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

Distributed Transactions in Cloud Computing: A Review Reliability and Consistency

Barwar Mela Ferzo[1], Subhi R. M. Zeebaree[2]
barwar.ferzo@auas.edu.krd, subhi.rafeeq@dpu.edu.krd
[1]IT Dept., Technical College of Informatics-Akre, Akre University of Applied Sciences, Duhok, Iraq,
[2]Energy Eng. Dept., Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq,

Article Information

Submitted : 18 Mar 2024
Reviewed : 6 Apr 2024
Accepted : 30 May 2024

Keywords

Distributed Transactions, Cloud Computing, Reliability, Consistency

Abstract

The challenges of managing distributed transactions in cloud computing are discussed in this paper. The paper places an emphasis on the critical balance that must be maintained between reliability and consistency in the face of complexities such as hardware failures, network outages, and varying latencies. It sheds light on the delicate balance that must be maintained in order to guarantee that transactions in cloud environments are both reliable and consistent. Cloud environments are prone to hardware glitches and network disruptions. In addition, the paper delves into novel approaches with the objective of cultivating a computing ecosystem that is both resilient and dependable in the face of the ever-changing requirements of cloud computing, also a comparison table is presented for all the literature reviewed.

A. Introduction

In the realm of cloud computing, where scalability and flexibility reign supreme, the management of distributed transactions emerges as a critical challenge. These transactions involve multiple components or services spread across different geographical locations, demanding a delicate balance between reliability and consistency[1]. As cloud environments introduce complexities such as hardware failures, network outages, and varying latencies, ensuring that transactions are both reliable and consistent becomes imperative[2].

Reliability, a fundamental requirement for distributed systems, hinges on the system's ability to deliver accurate results consistently, even in the face of unforeseen failures. Cloud computing, prone to hardware glitches and network disruptions, necessitates robust mechanisms for seamless recovery to maintain functionality and data integrity. Consistency, another vital aspect, refers to the uniform agreement among all nodes in a distributed system regarding the outcome of a transaction [3]. Achieving consistency in the cloud is particularly challenging due to factors like network delays, partitioning, and asynchronous communication between components. Striking a balance between consistency, availability, and partition tolerance, as outlined by the CAP theorem, becomes a critical consideration in the design of distributed systems for the cloud[4].

The evolution toward cloud computing brings about architectural shifts, moving away from traditional monolithic structures to microservices and containerization. While this brings numerous advantages, it also intensifies the challenges associated with managing transactions across diverse services and components. Coordinating transactions spanning multiple services or databases is a significant challenge in a distributed environment. Traditional monolithic applications manage transactions within a single database, ensuring atomicity and consistency. However, cloud-based systems often require distributed transaction protocols to orchestrate and synchronize activities across independent services and databases[5].

Various distributed transaction models and protocols address these challenges. Classic approaches like two-phase commit (2PC) ensure atomicity but may encounter blocking issues[6]. More sophisticated protocols like three-phase commit (3PC) and Paxos offer advanced solutions but introduce their own complexities. Cloud-native databases often adopt eventual consistency models, prioritizing availability and partition tolerance over strict consistency. While suitable for some applications, this model may not align with scenarios where strong consistency is crucial, such as financial transactions.[7]

In conclusion, navigating distributed transactions in cloud computing demands a careful exploration of protocols and models. The dynamic nature of cloud resources, coupled with architectural shifts, underscores the ongoing pursuit of balancing reliability and consistency. As researchers and practitioners continually delve into innovative approaches, the goal remains to foster a resilient and dependable computing ecosystem amidst the evolving demands of cloud computing.

B. Background Theory

Distributed transactions play a pivotal role in the realm of cloud computing, where applications and services are often spread across multiple nodes and data centers. The seamless execution of transactions in a distributed environment is a critical factor for ensuring the reliability and consistency of cloud-based systems[8]. This paper aims to provide a comprehensive review of the challenges, solutions, and advancements in achieving reliable and consistent distributed transactions within the context of cloud computing. The emergence of cloud computing has transformed the landscape of modern IT infrastructure. Cloud services offer scalability, flexibility, and cost-efficiency, making them an attractive choice for hosting applications and managing data. However, the distributed nature of cloud environments introduces complexities related to data consistency and transaction reliability, necessitating a closer examination of distributed transaction mechanisms[9].

