. Surendran, and P. Chen, “Toward fast and reliable active-active geo-replication for a distributed data caching service in the mobile cloud,” *Procedia Computer Science*, vol. 191, pp. 119–126, 2021.

[56] Y. Wang, Z. Wang, Y. Chai, and X. Wang, “Rethink the linearizability constraints of raft for distributed key-value stores,” in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021, pp. 1877–1882.

[57] A. Charapko, A. Ailijiang, and M. Demirbas, “Pigpaxos: Devouring the communication bottlenecks in distributed consensus,” in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 235–247.

[58] K. K. Agarwal and H. Kotakula, “Replication Based Fault Tolerance Approach for Cloud,” in *International Conference on Distributed Computing and Internet Technology*, 2022, pp. 163–169.

[59] Y. Li, L. Qiao, and Z. Lv, “An optimized byzantine fault tolerance algorithm for consortium blockchain,” *Peer-to-Peer Networking and Applications*, vol. 14, pp. 2826–2839, 2021.

[60] S. Kanwal, Z. Iqbal, A. Irtaza, M. Sajid, S. Manzoor, and N. Ali, “Head node selection algorithm in cloud computing data center,” *Mathematical problems in engineering*, vol. 2021, pp. 1–12, 2021.

[61] I. Abraham, D. Dolev, and J. Y. Halpern, “Distributed protocols for leader election: A game-theoretic perspective,” *ACM Transactions on Economics and Computation (TEAC)*, vol. 7, no. 1, pp. 1–26, 2019.

[62] S. S. Supase and R. B. Ingle, “Are coordinator election algorithms in distributed systems vulnerable?,” in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–5.

[63] A. Katangur, S. Akkaladevi, and S. Vivekanandhan, “Priority Weighted Round Robin Algorithm for Load Balancing in the Cloud,” in *2022 IEEE 7th International Conference on Smart Cloud (SmartCloud)*, 2022, pp. 230–235.

[64] P. Kijsanayothin, G. Chalumporn, and R. Hewett, “On using MapReduce to scale algorithms for Big Data analytics: a case study,” *Journal of Big Data*, vol. 6, pp. 1–20, 2019.