Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Optimizing Performance in Distributed Cloud Architectures: A Review of Optimization Techniques and Tools

Khalid Ibrahim Khalaf Jajan [1*], Subhi R. M. Zeebaree[2]
khalid.ibrahim@auas.edu.krd, subhi.rafeeq@dpu.edu.krd
[1]Information Technology Department, Technical College of Informatics-Akre, Akre University for Applied Sciences, Duhok, Iraq
[2]Energy Eng. Department, Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq

Article Information

Submitted : 5 Mar 2024
Reviewed: 12 Mar 2024
Accepted : 1 Apr 2024

Keywords

Distributed cloud, optimization

Abstract

This research paper presents a groundbreaking hybrid transactional / analytical processing (HTAP) architecture designed to revolutionize real-time point cloud data processing, particularly in autonomous driving environments. Integrating elements from both columnar and row-based tables within a spatial database, the proposed architecture offers unparalleled efficiency in managing and updating point cloud data in real-time. The architecture's distributed nature operates through a seamless synergy of Edge and Cloud components. The Edge segment operates within the Robot Operating System (ROS) environment of the vehicle, while the Cloud counterpart functions within the PostgreSQL environment of cloud services. The communication between these components is facilitated by Kafka, ensuring rapid and reliable data transmission. A pivotal aspect of the proposed system lies in its ability to autonomously detect changes in point cloud data over time. This is achieved through a sophisticated algorithm that analyzes dissimilarities in the data, triggering real-time updates in areas where high dissimilarity is detected. The system ensures the maintenance of the latest state of point cloud data, contributing significantly to the generation of safe and optimized routes for autonomous vehicles. In terms of optimization, the paper demonstrates how the HTAP architecture achieves real-time online analytical processing through query parallelization in a distributed database cluster. The system's efficacy is evaluated through simulations conducted in the CloudSim framework, showcasing its scalability, adaptability, and robustness in handling point cloud data processing for a single vehicle. While acknowledging the achievement of the proposed architecture, certain limitations are recognized. The study highlights the need for further investigation into the system's performance under simultaneous analysis and updates from multiple vehicles. Additionally, ensuring seamless scalability and robustness for uninterrupted operation and expansion during runtime is identified as an area requiring further development.

A. Introduction

In the intricate tapestry of modern technology, the convergence of cloud computing and advanced data management has become a linchpin for driving innovation, efficiency, and scalability across various sectors. This review encapsulates the collective insights from twenty research articles, each offering a unique lens into the evolving landscape of cloud computing and data management. The amalgamation of cloud-based infrastructures with cutting-edge algorithms, architectures, and technologies represents a pivotal frontier in addressing the challenges posed by an increasingly data-centric and interconnected world [1].

The overarching theme of optimization is woven throughout the diverse array of articles, collectively forming a mosaic that seeks to enhance the efficiency of computing resources and data handling. From the intricacies of deploying Service Function Chains (SFCs) across multiple domains to the orchestration of drone swarms in Quantum Metropolitan Area Networks (QMANs), researchers are actively engaged in unraveling the complexities of resource allocation, task scheduling, and data processing. This review serves as a comprehensive exploration of how optimization, in its myriad forms, serves as the backbone for realizing the full potential of cloud-based systems [2].

A recurrent motif is the introduction and refinement of optimization algorithms. Ranging from metaheuristic approaches like Evolution Strategies and Whale Optimization Algorithm to intricate Deep Neural Networks (DNNs) in Cloud Radio Access Networks (C-RANs), the articles underscore a commitment to pushing the boundaries of computational efficiency [3]. The utilization of belief propagation-based power allocation schemes and the integration of blockchain technology in hierarchical access control for IoT networks showcase the versatility and adaptability required to navigate the intricacies of diverse computing environments.

Architectural innovation emerges as a prominent thread, weaving through the exploration of Software-Defined Networking (SDN)-based architectures in Quantum Drone Networks (QDNs) to the introduction of Hybrid Cloud-Based Data Processing (HCBDP) models. Researchers are not only focusing on optimizing existing cloud architectures but also on creating flexible, scalable infrastructures capable of meeting the dynamic demands of emerging applications. Edge computing, fog networks, and vehicular clouds further underline the shift towards decentralized and distributed architectures, acknowledging the challenges posed by mobility, latency, and resource constraints [4].

Furthermore, the review highlights the exploration of advanced technologies such as quantum networks, blockchain, and digital twins. The integration of these cutting-edge technologies signifies a forward-looking approach aimed at enhancing security, reliability, and predictive capabilities[5]. Quantum networks, showcased in Quantum Drone Networks (QDNs), and blockchain-based solutions in hierarchical access control for IoT networks showcase the pursuit of innovation and security in tandem[7].

While the collective contributions push the boundaries of cloud computing, certain recurring considerations and limitations emerge. The challenges of mobility, energy efficiency, real-time processing, and scalability echo throughout the articles, reflecting the complex nature of today's computing landscape.

In essence, this review serves as a panoramic view of the diverse efforts propelling the realms of cloud computing and data management into the future. By encapsulating the essence of each article's contribution, it provides readers with a holistic perspective on the state-of-the-art advancements, challenges, and potential trajectories within this dynamic intersection of technology. As we navigate through this multifaceted terrain, it is evident that the relentless pursuit of optimization and innovation will continue to shape the future of cloud computing and data management, offering solutions to the challenges of today and opening doors to the possibilities of tomorrow.

B. Background Theory

2.1. Distributed Systems

Distributed systems have emerged as a pivotal force reshaping the landscape of modern computing, challenging the conventional notions of centralized architectures and opening new frontiers in scalability, efficiency, and fault tolerance. At their essence, distributed systems are an ensemble of interconnected nodes, each contributing to the collective processing and coordination of tasks[7]. This departure from monolithic structures acknowledges the limitations of traditional computing models in meeting the demands of today's dynamic and data-intensive applications.

In the realm of distributed systems, a node signifies an individual computational entity, whether it be a computer or any device capable of processing and communication. The intricate web of communication channels that link these nodes forms the backbone of collaborative efforts, enabling the exchange of information and facilitating the seamless orchestration of distributed tasks. The spectrum of distributed computing models is diverse, encompassing the classical client-server architecture, where a central server manages and responds to client requests, to the more decentralized peer-to-peer systems, where each node has equal status and contributes both computing resources and data[8].

The challenges posed by distributed systems are multifaceted, with managing concurrency and ensuring consistency across nodes standing out as paramount concerns. Concurrency, the simultaneous execution of multiple tasks, demands intricate synchronization mechanisms to prevent conflicts and maintain data integrity. Achieving consistency across distributed nodes, essential for coherent system behavior, requires sophisticated algorithms and protocols to navigate the complexities of data updates and communication delays[9].

However, the journey through the intricacies of distributed systems is not without its rewards. The advantages offered by distributed systems are compelling, driving their widespread adoption in diverse domains. Enhanced performance, achieved through parallel processing and load balancing, stands as a testament to the potential of distributed systems to tackle computationally intensive tasks with unparalleled efficiency[10]. Resilience to failures is another cornerstone of distributed systems, as their decentralized nature ensures that the failure of individual nodes does not cripple the entire system[11].