Distributed transactions involve the coordination of multiple entities or nodes in a network to ensure the consistent and reliable execution of operations across a distributed environment. Understanding the fundamentals of distributed transactions is essential for addressing the challenges associated with the dynamic and interconnected nature of cloud computing[10].

Effective communication and coordination are crucial for the success of distributed transactions. Nodes need to exchange information and reach a consensus on the outcome of the transaction. Communication may be hindered by network latency, potential failures, or partial unavailability of nodes[11].

Atomicity is one of the ACID properties (Atomicity, Consistency, Isolation, Durability) that ensures that a distributed transaction is treated as a single, indivisible unit[12]. Either all operations within the transaction are successfully completed, or none of them are. This property is essential for maintaining data integrity in the face of failures[13].

Consistency ensures that the system transitions from one consistent state to another after the execution of a distributed transaction. In other words, the transaction brings the system from one valid state to another, preserving the integrity of the data. Achieving consistency becomes challenging when multiple nodes are involved, and failures can occur[14].

Different transaction models, such as distributed transactional models and long-running transactions, cater to specific requirements in cloud computing. These models address scenarios where transactions span multiple services, involve complex workflows, or require coordination beyond the capabilities of traditional transactional systems[15].

Reliability in distributed transactions encompasses various aspects, including fault tolerance, durability, and the ability to recover from failures[16]. In cloud computing, where hardware and network failures are not uncommon, ensuring the reliability of distributed transactions becomes a paramount concern. Traditional two-phase commit protocols have been widely employed, but their limitations, such as blocking and vulnerability to coordinator failures, have prompted the exploration of more sophisticated approaches[17].

In recent years, various distributed transaction protocols and frameworks have been proposed to address the challenges of reliability and consistency in cloud computing environments. Innovations include the use of consensus algorithms, such as Paxos and Raft, that provide a robust foundation for achieving agreement among distributed nodes[18]. Additionally, distributed databases and middleware solutions have introduced novel techniques like sharding and replication to enhance the scalability and fault tolerance of distributed transactions[19].

The rise of blockchain technology has introduced a decentralized approach to achieving reliability and consistency in distributed transactions[20]. Blockchain, a distributed ledger that is tamper-resistant and transparent, ensures the immutability of transactions. Smart contracts, self-executing scripts deployed on blockchain networks, further enhance the automation and reliability of distributed transactions. The decentralized nature of blockchain mitigates the risks associated with centralized control and single points of failure[21].

Understanding these fundamental aspects of distributed transactions provides the groundwork for exploring advanced protocols, frameworks, and technologies aimed at overcoming the challenges posed by the distributed nature of cloud computing environments. The subsequent sections of this paper will delve into specific distributed transaction protocols and their implications for reliability and consistency in cloud-based systems[22], [23].

C. Literature Review

For the security challenges associated with storing and maintaining data in cloud databases. [24] proposed multi-level security architecture aimed at addressing these challenges and improving the consistency of data transactions in the cloud. The architecture includes Authentication Authorization and Auditing (AAA), Depth 1 Fixed Tree-Based Consistency (D1FTBC) approach, Cloud Service Controller (CSC), Server and Cloud Data Locker (CDL). The main focus is on ensuring both consistency and security levels in cloud transactions. The analysis includes measurements of execution time in various aspects such as service authentication, user authentication, secure communication, data transaction analysis, data level access time with security, hit ratio latency and system throughput. The performance analysis confirms the efficient functioning of the proposed architecture with advanced performance. Empirical evaluation shows positive results in latency analysis, and consistency is improved through the components of the D1FTBC approach. The research aims to strengthen data transaction services in the cloud environment

STREAMDB is a system that models transaction execution as event processing within a stream processing system. It was created because it is hard to scale transactions for large-scale cloud applications (Chen & Migliavacca, 2019)[25]. Because it uses a dataflow style to run transactions and shows them as acyclic transaction graphs, STREAMDB is not the same as other DBMS. This method cut down on the costs of distributed execution overhead, such as locking and latching. It also lets multiple tasks run at the same time without the need for distributed transactions. It has three types of operators: Source Operators, Data Operators, and Sink Operators. Source Operators handle transactions, Data Operators make sure that data is correct, and Sink Operators send responses. It also has a global transaction graph and protocols for controlling concurrency that are based on timestamps. These make sure that transactions can be serialized and that event streams from multiple transactions can be handled at the same time. Finally, STREAMDB's special way of handling transactions and controlling concurrency solves the issues that come up when trying to make transactions work for big cloud apps. It does this by giving a solution that works better and can be changed more easily than regular DBMS. Each node can hold more than one site, and each site has a STREAMDB Operator that stores data tables. It is safe for TCP/IP sockets to send transaction events, and the Timestamp Sequencer makes sure that events happen in the right order and that data stays the same. The prototype shows improvements in performance, but it needs to be tested in more modern settings and applications. It was found that StreamDB's throughput goes up linearly with the number of warehouses, and with the best partitioning strategies, it can reach a maximum of 30K TPS.

[26] introduces 2PC*, a new distributed transaction control protocol specifically designed for a microservice architecture. The need for a solution to achieve strong data consistency and high processing performance under scenarios of extremely high numbers of concurrent user requests is highlighted. This new protocol, 2PC*, features a two-level asynchronous lock, a novel optimistic lock named SAOL, and a runtime protocol for transaction concurrency control. evaluation of the performance of 2PC* compared to 2PC in workloads with varying degrees of contention, they highlighting the significant improvement in throughput achieved by 2PC*. Compared to earlier methods for multi-microservice, 2PC* can extract more concurrent processing capabilities under high-intensity competitive workloads. They added a secondary asynchronous optimistic lock to avoid locks during the transaction process and reduce the likelihood of concurrent conflicts between multiple transactions. Experimental results demonstrate higher throughput and lower latency in high-level contention scenarios, indicating that 2PC* allows for greater concurrency across multiple microservices compared to the original 2PC.

[27]presents the idea of offloading computing as a way to deal with resource scarcity in mobile hosts. The concept of SOMC (surrogate object in mobile cloud) involves the use of surrogate objects to facilitate dynamic task offloading across a network of connected mobile hosts. It presents a dynamic task offloading algorithm and describes the task offloading decision-making process under wireless communication based on priority task scheduling. Representing mobile hosts, surrogate objects housed on mobile support stations actively participate in mobile data access by upholding mobile host details and guaranteeing appropriate data transmission. The main insight is that if an application takes a significant amount of time to compute on the mobile host, leading to high energy consumption and battery usage, and exceeds the minimum threshold value, it is more effective to offload the application onto the cloud. The model comprises components that generate read-only transactions requesting specific or frequently used data items from a database server. The proposed algorithm involves initializing the data center broker for MSS, analyzing the status, selecting virtual machines, sending cloudlets for processing, and updating statuses. The results show that the proposed algorithm has better response times compared to existing models without task offloading. The DTO model is designed to divide transactions

into sub-transactions and transmit them to surrogate objects for local validation, which differs from existing models where transactions are executed globally. The proposed model aims to optimize network traffic, minimize access, achieve low response times, and improve quality of service.

Balusamy and Krishna (2020) [28]came up with the idea of Secured Access Provision with Effective Role Management (SAPERM) to show how important cloud security is becoming and how hard it is to control who can access what in the cloud. This plan will make it easier for big companies to let people into their systems and will also make the systems faster and safer in the future. SAPERM manages cloud-based businesses well by using attribute-based encryption (ABE) and role-based access control (RBAC). Bugs like duplicate user attributes and less precise access for users are fixed so that this can happen. It's important for larger transaction processing systems that the plan gives standard users access at higher process threshold rates. Its distributed nature also makes it easier to do calculations. Priority-based session management, giving different system administrators more work, and good algorithms for saving and retrieving data are all parts of the model that is being put forward to make the system work better. Researchers have found that the SAPERM scheme is better at meeting most needs for security and access control. It's built this way. Keep up with the structure of user logs to improve security and data access to cloud data. This leads to better control over security and access. It was possible to measure CSI latency, which is the time between when a user asks for something and when a server responds. With SAPERM, the CSI latency stayed the same even when more than 8,000 users made requests at the same time. With other schemes, the latency went up a lot. One study found that SAPERM had less average latency than RBAC and ABE when it came to handling 1,000 requests at once. The difference was between 0.15 and 2 seconds. Also, RBAC and ABE could only handle 7,000 and 8,000 requests at the same time before they ran out of work to do because their system procedures were too complicated. But SAPERM stayed the same.