As we navigate this dynamic landscape, it becomes evident that a nuanced understanding of the theoretical foundations of distributed systems is crucial. From classical distributed algorithms to contemporary consensus protocols, these theoretical underpinnings guide the design and implementation of robust and scalable solutions. The evolving nature of computing demands not only technological innovation but also a deep comprehension of the collaborative and decentralized principles that underpin the distributed systems paradigm[12].

In conclusion, distributed systems represent a transformative force in the realm of computing, offering a paradigm that aligns with the demands of the modern era. As we continue to push the boundaries of what is possible in computational capabilities, the distributed systems framework stands as a beacon, illuminating a path towards scalable, efficient, and resilient computing architectures.

2.2. Optimization Performance

In the fast-paced realm of technology, Optimization Performance stands as a linchpin in achieving the full potential of computing systems. It embodies the strategic pursuit of refining algorithms, architectures, and processes to enhance efficiency, minimize resource utilization, and amplify overall system functionality. At its core, optimization performance seeks to extract the maximum output from a given set of resources, ensuring that computational tasks are executed with precision and speed[13].

Efficiency, a central tenet of optimization performance, is intricately linked to resource utilization. This entails the judicious allocation and management of computing resources such as CPU cycles, memory, and storage. By fine-tuning these elements, optimization performance aims to eliminate bottlenecks and streamline workflows, resulting in faster and more responsive systems[14]. The pursuit of efficiency extends beyond hardware considerations to encompass software algorithms and code structures, where optimization techniques aim to minimize computational overhead and streamline the execution of instructions.

In the context of software development, optimization performance is often synonymous with crafting code that not only accomplishes its intended purpose but does so with minimal computational cost. This involves leveraging algorithmic optimizations, data structures, and programming paradigms that strike a delicate balance between functionality and resource efficiency[15]. In a world where applications span diverse domains, from web services to scientific simulations, the impact of optimization on user experience, responsiveness, and energy consumption is profound.

The quest for optimization performance is not a one-size-fits-all endeavor but a nuanced exploration that takes into account the specific characteristics and requirements of each computing task. Techniques such as parallelization, caching, and just-in-time compilation are embraced to tailor solutions to the unique demands of diverse applications. In the realm of parallel computing, optimization performance plays a pivotal role in harnessing the power of multi-core architectures, ensuring that computational tasks are distributed efficiently to achieve maximum throughput[16].

In conclusion, Optimization Performance is the driving force behind the quest for efficiency in computing. It transcends hardware and software boundaries, seeking to unlock the latent capabilities of systems and applications. As technology continues its relentless evolution, the significance of optimization performance becomes even more pronounced, shaping the way we design, implement, and experience the ever-expanding landscape of computational possibilities[17]. Through continuous refinement and innovation, optimization performance charts a course toward a future where computing systems operate at the peak of their potential, delivering speed, responsiveness, and efficiency in equal measure.

2.3. Cloud Computing

Cloud computing has emerged as a transformative force in the digital landscape, revolutionizing the way individuals and businesses access and manage computing resources. At its core, cloud computing is a model that enables ubiquitous, on-demand access to a shared pool of configurable computing resources, including networks, servers, storage, applications, and services. This paradigm shift moves away from traditional, on-premises infrastructure, allowing users to leverage the power of remote servers and services delivered over the internet[18].

One of the defining characteristics of cloud computing is its scalability, offering the flexibility to scale resources up or down based on demand. This elasticity empowers organizations to efficiently manage fluctuating workloads, ensuring optimal resource utilization and cost-effectiveness. Cloud services are typically categorized into Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), each offering different levels of abstraction and control over the underlying infrastructure [19].

Moreover, cloud computing fosters collaboration and accessibility by providing a centralized platform for data storage and application deployment. Users can access their data and applications from anywhere with an internet connection, breaking down geographical barriers and promoting a more connected global ecosystem[20]. This democratization of resources levels the playing field, allowing startups and enterprises alike to harness powerful computing capabilities without the need for substantial upfront investments in physical infrastructure.

Security concerns and data privacy are integral considerations in cloud computing, leading to the development of robust security measures, encryption protocols, and compliance standards. As organizations increasingly migrate their operations to the cloud, the industry continues to evolve, introducing innovations like serverless computing and edge computing to address specific needs and further enhance the efficiency and responsiveness of cloud-based systems.

In conclusion, cloud computing represents a pivotal shift in how we conceptualize and utilize computing resources. Its scalability, accessibility, and collaborative potential make it a cornerstone in the digital transformation of businesses and societies, fostering innovation and enabling a more dynamic and interconnected future.

2.4. Optimization Techniques and Tools

Optimization techniques and tools play a pivotal role in fine-tuning and enhancing the performance of computational systems, ensuring they operate at peak efficiency. These strategies encompass a diverse array of approaches, ranging from algorithmic optimizations to hardware-level enhancements, all geared towards maximizing the utilization of resources and minimizing bottlenecks[21].

Algorithmic optimizations form a cornerstone of performance enhancement, involving the refinement of computational processes to achieve faster execution and reduced computational complexity. By streamlining algorithms, developers can significantly impact the efficiency of software applications, leading to improved response times and resource utilization[22].

On the hardware front, tools for profiling, debugging, and performance monitoring are essential in identifying and addressing bottlenecks. Profiling tools, for instance, help analyze the execution of a program, revealing areas that may benefit from optimization. Debugging tools assist in identifying and rectifying errors, contributing to overall system stability and reliability[23].

Parallelization, a technique that involves breaking down tasks into smaller sub-tasks processed concurrently, is another powerful optimization strategy. This is particularly relevant in the era of multi-core processors, where parallel computing can unlock significant performance gains[24].

Furthermore, just-in-time (JIT) compilation and caching mechanisms are optimization tools that dynamically adapt code execution to improve runtime efficiency. JIT compilation translates high-level code into machine code at runtime, allowing for targeted optimizations based on the actual execution environment[25].

In conclusion, optimization techniques and tools are indispensable in the pursuit of efficient computing. Whether through algorithmic refinements, hardware profiling, or parallelization, these approaches collectively contribute to elevating the performance of computational systems, ensuring they meet the demands of today's dynamic and resource-intensive applications.

C. Introduction Related Work

In navigating the complex landscape of cloud computing, optimization strategies, and data management, it is imperative to delve into the body of related work that has laid the foundation for the current state of research. The interplay between cloud computing and advanced data management has been a subject of continuous exploration, with a focus on refining existing paradigms and introducing novel approaches to address the evolving demands of modern computing environments. A recurring theme in the related work is the deployment of optimization algorithms to enhance the efficiency and effectiveness of cloud-based systems. Metaheuristic algorithms, such as the Evolution Strategies and Whale Optimization Algorithm, have been investigated for their efficacy in task scheduling and resource allocation across diverse cloud computing environments. These studies form the backbone of efforts to balance the computational load, minimize response times, and optimize resource utilization in the cloud.

Kibalya et al. [26] Focused on deploying Service Function Chains (SFCs) across multiple domains, where determining the optimal Infrastructure Providers (InPs) to embed these chains poses a complex problem due to various reasons. To address this, the researcher introduced a multi-stage graph-based algorithm that considered multiple request constraints (like bandwidth or delay) in a distributed manner, solving the NP-hard problem of embedding SFCs. The algorithm used a candidate search technique to construct the multi-stage graph, reducing computational complexity and achieving improved acceptance ratio and embedding cost compared to a benchmark state-of-the-art algorithm. The researchers used the Multi-stage graph-based algorithm aided by a candidate search technique as an Optimizer Method. However, the limitations of the researcher were they assumed immutable requirements for requests regarding link and node resources throughout their lifetime. However, in real-world scenarios, requirements may vary over time, necessitating an adaptive embedding algorithm. Additionally, handling limited information disclosure elastically is challenging, requiring algorithms to intelligently infer resource availability or changes in request requirements based on partially disclosed information and past experience.