To overcome the limitations of using blockchain for online transactional processing (OLTP) workloads and the outstanding limitations of current cross-blockchain solutions which are centralized brokers, two-party transactions, performance issues, and interface limitations, [29] introduces a new protocol, cross-blockchain transaction (CBT), designed to overcome these limitations by eliminating the centralized component and ensuring non-blocking transactions. The researchers executed the Coordinated Blockchain Transaction (CBT) on a 3-blockchain scenario and showed its extension to an arbitrary number of blockchains. Also to overcome the limitations and blocking scenarios of the Two-Phase Commit (2PC) protocol, , an Improved 2PC protocol with a heartbeat monitoring mechanism is proposed, along with an Interactive Recovery Protocol for handling uncertainties. The CBT protocol is found to be nonblocking in contrast to 2PC, as demonstrated in an experiment where the coordinator failure causes 2PC to block while CBT continues. Results also showed that CBT's scalability was nearly linear scalability and minimal overhead compared to 2PC, Additionally, CBT is compared with AC3 in cross-blockchain transactions, demonstrating its superior performance and scalability, especially at larger scales.

Fog computing tries to lower traffic on the backbone, lower latency for IoT apps, and improve the user experience. But it's hard to make good schedules for transactional services in the cloud because applications that do a lot of computation have a lot of latency and change quickly. To deal with the problem of allocating resources for fog computing in IoT systems, the price of sending data from faraway IoT sensors to servers, and the need to handle multiple transactions at the fog end to lower the amount of data sent, Al-Qerem et al. (2020)[30] suggest a collaborative way to check transactions partially at the fog and globally at the cloud server. The goal is to find conflicts quickly, save processing power, and cut down on communication delays. They proposed a system called "partial validation" that lets the fog node handle read-only IoT transactions locally. Only update transactions are sent to the cloud for final approval. The study looks at the partial validation procedure using three different concurrency protocols and finds that the miss rate, restart rate, and communication delay all go down. This study also used computer simulations to look at how well the cooperative control of concurrency protocol with partial validation worked at the fog node and in regular cloud environments. The results show that the suggested method cuts down on communication delays by a large amount and makes low-latency fog computing services possible for IoT applications. The researchers stressed how important the proposed protocol was for managing multiple transactions at the fog end, cutting down on the amount of data sent over the network, and making the system work better by lowering the rate of misses.

There are many challenges for achieving integration for security and consistency in cloud environments and cloud computing in IT industries with its various services such as PaaS, SaaS, IaaS, and DaaS. Especially challenges related to data transaction systems in cloud computing particularly security and consistency issues. Hence [31] proposed an integrated architecture for secured data transactions with enhanced consistency in the cloud environment. An integrated architecture was setup using Visual Studio 2012 ASP.NET as a cloud data transactional service and or creating web pages with user interaction, deployed in the Azure platform. Various servers are tested to ensure data transaction security and consistency in the cloud environment such as CSC server, D1FTBC server, AAA server, Cloud Storage server, CDL and cloud DB. Management of authentication protocols, IASCDTCE was used to validate user identities at local and remote levels , however, in the experimental result there was an indication of the relationship between processing times and the number of requests, showing that the processing time for retrieving encrypted user credentials is slightly higher than for decrypting and encrypting user credentials. The system shows linearly growing processing time between 600 to 1000 requests compared to 100 to 500 requests, furthermore, the comparison of component performances reveals that Decrypt SSRS DB has a maximum processing time of 39.09793 ms, while VMM Profiler has a minimum processing time of 21.9475 ms for 1000 service requests.

In the recent years the growing popularity of cloud computing and its potential to enable innovative solutions and services leading to cost-effective and flexible services, particularly focusing on storage as a service (STaaS).[32] proposed system for building a private cloud using the open source software OpenStack also the proposed system architecture includes various modules such as Keystone, Nova, Horizon, Swift and Glance. Each providing deferent purpose that are: identity and access policy services for all components in the OpenStack family; Computing Fabric controller for the OpenStack Cloud; managing images, volumes, instances, key pairs, and Swift containers; store a large number of virtual objects; catalog service for storing and querying virtual disk images; respectively. The proposed system utilizes OpenStack and Ubuntu operating system to implement three nodes: Compute, Controller, and Storage, each node is installed with specific packages and services to fulfill its role, with Compute node having Nova packages, Controller node having packages of Keystone, Glance, and Horizon, and Storage node having Cinder packages or Swift packages. Furthermore, the AES (Advanced Encryption Standard) algorithm was presented, highlighting the fact that ciphertext, plaintext, and cipher key are all handled as byte arrays. Because the AES algorithm uses a polynomial representation to represent byte values as finite field elements, the number of transformation rounds and the final round's variation from the previous rounds are dependent on the length of the cipher key. The transaction will be committed by the system if it completed without any errors. All data manipulations carried out inside the transaction's boundaries are carried out and saved to the cloud database during a transaction commit operation. The changes made to the data within the transaction are not saved to the database in the event of a transaction error or if the user specifically requests a rollback operation. Partial transactions are never allowed to be committed to the database because doing so would cause the database to become inconsistent.

The rise of serverless computing, specifically the Function-as-a-Service (FaaS) model, and the challenges it faces with consistency in storage layers. The trade-offs between performance and consistency in FaaS implementations, such as AWS Lambda, and the potential consequences of weak consistency, including capital loss ,also, the need for stronger consistency levels in FaaS with studying of Transactional Causal Consistency (TCC) as a potential solution led to [33] to propose FaaSTCC, which enhances FaaS middleware with Transactional Causal Consistency (TCC). FaaSTCC aims to combine the positive side of both approaches by using a caching layer to reduce unnecessary storage accesses and a TCC storage layer with mechanisms to facilitate consistency checking, it consists of four main components: client library, caching layer, augmented TCC storage layer, and coordination layer. Different scenarios are presented, illustrating how FaaSTCC determines the consistency of values in the cache and when to refresh from the storage system. The researchers showed the relevance of using promises and snapshot intervals to maximize cache utilization in FaaSTCC, pecifically, promises contribute to 29% of the gains, while snapshot intervals bring an additional 23% reduction in latency. FaaSTCC demonstrates consistent performance for both transaction types, leveraging the storage layer to avoid aborts, while HydroCache had transaction aborts of nearly 9% of the time.

[34] proposed a design of FaaSSI, a system aimed at providing strong transactional support in a Function as a Service (FaaS) environment while maintaining its advantages, the system is designed to scale with executor nodes and impose minimal cost overheads. FaaSSI was built as an extension to Cloudburst and as a storage layer Anna was used that supports eventual consistency, this design allows users to choose the appropriate consistency model

for their applications within a single system. The system relies on an intermediate layer for concurrency control and implements Snapshot Isolation as its primary focus for strong consistency. The system was evaluated on the Grid'5000 experimental platform using dedicated servers with specific hardware configurations (Each server is composed of 1 Intel Xeon Gold 5220 CPU with 18 cores, $96GB$ RAM and $480GB$ ofSSD storage). The experimental load was based on previous test benches, with 12 concurrent clients executing function graph requests sequentially, involving read and write operations with specific data set characteristics. The results show that enforcing snapshot isolation can lead to up to 4× the latency of the eventual consistent solution. he performance of FaaSSI degrades with an increase in skew, while the eventual consistency system shows a slight improvement in performance under the same conditions.

JointCloud has a huge impact on small and medium enterprises (SMEs) becoming cloud services customers (CSCs) and the increasing payment activities between cloud service providers (CSPs) and CSCs. Thus [35] proposed JCLedger, a Blockchain-based distributed ledger for JointCloud, aims to enhance reliability and convenience in cloud resource and value exchange, proof of previous transaction (PoPT) as a more efficient alternative to proof of work (PoW) for JCLedger, the PoPT consensus algorithm, which selects accountant candidates based on user participation in previous transactions. The simulation in JCLedger involved 250 block heights, with 250 accounting nodes randomly sending transactions to each other, It specifies the settings for the simulation, including 50 accountant candidates and a maximum of 200 transactions in a block with specific parameters for decision-making by nodes. The number of blocks for each block height dynamically changes based on historical data, and parallel accounting can be used in conjunction with micropayment channels and sharding to solve the scalability problem.