Deng et al. [27] addressed the limitations of IoT development caused by latency in wireless networks and computation failures in resource-constrained environments by proposing a mobile edge computing (MEC) model for IoT service provision. In this model, edge servers with cached services via wireless networks aim to reduce latency and offload computations. The researcher presented an optimized service cache policy leveraging service composability to enhance system performance, reducing the average response time for IoT services. The optimizer method they used was consumption-driven searching algorithm for cache policy determination. Regarding the limitations, the proposed model assumed service deployment policy based on historical service frequency or popularity, leading to a potential cold start problem when new services are introduced. Additionally, assuming equal execution times on edge and cloud servers might oversimplify the model; incorporating more detailed parameters could further optimize the system. The absence of consideration for latent service characteristics is identified as a future avenue for improvement.

Askri et al. [28] Explored uplink reception in Cloud Radio Access Networks (C-RANs), proposing a novel method using Deep Neural Networks (DNNs) for efficient uplink C-RAN operations, subject to quantization rules. The introduced architecture, QDNet, optimized processing both at Remote Radio Units (RRUs) and at the Central Processor (CP), addressing quantization constraints. QDNet, a distributed DNN with sparse connections, is inspired by the projected gradient descent algorithm and outperforms current solutions like zero-forcing (ZF) equalizers and sphere decoders (SD) in certain scenarios, achieving up to 2 dB SNR gain over ZF with comparable or lower computational complexity. It also demonstrated near-optimal performance compared to the SD algorithm, especially under specific fronthaul link capacity and multiple RRU scenarios. The QDNet, a distributed DNN with sparse connections, inspired by projected gradient descent as an optimizer method and the tool they used was deep neural networks (DNNs). The limitation of the researcher was they focused on mimicking the entire transmission process in uplink C-RAN through the proposed DNN architecture but does not address perfect knowledge assumptions about observations, which could affect real-world implementation. Additionally, the uniform quantization encoding with a fixed number of bits across detection algorithms might limit capacity; a more adaptive approach could enhance performance. Further comparisons with NN-assisted MIMO receivers in the literature, even under corrupted observations, could provide more comprehensive insights.

CHITI et al. [29] Explored the utilization of drone swarms in Quantum Networks (QNs) for establishing Quantum Metropolitan Area Networks (QMANs), overcoming limitations of Optical Fibers (OFs). They highlighted drones' potential in creating a non-terrestrial QMAN and addresses challenges posed by random drone fluctuations and atmospheric turbulence on Free Space Optic (FSO) links. The proposed solution involved a Software-Defined Networking (SDN)-based architecture managing end-to-end (E2E) entangled pairs distribution, crucial for Quantum Drone Networks (QDNs) in quantum computing and Quantum Key Distribution (QKD) services. Additionally, an objective function is introduced to optimize swarm mission planning and operation, considering meteorological conditions. The used optimizer method was objective function for optimizing drone number based on meteorological conditions and the tool they used was Software-Defined Networking (SDN) architecture integrated with drone swarms. However the limitation can be described as the researchers focused on addressing atmospheric challenges and optimizing mission planning for drone swarms in QMANs. However, it didn’t extensively discuss the complexities of multi-Controller segments or interoperability among QMANs using different aerial platforms or satellite segments, which could be crucial for practical implementation. Future developments should consider these aspects for a more comprehensive QDN framework.

WANG et al. [30] addressed the impact of User Equipment’s (UEs) mobility in Fog Computing Networks (FCNs), where the limited coverage of FCNs may lead to radio handovers and execution results migration, causing extra costs in terms of energy consumption and latency for UEs. To maximize UE revenue, the researchers proposes a three-layer fog computing network architecture considering UE mobility characterized by sojourn time distribution across FCN coverage areas. The problem is formulated as a Mixed Integer Nonlinear Programming (MINLP) problem, split into tasks off-loading and resource allocation components. The proposed Gini coefficient-based FCNs selection algorithm (GCFSA) and a distributed resource optimization algorithm based on genetic algorithms (ROAGA) aim to minimize migration probabilities. They used gini coefficient-based FCNs selection algorithm and genetic algorithm-based resource optimization as an optimizer method. The limitation can be noticed as the reserachers modeled UEs' mobility using the exponential distribution of sojourn times but does not explicitly address reducing migration costs for tasks. Future work should focus on strategies to minimize the cost of migration for migrated tasks, enhancing the practicality and efficiency of the proposed algorithms in real-world fog computing scenarios.

Suthakar et al. [31] delved into the challenges of monitoring vast computational jobs and user activities within scientific infrastructures, traditionally managed via Relational Database Management Systems (RDBMS). Recent assessments explored the Lambda Architecture (LA) using Hadoop and MapReduce for large-scale monitoring datasets, proving its superiority to RDBMS solutions but highlighting complexity in implementation and maintenance. To address this, the researchers introduced an Optimized Lambda Architecture (OLA) using the Apache Spark ecosystem. OLA seamlessly integrates batch and real-time computation without added complexity. The optimizer method used was optimized Lambda Architecture (OLA) leveraging Apache Spark for combined batch and real-time processing and the tool used was Apache Spark ecosystem. However, the limitations of the researchers were they demonstrated the OLA's benefits over RDBMS and the traditional Lambda Architecture in terms of execution time, low latency, maintenance, and scalability, it didn’t delve into potential drawbacks or specific challenges faced during the adoption or implementation of OLA. A more nuanced analysis of its limitations or potential bottlenecks could provide a comprehensive understanding of its practicality in different infrastructures.

Behbehani et al. [32] explored leveraging underutilized vehicular resources to form a vehicular cloud, augmenting conventional cloud and fixed edge computing in a distributed processing architecture. They introduced a Mixed Integer Linear Programming (MILP) model to optimize processing demand allocation, minimizing overall power consumption. Results showed a significant power savings, up to 84% compared to conventional cloud processing, even with variations in processing and traffic demand. They used MILP model optimizing processing demand allocation while minimizing power consumption as an optimizer method. On the other side, the limitation were that they highlighted the energy efficiency of processing in vehicles compared to conventional cloud decreases with increasing demand size. Additionally, it emphasized the challenge of limited data rates in vehicular wireless interfaces, hindering efficient distributed processing in vehicles and edge nodes due to traffic replication. Higher data rate interfaces are identified as crucial for better vehicular cloud utilization. While a heuristic is developed to allocate processing demands in real-time, the researchers could have provided more insight into the practical challenges of implementing this vehicular cloud architecture in dynamic vehicular environments.

Hardjawana et al. [33] Introduced belief propagation-based power allocation schemes for Cloud-based Small Cell Networks (C-SCNs), addressing the challenge of high computational complexity and latency at cloud computational units (CCUs) due to the increasing number of base stations (BSs). The proposed schemes optimize various network objectives such as minimizing energy consumption and maximizing spectral efficiency. By distributing computation across multiple processors and leveraging parallel computation, these schemes achieve low latency and computational complexity even with a growing number of BSs. The used optimizer method was belief propagation-based power allocation schemes for C-SCNs, optimized for various network objectives, computed in parallel. The limitation of the researchers emphasized the advantages of the proposed schemes—low complexity, latency, and high spectral/energy efficiencies—it didn't discuss potential limitations or challenges in implementing these schemes in real-world scenarios. More insights into practical deployment challenges or scalability concerns would have provided a comprehensive understanding of the feasibility of these schemes in dynamic network environments.

Mingxin et al. [34] Proposed a Blockchain-based Distributed Key Management Architecture (BDKMA) coupled with fog computing for hierarchical access control in IoT networks. They addressed limitations of centralized key management by leveraging blockchain technology to achieve decentralization, auditability, scalability, and extensibility while prioritizing privacy. The architecture employed a multiblockchain structure to support cross-domain access, with side blockchains managed by Security Access Managers (SAMs) in each domain and multiblockchains stored in the cloud for inter-domain interaction. Simulation results indicated improved system performance and scalability with the multiblockchain structure, adaptable to varying network sizes and environments through dynamic transaction collection time adjustment. The blockchain-based distributed key management architecture with fog computing for hierarchical access control in IoT networks is used as optimizer method and the tool used was blockchain technology, fog computing. The limitations could be summarized that the researchers presented the advantages and performance improvements of the proposed BDKMA, it lacked discussion on potential drawbacks or challenges in implementing blockchain-based solutions in IoT environments. Future work could focus on exploring and implementing feedback mechanisms for SAMs and cloud managers to ensure the persistence and robustness of the blockchain-based IoT ecosystem.

PETRA and PAULA. [35] Proposed an innovative approach for optimizing task scheduling in heterogeneous Cloud computing environments using the (μ, λ)-Evolution Strategies metaheuristic algorithm. The researchers aimed to enhance task scheduling across various virtual machines and data centers, particularly for resource-intensive scientific jobs. The Evolution Strategies algorithm, novel in this domain, is applied to schedule tasks in CloudSim simulations based on ALICE job production. Comparisons with the Genetic Algorithm highlight the proposed approach's superiority, demonstrating improvements in metrics like makespan, resource utilization, throughput, execution time, imbalance, and scalability. The optimizer method was (μ, λ)-Evolution Strategies metaheuristic algorithm for task scheduling in heterogeneous Cloud environments and the tool used was CloudSim framework for simulations. The limitations of the researchers were that they emphasized the advantages of the proposed Evolution Strategies-based scheduling over existing algorithms, it could further delve into potential challenges or constraints in implementing this approach in real-world, dynamic Cloud environments. Moreover, the researchers discussed future research directions aiming to optimize the algorithm's fitness operator and design a proactive scheduling approach but lacked specific details on how these improvements will be integrated or tested against related state-of-the-art algorithms.

Chen et al. [36] Focused on improving task scheduling in cloud computing to enhance system efficiency and reduce operational costs. The Whale Optimization Algorithm (WOA) is introduced as a metaheuristic algorithm for cloud task scheduling, with a multiobjective optimization model. The researchers proposed an advanced approach called Improved WOA for Cloud task scheduling (IWC) to enhance the optimal solution search capability of the WOA-based method. Experimental results demonstrated that IWC exhibits better convergence speed and accuracy compared to existing metaheuristic algorithms, leading to improved performance in system resource utilization for both small and large-scale tasks. The future work involves refining the balance between exploration and exploitation in the IWC approach, exploring parallel implementations in cloud environments to reduce scheduling overhead, and extending the model to handle more complex task jobs, including workflows and cloud-based deep learning workloads. optimizer method used was Whale Optimization Algorithm (WOA) is used for cloud task scheduling. An advanced approach called Improved WOA for Cloud task scheduling (IWC) is proposed to further enhance the optimal solution search capability. One potential limitation could be the generalization of the proposed approach to handle diverse and complex task workloads. The researchers mentioned future work to address issues such as QoS problems and handling more complex task jobs, but it is essential to assess the scalability and adaptability of the proposed system in practical cloud computing environments.

Awad et al. [37] Addressed the challenge of optimizing data replication in cloud computing to enhance usability, performance, and stability of application systems. Two bio-inspired algorithms, Multi-Objective Particle Swarm Optimization (MO-PSO) and Multi-Objective Ant Colony Optimization (MO-ACO), are proposed for dynamic data replication and placement. MO-PSO focused on selecting the best data replica based on frequency, while MO-ACO optimizes replica placement considering distance and availability. CloudSim is used for simulation, where data centers with hosts and virtual machines are randomly configured, and replication files are distributed randomly. Performance evaluation against various approaches demonstrated that MO-PSO yields improved data replication, while MO-ACO achieves higher data availability, lower cost, and reduced bandwidth consumption compared to other algorithms. Two optimizer methods were used which were Multi-Objective Particle Swarm Optimization (MO-PSO) for selecting the best data replica based on frequency and Multi-Objective Ant Colony Optimization (MO-ACO) for optimizing data replica placement based on distance and availability. The tool used was CloudSim is employed for the simulation of the proposed strategy in a cloud computing environment. Limitation of the researchers were that they highlighted the effectiveness of the proposed algorithms in simulation, it does not explicitly mention potential limitations. It would have been valuable to assess how the algorithms perform in more diverse and complex real-world cloud computing scenarios. Additionally, the researchers mentioned future work involving testing on a real cloud computing environment, but the specific challenges and considerations for transitioning from simulation to real-world implementation were not discussed in detail.

Dos et al. [38] Addressed challenges in Big Data analytics, specifically related to user unfamiliarity with Cloud infrastructure, performance improvement requirements, and resource management for stable processing. To address these issues, the researchers proposed a hybrid model combining Cloud Computing and Volunteer Computing environments for Big Data analytics, referred to as Hybrid Cloud-Based Data Processing (HCBDP). The contributions included an evaluation for efficient deployment, the development of an HR_Alloc Algorithm for data placement, and a resource allocation model in hybrid infrastructures. Results indicated the feasibility of using a hybrid infrastructure with up to 35% unstable machines without performance loss and a lower cost than Classical Cloud Computing. The proposed model demonstrated viability for decreasing computing expenditures and improving data load balancing in hybrid infrastructures. Optimizer Method was the HR_Alloc Algorithm is introduced for data placement in the HCBDP model, ensuring operational continuity in environments with unstable machines without a loss of performance. Additionally, the article discusses the use of parameters (ϕHC and ϕVC) to find suitable resources and establish relationships between Cloud Computing Hybrid Environment (CCHE) and Volunteer Data Centers (VDC). The limitation of the researchers was that they did not discuss potential limitations explicitly. It would have been beneficial to address scalability concerns and potential challenges in real-world deployment.

Additionally, the proposed model's performance in diverse and complex scenarios is not thoroughly explored, and further evaluation in real-world environments is recommended. Future works included considerations for Fog computing, evaluation of storage mechanisms, and exploration of accelerators, suggesting an evolving research agenda.