The challenges of outsourcing data mining to a cloud service provider (CSP) due to privacy risks and the potential violation of individuals' privacy and also the security concerns related to CSP being a single point of attack and the challenges of conducting operations on encrypted data for cloud servers, led to [36] to propose a model consisting of three components: Data owners (DO), Intermediate cloud servers (ICS), and Frequent item-sets computing cloud server (FCCS). Firstly, the double encryption algorithm (DEA) was used by the Data owners in order to encrypt their data so that it would be protect from ICS and FCCS. Second, the Transaction Splitter algorithm (TSA) is used to divide and allocate encrypted datasets to N clouds. In order to find frequent global itemsets, ICS gathers transactions, computes local frequent itemsets using a customized Apriori algorithm, and sends the results to FCCS. By comparing this with less secure techniques, the results of multiple tests conducted on real-world data sets showed that a high level of privacy could be achieved without disclosing user data. Furthermore, a tailored Apriori method is introduced to reduce the necessity of examining the complete database, leading to more accurate and quicker mining outcomes.

Cloud computing is seen as a solution offering better storage, reduced system overhead, and improved data transaction services, security concerns, including the limitations of traditional methods and the potential for data intrusion led to seeing blockchain as a promising approach for enhancing security in cloud-based systems. [37] introduced the SEC-LearningChain, a secure framework for cloud computing that combines blockchain, machine learning (ML), and cloud computing technologies to ensure data integrity, availability, and efficiency. The architecture comprises four design models: attack detection, blockchain-based identity verification, ML-based transaction record optimization, and cloud assessment. Simulation parameter settings for a CPU-based computer system with specific hardware and software configurations(RAM of 8GB, 2GHz Intel core i7, and 256GB memory), the simulation consisted of four different attacks in a peer-to-peer network to evaluate attack strength based on computational resources. Key parameters such as learning rate, number of iterations, minimal batch size, and number of epochs are specified to enhance the utility of services until stable values are reached. The result show that the proposed method had achieved low latency, detects attacks by monitoring variations in throughput and performs more quickly than SH-BlockCC; The comparison results show that SEC-LearningChain achieves higher throughput in both attack and normal cases. Comparisons with existing schemes show that the proposed model offers better utility for miners and effectively utilizes CPU cycles for tasks.

The emergence of distributed energy generation has enabled individuals and companies to engage in energy trading, facilitated by blockchain technology.[38] presents integrating the Practical Byzantine Fault Tolerance (pBFT) model with the private Hyperledger Sawtooth blockchain to minimize transaction latency and predict transaction time. Additionally, the researchers proposed a transaction time controllability model to accommodate participants' transaction constraints, aiming for more flexible and user-friendly transactions. Less than one-third of all nodes must be malicious in order for the pBFT system to function. The CPLEX optimization package was used to conduct the simulation in Python, giving blockchain participants the ability to choose and accept the minimum transaction time depending on desired constraints. Twelve people took part in the simulation, which compared the transaction times in public, private, and P-pBFT blockchain models. The findings indicate that P-pBFT reduces transaction times by 23.8% and 55.56%, respectively, when compared to private and public blockchains.

The challenges of distributed transaction processing in modern database systems, the impact of inter-DC latency, existing protocols for reducing network round-trips, the limitations of RDMA in heterogeneous networks was the element that made [39] introduced RedT, an RDMA-enhanced distributed transaction processing protocol designed to reduce inter-DC round-trips and improve system performance. Each node in the system is equipped with Txn Workers and MemStore, RedT executes distributed transactions in two phases: the execution phase and the commit phase. experimental setups and benchmarks (YCSB and TPC-C) used in the study. The experimental result showed that:1) Impact of Inter-DC Transaction Ratio: RedT outperforms TAPIR, EP-Paxos, and 2PC-Paxos, showing the best performance 2) Impact of Inter-DC Networks: RedT outperforms other protocols in terms of throughput and latency due to its reduced exposure to inter-DC networks and fewer inter-DC communications 3) Impact of Read-Write Ratio: RedT's performance remains insensitive to changing write-ratios due to its imperceptible overheads compared to the cost of inter-DC round-trips 4) Impact of Contention Level: RedT and MDCC minimize round-trips and outperform others.