Devaraj et al. [39] Proposed a new load balancing algorithm for Cloud Computing (CC) called FIMPSO, which was a hybrid of the Firefly (FF) algorithm and Improved Multi-Objective Particle Swarm Optimization (IMPSO) technique. FIMPSO aimed to effectively distribute workloads and computing resources in the cloud environment. The FF algorithm minimized the search space, while IMPSO is employed to identify enhanced responses. The algorithm selected the global best (gbest) particle using a small distance from a point to a line, improving resource usage and response time of tasks. Simulation results demonstrated that FIMPSO outperforms other methods, achieving an effective average response time of 13.58 ms, maximum CPU utilization of 98%, memory utilization of 93%, reliability of 67%, throughput of 72%, and a make span of 148. The proposed algorithm was deemed energy-efficient and exhibits superior performance in comparison to other load balancing methods. The optimizer method was FIMPSO algorithm is introduced as a hybrid of Firefly (FF) algorithm and Improved Multi-Objective Particle Swarm Optimization (IMPSO) technique for load balancing in the cloud environment. The limitation of the researchers were that they didn’t explicitly discuss potential limitations of the proposed FIMPSO algorithm. Future improvements are suggested, including the use of data deduplication algorithms. It would have been beneficial to address scalability concerns and evaluate the algorithm's performance in more diverse and complex cloud computing scenarios. Additionally, the comparison with other methods in the simulation outcomes should have considered different workload scenarios to ensure the algorithm's robustness.

Alzuhair and Alghaihab [40] Focused on the application of technology in agriculture, particularly in the context of acoustic sensing. It presented the design and performance analysis of a machine learning (ML) model integrated into an Artificial Intelligence of Things (AIoT) platform tailored for agriculture. The researchers emphasized the optimization of a sensor node by combining the ML model and a wireless network, resulting in an agricultural-specific AIoT platform. The co-design aimed to enhance performance and efficiency for acoustic and ambient sensing applications. The researchers covered the design, analysis, and optimization of critical components in the sensor node used for AI deployment in agricultural applications, considering both architectural and hardware aspects. The research evaluates the energy and spectrum efficiency of utilizing edge intelligence compared to traditional RF modules for acoustic and ambient sensing. The results suggested that edge intelligence is more efficient, particularly for lightweight applications in agriculture, though potential limitations are acknowledged, and the need for different architectures for compute-intensive applications is highlighted. The optimizer method use was an integrated sensor node for agricultural applications by combining a machine learning model and a wireless network, creating an AIoT platform. The study evaluates the distribution of computing load between edge devices and the cloud, focusing on the design and performance analysis of critical components in the sensor node. The tools used was machine learning models for sound classification in agricultural applications and utilizes specific hardware implementations for optimized overall system performance. The paper discusses the use of edge intelligence for lightweight applications, emphasizing its energy and spectrum efficiency. The limitation of the researchers was that they acknowledged that processing data at the edge could introduce limitations, particularly in terms of computing resources compared to centralized infrastructure. The impact of these limitations depended on the application.

Wang et al. [41] Addressed the reliability and safety of dynamically tuned passive power filters (DTPPFs) used for harmonic mitigation in industrial applications. The focus is on identifying core device faults in DTPPFs to ensure safe and efficient operation. The researchers proposed a cloud server-assisted remote monitoring and fault identification system for DTPPFs, encompassing the monitoring system's architecture, cloud servers' software architecture, and software design for both the back-end service layer and the front-end application layer. The system aimed to provide early warning of core device faults in real-time, optimizing equipment maintenance schedules and leveraging manufacturers' service capacities. The experiments conducted demonstrated the system's capability to monitor the operational status of DTPPFs remotely and identify core device faults effectively. The optimizer method was proposing a cloud server-assisted remote monitoring and fault identification system for DTPPFs, utilizing online monitoring technology. The system is designed to identify faults in core devices of DTPPFs and optimize equipment maintenance schedules through real-time monitoring. The tools used was online monitoring technology and database design to store relevant operational parameters. It utilized cloud servers for remote monitoring and fault identification in DTPPFs. The limitation was in acknowledging that, while they monitored a few dozens of DTPPFs in their experiments, monitoring hundreds of DTPPFs or more may introduce challenges related to high-concurrency data impact on the cloud server. The authors planed to investigate this in future work.

Benblidia et al. [42] Addressed the growing importance of data centers in the context of 5G networks, IoT, and cloud computing, emphasizing the energy-intensive nature of cloud data centers. They explored the potential benefits of managing them in microgrids for reliability and sustainability. The challenge lies in optimizing the power allocation to data centers within microgrids to minimize energy costs. The proposed two-stage optimization approach in a microgrid-cloud architecture was detailed. In the first stage, the researchers modeled the power allocation to cloud data centers as a non-cooperative game, considering multiple providers. The microgrid controller calculates optimal power assignments based on factors such as Power Usage Effectiveness (PUE), the number of real-time applications, and network bandwidth usage. In the second stage, microgrids aimed to minimize their energy costs by strategically purchasing power from the main grid and other microgrids while selling back stored energy. The researchers compared the proposed approach with three existing power minimization approaches: "Basic Game Scheme" (BGS), traditional, and price-based methods. Simulation results indicated that the two-stage optimization approach is up to 25%

more effective in terms of energy cost. Additionally, the proposed scheme promoted the usage of green energy in microgrids, leading to a significant reduction in power load rate and CO2 emissions compared to other schemes. The optimizer method was employing a two-stage optimization approach, with the first stage modeling the power allocation to cloud data centers as a non-cooperative game within the microgrid. The second stage focused on minimizing energy costs by strategically purchasing and selling power. Tool used was Matlab optimization tool to solve the constrained linear optimization problem in the second stage of the proposed approach. The limitation was it did not discuss potential limitations explicitly. Future work is suggested, considering dynamic pricing using microgrids and cooperative cloud data centers to minimize energy costs while meeting client Quality of Service (QoS) requirements. Further research could explore the practical implementation and scalability of the proposed approach in real-world microgrid-cloud environments.

Vries et al. [43] Introduced a cost-profiling solution for optimizing the operating expenses of Kubernetes-based microservice applications. They addressed the challenges of understanding the cost implications in the context of distributed and heterogeneous microservice architectures. The proposed solution leverages an open-source application performance monitoring (APM) stack, specifically built for Kubernetes-based microservices. A case study with a data engineering company demonstrates how the tool offered deeper insights into the cost profile of various application components, enabling informed decision-making in managing deployments. The optimizer method was implementing a cost-profiling tool on top of the Elastic Stack for monitoring and optimizing Kubernetes-based microservice applications. The tool enables proportional attribution of operating expenses to different levels of the system architecture, allowing for a more granular understanding of cost distribution. The tools used was the cost-profiling tool is built on the Elastic Stack, an open-source observability solution. The Elastic Stack facilitates the monitoring and analysis of the microservice application's operational expenses. One limitation of the proposed solution is the steep learning curve and significant costs associated with managing the volume of produced data. Organizations implementing this tool would need to weigh these challenges against the potential value it adds in reducing cloud expenses.

Kim and Moon [44] Addressed the challenge of efficiently managing and processing point cloud data in real-time for autonomous driving environments. The proposed solution introduced a distributed hybrid transactional/analytical processing (HTAP) architecture that leverages both columnar and row-based tables in a spatial database. This architecture ensured real-time online analytical process query performance through query parallelization in a distributed database cluster. An algorithm analyzed dissimilarity in point cloud data, updating relevant areas in real-time when high dissimilarity is detected. The proposed system aimed to support autonomous vehicles in generating safe and optimized routes. The architecture was divided into an Edge part operating in the ROS environment of the vehicle and a Cloud part operating in the PostgreSQL environment of cloud services, with data transmission facilitated through Kafka. The system autonomously detected changes over time and maintains the latest state of point cloud data, contributing to the generation of safe, optimized routes. The optimizer method was a distributed HTAP architecture that efficiently manages and updates point cloud data in real-time. The architecture utilized both columnar and row-based tables, ensuring real-time processing through query parallelization in a distributed database cluster. The tools used involved distributed database technology, Kafka for data transmission, and AWS Lambda for data preprocessing. Spatial databases, specifically PostgreSQL, play a crucial role in real-time insertion, updating, and analysis of point cloud data. The imitation were that the architecture was designed to handle point cloud data processing and updates for a single vehicle. Its performance under simultaneous analysis and updates from multiple vehicles needed further investigation as well as ensuring seamless scalability and robustness of the architecture for uninterrupted operation and expansion during runtime remains an area that required further development.

D. Discussion and Comparison

After reviewing all the papers mentioned in the related studies section, some most relevant information and metrics have been extracted from the papers. Including the main focus of the researchers in their studies as wells the optimizer methods they used. In addition, the tools used for optimizing performance in distributed cloud architectures were extracted. Later we focused on the limitations of each reviewed paper and finally we concluded the contribution of each paper in optimizing the performance in distributed cloud architecture. The most useful metrics from the reviewed papers are compared and presented in the Table1.

Table 1. Comparison among the reviewed works.

Ref.	Main Focus	Optimizer Method	Tools Used	Limitation	Contribution
[26], 2021	SFC Deployment	Multi-stage graph-based algorithm with candidate search	Not explicitly mentioned	Assumes immutable requirements; Limited information disclosure handling is challenging	Proposes a comprehensive algorithm for efficient SFC deployment
[27], 2018	IoT Service Provision	Consumption-driven searching algorithm for cache policy	Not explicitly mentioned	Assumes service deployment policy based on historical service frequency; Equal execution times assumption might oversimplify the model	Introduces a mobile edge computing model for IoT service provision with an optimized service cache policy
[28], 2021	Uplink C-RAN Operations	QDNet, distributed DNN with sparse connections	Deep Neural Networks (DNNs)	Does not address perfect knowledge assumptions; Uniform quantization encoding might limit capacity	Proposes QDNet architecture for efficient uplink C-RAN operations with improved SNR gain
[29], 2022	Quantum Drone Networks	Objective function for optimizing drone number based on meteorological conditions	Software-Defined Networking (SDN) architecture with drone swarms	Limited discussion on multi-Controller segments and interoperability among QMANs	Introduces a SDN-based architecture for Quantum Drone Networks (QDNs) with an objective function for optimizing drone swarms

[30], 2019	Fog Computing Networks	Gini coefficient-based FCNs selection algorithm and genetic algorithm-based resource optimization	Not explicitly mentioned	Models UEs' mobility using exponential distribution; Future work needed to focus on minimizing migration costs	Proposes GCFSA and ROAGA algorithms for fog computing networks to minimize migration probabilities
[31], 2021	Optimized Lambda Architecture	Optimized Lambda Architecture leveraging Apache Spark	Apache Spark ecosystem	Does not delve into potential drawbacks or specific challenges faced during the adoption or implementation	Introduces OLA using Apache Spark for efficient combined batch and real-time processing
[32], 2022	Vehicular Cloud	MILP model optimizing processing demand allocation while minimizing power consumption	Not explicitly mentioned	Energy efficiency decreases with increasing demand size; Limited data rates in vehicular wireless interfaces	Proposes a MILP model for optimizing processing demand allocation in vehicular clouds
[33], 2016	Cloud-based Small Cell Networks	Belief propagation-based power allocation schemes	Not explicitly mentioned	Does not discuss potential limitations or challenges in real-world scenarios	Addresses the high computational complexity and latency challenges in Cloud-based Small Cell Networks (C-SCNs) using belief propagation-based schemes
[34], 2019	Blockchain-based Distributed Key Management	Blockchain-based distributed key management architecture with fog computing	Blockchain technology, fog computing	Lacks discussion on potential drawbacks; Future work could focus on feedback mechanisms	Introduces a Blockchain-based Distributed Key Management Architecture (BDKMA) for hierarchical access control in IoT networks
[35], 2022	Cloud Task Scheduling	(μ, λ)-Evolution Strategies metaheuristic algorithm	CloudSim framework	Could delve into potential challenges or constraints in real-world dynamic Cloud environments	Proposes an innovative Evolution Strategies-based approach for task scheduling in heterogeneous Cloud computing environments
[36], 2020	Whale Optimization Algorithm	Whale Optimization Algorithm (WOA) for cloud task scheduling	Not explicitly mentioned	Does not explicitly discuss potential limitations; Future work could address scalability concerns	Introduces WOA for cloud task scheduling with an advanced approach called Improved WOA
[37], 2021	Multi-Objective Optimization for Data Replication	Multi-Objective Particle Swarm Optimization (MO-PSO) and Multi-Objective Ant Colony Optimization (MO-ACO)	CloudSim framework	Does not explicitly mention potential limitations; Real-world scenarios' exploration is recommended	Proposes MO-PSO and MO-ACO algorithms for dynamic data replication and placement in Cloud computing

[38], 2020	Hybrid Cloud-Based Data Processing	HR_Alloc Algorithm for data placement in HCBDP model	Not explicitly mentioned	Does not explicitly discuss potential limitations; Scalability concerns and challenges in real-world deployment need consideration	Introduces HR_Alloc Algorithm for data placement in a Hybrid Cloud-Based Data Processing (HCBDP) model
[39], 2020	FIMPSO Algorithm	FIMPSO algorithm (Hybrid of FF and IMPSO) for cloud load balancing	Not explicitly mentioned	Does not explicitly discuss potential limitations; Scalability concerns and performance in diverse scenarios could be explored	Presents FIMPSO algorithm for cloud load balancing with improved convergence speed and accuracy
[40], 2023	AIoT Platform for Agriculture	Integrated Sensor Node Optimization	Machine Learning Models	Edge processing limitations for compute-intensive applications	ML model integration for AIoT in agriculture, energy efficiency
[41]	Cloud-assisted Monitoring of DTPPFs	Remote Monitoring System	Online Monitoring Technology	Impact of high-concurrency data on cloud server	Early fault detection in DTPPFs, cloud-assisted monitoring
[42], 2022	Microgrid-Cloud Architecture	Two-stage optimization approach for power allocation to cloud data centers	Matlab optimization tool	Does not discuss potential limitations explicitly; Further exploration for scalability and robustness is suggested	Proposes a two-stage optimization approach for power allocation in microgrid-cloud architecture
[43], 2023	Cost Profiling for Kubernetes-based Microservices	Elastic Stack	Elastic Stack	Steep learning curve and costs associated with data volume management	Deeper insights into cost profiles for microservices in Kubernetes
[44], 2023	Digital Twin for Smart Power Distribution	Digital twin technology for predictive maintenance in smart power distribution	Not explicitly mentioned	Limited application with insufficient sensors; Need for more digital twin models	Explores the application of digital twin technology for predictive maintenance in smart power distribution systems

E. Extracted Statistics

According to the reviewed papers, there are various tools used for Optimizing Performance in Distributed Cloud Architectures. Some of the researchers haven’t stated the tools they used in their papers and others mentioned them. From these researches we can notice that CloudSim framework and machine learning models were the most used tools in the reviewed papers as shown in Figure1.