the increasing use of cloud computing to modernize transaction systems, specifically focusing on the application of Infrastructure as a Service (IaaS) to enhance cashier services.[40] proposed a system, focusing on modeling activities such as architectural modeling, system modeling, and modeling for database using Unified Modeling Language (UML) diagrams. The application of cloud computing to cashiers using Microsoft Azure is also mentioned, highlighting the installation and configuration of IaaS, implementation of the cashier system, and trial conduct. The results highlight the suitability of Azure IaaS for cloud computing due to its ease of use and power, It also emphasizes the accessibility of existing public IP addresses via the internet.

For the the growing importance of edge computing in the context of the Internet of Things (IoT) and its integration with cloud computing.[41] proposed framework for IoTT (internet of things transaction) processing and the model for studying concurrency control techniques in IoT transaction processing, The key QoS parameters include security, cost, service time, energy usage, reliability, and availability, all of which should be addressed by IoT offerings. The potential of transaction processing in Social Web of Things (SWoT) for various industries, such as smart home automation, industrial automation, healthcare monitoring, and transportation management, is also outlined. The researcher’s analysis simulation of three execution approaches for IoTT processing under varying network conditions including Cloud Host Execution Strategy (CHS), Edge Host Execution Strategy (EHS), and Both Execution Strategy (BHS). When an edge host does not experience a disconnection, the results are shown, It is clear that out of the three strategies, the CHS is the most effective, and the EHS is the least effective, while the BHS is in the middle of them as far as the effectiveness is considered.

The ease and cost-effectiveness of accessing cloud services when it comes to sharing information, however, it has challenges and concerns about data security.[42] proposed a solution for enhancing the security of the payment process by introducing a certified authority between the customer and the merchant. The proposed method hides the customer's password within cover text using text steganography, two shares are generated from the snapshot of the text, with one share given to the customer and the other to the certified authority (CA). Upon receiving the customer's share, the CA combines it with their share to retrieve the original image.

D. Discussion and Comparison

In this section we provide a comparison table shown in table (1) discussing the important aspects about the reviewed literature in term of proposed methods, tools, results, main idea and limitation of the work done. The reviewed papers each wok on deferent aspect of the distributed system transaction , having some working on enhancing security and privacy od distributed transaction such as [24][28] [32] [36] and [42], while others focused more on internet of thing (IOT) and using cloud as a service for example [30] [33] [34] [40] and [41] , while the improvement of database and mobile applications were covered by [25] [27] and [39] , as for the blockchain transaction were presented in [29] [35] [37] and [38].

Table 1. A Comparison Among The Work Reviewed

Ref. No.	**Proposed method**	**Tools**	**Results**	**Main Idea**	**Limitation**
[24] 2019	proposed multi-level security architecture	The architecture includes, Authentication Authorization and Auditing (AAA) Server, Depth 1 Fixed Tree-Based Consistency (D1FTBC) approach, Cloud Data Locker (CDL) and Cloud Service Controller (CSC),	Empirical evaluation shows positive results in latency analysis, and consistency is improved through the components of the D1FTBC approach.	Aimed at addressing security challenges and improving the consistency of data transactions in the cloud.	Dose not support biometric authentication.
[25] 2019	introduced STREAMDB, a system that models transaction execution as event processing within a stream processing system.	It has three types of operators: Source Operators, Data Operators, and Sink Operators.	The throughput of StreamDB is found to increase linearly with the number of warehouses, reaching a maximum of 30K TPS with optimized partitioning strategies. The prototype demonstrates performance gains but requires further validation for different modern applications and environments.	STREAMDB uses a dataflow style for transaction execution, representing transactions as acyclic transaction graphs.	Did not experiment with more complex applications and environments,
[26] 2020	introduces 2PC*, a new distributed transaction control protocol	This new protocol, 2PC*, features a two-level asynchronous lock, a novel optimistic lock named SAOL, and a runtime protocol for transaction concurrency control.	Compared to the original 2PC, 2PC* allows for greater concurrency across multiple microservices, as demonstrated in experimental results showing higher throughput and lower latency in high-level	Specifically designed for a microservice architecture	Not applies on DevOps, PaaS cloud platform. In the IoT (Internet of Things). QoS (quality of service) needs improvement