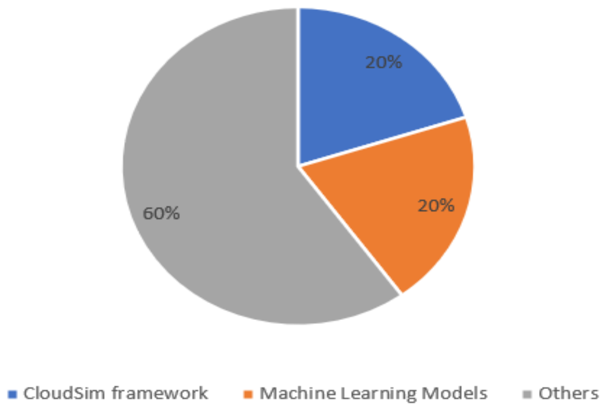

Figure 1. Optimization tools used.

F. Recommendations

Optimizing the performance in distributed cloud architectures is crucial for ensuring a smooth, efficient, and cost-effective experience for your users. It's an ongoing process that involves multiple phases, including design, deployment, monitoring, and improvement. One of the important aspects for future optimization is load balancing. Few of the reviewed paper were focusing on this important aspect. We recommend more research to be done on this particular subject in order to avoid bottlenecks and maximize resource utilization by distributing the traffic evenly across multiple instances. The second important aspect is security. In the reviewed papers, there was more focus on the performance without properly considering the security aspects. Therefore, we recommend the best practices for security in the cloud to be considered without compromising the performance. Finally, performance optimization is an ongoing process, therefore, we recommend continuous monitoring and analysis to maintain the optimal performance distributed cloud architectures.

G. Conclusion

In conclusion, this review paper has provided a comprehensive synthesis of diverse research articles in the fields of cloud computing, IoT, network optimization, and emerging technologies. The contributions of the individual papers span a spectrum of innovative algorithms, resource allocation models, and the integration of cutting-edge technologies to address various challenges in network architectures and emerging paradigms. The advancements in network optimization, as exemplified by the introduced algorithms for optimal Service Function Chaining (SFC) deployment, consumption-driven searching for IoT services, and SDN-based architectures for Quantum Drone Networks (QDNs), underscore the ongoing efforts to enhance the efficiency and adaptability of modern networks. Resource allocation models, employing techniques such as Mixed Integer Linear Programming (MILP) and belief propagation-based power allocation schemes, have shown promise in optimizing processing demand allocation in vehicular cloud architectures and Cloud-based Small Cell Networks (C-SCNs). The integration of emerging technologies, notably blockchain and fog computing, has been explored to fortify hierarchical access control in IoT networks, presenting scalable and secure solutions for network management.

Cloud task scheduling has seen notable improvements through the application of metaheuristic algorithms, with the (μ, λ)-Evolution Strategies, Whale Optimization Algorithm (WOA), and Improved WOA demonstrating enhanced load balancing, task scheduling efficiency, and resource utilization.

The introduction of digital twin tec hnology in smart power distribution systems has emerged as a significant contribution, revolutionizing fault detection, predictive maintenance, and equipment life assessment, all crucial aspects in ensuring the reliability and sustainability of power networks.

In summary, this review paper serves as a cohesive amalgamation of these multifaceted contributions, identifying common threads and research gaps. By synthesizing these diverse perspectives, it adds value to the ongoing discourse on optimizing network architectures, enhancing security, and advancing the deployment of emerging technologies. The collective insights from these papers contribute to the broader understanding of current challenges and future directions in the ever-evolving landscape of network optimization and emerging technologies.

H. References

[1] F. Al-Doghman, N. Moustafa, I. Khalil, N. Sohrabi, Z. Tari, and A. Y. Zomaya, "AI-Enabled Secure Microservices in Edge Computing: Opportunities and Challenges," IEEE Trans Serv Comput, vol. 16, no. 2, pp. 1485–1504, 2023, doi: 10.1109/TSC.2022.3155447.

[2] P. K. Mishra and A. K. Chaturvedi, "State-Of- The-Art and Research Challenges in Task Scheduling and Resource Allocation Methods for Cloud-Fog Environment," in 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), 2023, pp. 1–5. doi: 10.1109/ICCT56969.2023.10076030.

[3] X. Li, G. Zhang, X. Zheng, and S. Hua, "Delay Optimization Based on Improved Differential Evolutionary Algorithm for Task Offloading in Fog Computing Networks," in 2020 International Conference on Wireless Communications and Signal Processing (WCSP), 2020, pp. 109–114. doi: 10.1109/WCSP49889.2020.9299850.

[4] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "A Survey on Cloudlets, Mobile Edge, and Fog Computing," in 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2021, pp. 139–142. doi: 10.1109/CSCloud-EdgeCom52276.2021.00034.

[5] D. A. Hasan, K. Hussan, S. R. M. Zeebaree, D. M. Ahmed, O. S. Kareem, and M. A. M. Sadeeq, "The Impact of Test Case Generation Methods on the Software Performance: A Review," 2021, doi: 10.5281/zenodo.4623940.

[6] Z. S. Ageed, S. R. M. Zeebaree, and R. H. Saeed, "Influence of Quantum Computing on IoT Using Modern Algorithms," in 2022 4th International Conference on Advanced Science and Engineering (ICOASE), 2022, pp. 194–199. doi: 10.1109/ICOASE56293.2022.10075583.

[7] H. Shukur et al., "Characteristics and Analysis of Hadoop Distributed Systems." [Online]. Available: https://www.researchgate.net/publication/341775003

[8] S. R. M. Zeebaree et al., “Multicomputer Multicore System Influence on Maximum Multi-Processes Execution Time”.
[9] R. K. Ibrahim et al., “Clustering Document based on Semantic Similarity Using Graph Base Spectral Algorithm,” in 2022 5th International Conference on Engineering Technology and its Applications (IICETA), 2022, pp. 254–259. doi: 10.1109/IICETA54559.2022.9888613.
[10] 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, Jun. 2020, doi: 10.38094/jastt1329.
[11] P. Y. Abdullah, S. R. M. 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, Aug. 2020, doi: 10.29121/granthaalayah.v8.i8.2020.926.
[12] S. R. M. 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. doi: 10.1109/ICOASE.2019.8723792.
[13] Z. Najat, Z. N. Rashid, H. Sharif, S. Rafeeq, and M. Z. Sulaimani, “Client/Servers Clustering Effects on CPU Execution-Time, CPU Usage and CPU Idle Depending on Activities of Parallel-Processing-Technique Operations ",” Article in International Journal of Scientific & Technology Research, vol. 7, no. 8, 2018, [Online]. Available: www.ijstr.org
[14] Y. S. Jghef, S. R. M. 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. doi: 10.1109/IT-ELA57378.2022.10107923.
[15] L. M. Haji, S. R. M. Zeebaree, Z. S. Ageed, O. M. Ahmed, M. A. M. 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. doi: 10.1109/IT-ELA57378.2022.10107953.
[16] L. M. Haji, S. R. M. Zeebaree, O. M. Ahmed, M. A. M. 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. doi: 10.1109/ICICT52195.2021.9568445.
[17] H. Malallah et al., “A Comprehensive Study of Kernel (Issues and Concepts) in Different Operating Systems,” Asian Journal of Research in Computer Science, pp. 16–31, May 2021, doi: 10.9734/ajrcos/2021/v8i330201.
[18] S. R. M Zeebaree, R. R. Zebari, K. Jacksi, and D. Abas Hasan, “Security Approaches For Integrated Enterprise Systems Performance: A Review,” INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, vol. 8, 2019, [Online]. Available: www.ijstr.org

[19] H. Majeed Zangana and S. R. M. Zeebaree, "Distributed Systems for Artificial Intelligence in Cloud Computing: A Review of AI-Powered Applications and Services," 2024.

[20] Z. Najat, A. Sengur, Z. N. Rashid, and S. R. M. Zeebaree, "Novel Remote Parallel Processing Code-Breaker System via Cloud Computing," 2020. [Online]. Available: https://www.researchgate.net/publication/341767165

[21] T. Mohammed, G. Sami, S. R. M. Zeebaree, and S. H. Ahmed, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING A Novel Multi-Level Hashing Algorithm to Enhance Internet of Things Devices' and Networks' Security." [Online]. Available: www.ijisae.org

[22] Z. S. Ageed and S. R. M. Zeebaree, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Distributed Systems Meet Cloud Computing: A Review of Convergence and Integration." [Online]. Available: www.ijisae.org

[23] P. Abdullah, H. Shukur, K. Jacksi, P. Y. Abdullah, S. R. M. Zeebaree, and H. M. Shukur, "HRM System using Cloud Computing for Small and Medium Enterprises (SMEs)," 2020. [Online]. Available: https://www.researchgate.net/publication/341883552

[24] S. R. M. Zeebaree, "DES encryption and decryption algorithm implementation based on FPGA," Indonesian Journal of Electrical Engineering and Computer Science, vol. 18, no. 2, pp. 774–781, 2020, doi: 10.11591/ijeecs.v18.i2.pp774-781.

[25] T. Mohammed, G. Sami, S. R. M. Zeebaree, and S. H. Ahmed, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Designing a New Hashing Algorithm for Enhancing IoT Devices Security and Energy Management." [Online]. Available: www.ijisae.org

[26] G. Kibalya, J. Serrat-Fernandez, J. L. Gorricho, D. G. Bujjingo, and J. Serugunda, "A Multi-Stage Graph Aided Algorithm for Distributed Service Function Chain Provisioning across Multiple Domains," IEEE Access, vol. 9, pp. 114884–114904, 2021, doi: 10.1109/ACCESS.2021.3104841.

[27] S. Deng, Z. Xiang, J. Yin, J. Taheri, and A. Y. Zomaya, "Composition-driven IoT service provisioning in distributed edges," IEEE Access, vol. 6, pp. 54258–54269, 2018, doi: 10.1109/ACCESS.2018.2871475.

[28] A. Askri, C. Zhang, and G. R. Ben Othman, "Distributed Learning Assisted Fronthaul Compression for Multi-Antenna C-RAN," IEEE Access, vol. 9, pp. 113997–114007, 2021, doi: 10.1109/ACCESS.2021.3104660.

[29] F. Chiti, R. Picchi, and L. Pierucci, "Metropolitan Quantum-Drone Networking and Computing: A Software-Defined Perspective," IEEE Access, vol. 10, pp. 126062–126073, 2022, doi: 10.1109/ACCESS.2022.3226127.

[30] D. Wang, Z. Liu, X. Wang, and Y. Lan, "Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks," IEEE Access, vol. 7, pp. 43356–43368, 2019, doi: 10.1109/ACCESS.2019.2908263.

[31] U. Suthakar, L. Magnoni, D. R. Smith, and A. Khan, "Optimised Lambda Architecture for Monitoring Scientific Infrastructure," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 6, pp. 1395–1408, Jun. 2021, doi: 10.1109/TPDS.2017.2772241.

[32] F. S. Behbehani, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Optimized Processing Placement Over a Vehicular Cloud," IEEE Access, vol. 10, pp. 41411–41428, 2022, doi: 10.1109/ACCESS.2022.3167479.
[33] W. Hardjawana, N. I. Anwar Apandi, and B. Vucetic, "Parallel Optimization Framework for Cloud-Based Small Cell Networks," IEEE Trans Wirel Commun, vol. 15, no. 11, pp. 7286–7298, Nov. 2016, doi: 10.1109/TWC.2016.2600302.
[34] M. Ma, G. Shi, and F. Li, "Privacy-Oriented Blockchain-Based Distributed Key Management Architecture for Hierarchical Access Control in the IoT Scenario," IEEE Access, vol. 7, pp. 34045–34059, 2019, doi: 10.1109/ACCESS.2019.2904042.
[35] P. Loncar and P. Loncar, "Scalable Management of Heterogeneous Cloud Resources Based on Evolution Strategies Algorithm," IEEE Access, vol. 10, pp. 68778–68791, 2022, doi: 10.1109/ACCESS.2022.3185987.
[36] X. Chen et al., "A WOA-Based Optimization Approach for Task Scheduling in Cloud Computing Systems," IEEE Syst J, vol. 14, no. 3, pp. 3117–3128, Sep. 2020, doi: 10.1109/JSYST.2019.2960088.
[37] A. Awad, R. Salem, H. Abdelkader, and M. A. Salam, "A Novel Intelligent Approach for Dynamic Data Replication in Cloud Environment," IEEE Access, vol. 9, pp. 40240–40254, 2021, doi: 10.1109/ACCESS.2021.3064917.
[38] J. C. S. Dos Anjos et al., "Data processing model to perform big data analytics in hybrid infrastructures," IEEE Access, vol. 8, pp. 170281–170294, 2020, doi: 10.1109/ACCESS.2020.3023344.
[39] A. F. S. Devaraj, M. Elhoseny, S. Dhanasekaran, E. L. Lydia, and K. Shankar, "Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments," J Parallel Distrib Comput, vol. 142, pp. 36–45, Aug. 2020, doi: 10.1016/j.jpdc.2020.03.022.
[40] A. Alzuhair and A. Alghaihab, "The Design and Optimization of an Acoustic and Ambient Sensing AIoT Platform for Agricultural Applications," Sensors, vol. 23, no. 14, Jul. 2023, doi: 10.3390/s23146262.
[41] Y. Wang, Z. Chen, and Y. Deng, "Cloud Server-Assisted Remote Monitoring and Core Device Fault Identification for Dynamically Tuned Passive Power Filters," Applied Sciences (Switzerland), vol. 13, no. 17, Sep. 2023, doi: 10.3390/app13179830.
[42] M. A. Benblidia, B. Brik, M. Esseghir, and L. Merghem-Boulahia, "Power Allocation and Energy Cost Minimization in Cloud Data Centers Microgrids: A Two-Stage Optimization Approach," IEEE Access, vol. 10, pp. 66213–66226, 2022, doi: 10.1109/ACCESS.2022.3184721.
[43] S. de Vries, F. Blaauw, and V. Andrikopoulos, "Cost-Profiling Microservice Applications Using an APM Stack," Future Internet, vol. 15, no. 1, Jan. 2023, doi: 10.3390/fi15010037.
[44] J. Kim and C. Moon, "The Distributed HTAP Architecture for Real-Time Analysis and Updating of Point Cloud Data," Electronics (Switzerland), vol. 12, no. 18, Sep. 2023, doi: 10.3390/electronics12183959.