			contention scenarios.		
[27] 2020	SOMC (surrogate object in mobile cloud) was proposes as a dynamic task offloading in a network of connected mobile hosts using surrogate objects.	The proposed algorithm involves initializing the data center broker for MSS, analyzing the status, selecting virtual machines, sending cloudlets for processing, and updating statuses.	The results show that the proposed algorithm has better response times compared to existing models without task offloading. The DTO model is designed to divide transactions into sub-transactions and transmit them to surrogate objects for local validation, which differs from existing models where transactions are executed globally	The proposed model aims to optimize network traffic, minimize access, achieve low response times, and improve quality of service.	Problems like resource management, which distributes the work among the available service providers.
[28] 2020	proposed Secured Access Provision with Effective Role Management (SAPERM)	SAPERM manages cloud-based businesses well by using attribute-based encryption (ABE) and role-based access control (RBAC).	The CSI latency of SAPERM remained consistent even after exceeding 8,000 concurrent user requests, while traditional schemes experienced drastic increases in latency.	a scheme that aims to establish a clear framework for user access provision in large enterprises, achieving forward and backward security and better system performance.	been validated using open stack Nova, an open source cloud platform but not in a mission critical applications in multi-cloud environments. It also have issues related to energy efficiency.
[29] 2020	introduces a new protocol, cross-blockchain transaction (CBT), designed to overcome these	an Improved 2PC protocol with a heartbeat monitoring mechanism is proposed, along with an Interactive Recovery Protocol	Results showed that CBT's scalability was nearly linear scalability and minimal overhead compared to 2PC,	Overcoming the limitations and blocking scenarios of the Two-Phase Commit (2PC) protocol	Currently the CBT protocol only perform on scale of couple tens of nods

	limitations by eliminating the centralized component and ensuring non-blocking transactions.	for handling uncertainties.	Additionally, CBT is compared with AC3 in cross-blockchain transactions, demonstrating its superior performance and scalability, especially at larger scales.		
[30] 2020	proposed a system called "partial validation" that lets the fog node handle read-only IoT transactions locally. Only update transactions are sent to the cloud for final approval.	The study looks at the partial validation procedure using three different concurrency protocols namely AOCCRBSC, AOCCRB and STUBcast.	The results show that the suggested method cuts down on communication delays by a large amount and makes low-latency fog computing services possible for IoT applications.	finds that the miss rate, restart rate, and communication delay all go down. The researchers stressed how important the proposed protocol was for managing multiple transactions at the fog end, cutting down on the amount of data sent over the network, and making the system work better by lowering the rate of misses.	Lack of Real-World Implementation: Investigating the practical implementation of the proposed protocol in real-world fog-cloud environments to assess its effectiveness and scalability.
[31] 2021	suggested an integrated architecture for cloud-based, secure data transactions with improved consistency.	Utilizing ASP.NET as a cloud data transactional service and the Azure platform, an integrated architecture was set up. Various servers are tested to ensure data transaction security and consistency in the cloud environment such as CSC server, D1FTBC server, AAA server, Cloud Storage server,	The system shows linearly growing processing time between 600 to 1000 requests compared to 100 to 500 requests, furthermore, the comparison of component performances reveals that Decrypt SSRS DB has a maximum processing time of 39.09793 ms,	Management of authentication protocols, IASCDTCE was used to validate user identities at local and remote levels	The proposed architecture was tested with moderate load and not a heavy load

		CDL and cloud DB.	while VMM Profiler has a minimum processing time of 21.9475 ms for 1000 service requests.		
[32] 2021	proposed system for building a private cloud using the open source software OpenStack	the proposed system architecture includes various modules such as Keystone, Nova, Horizon, Swift and Glance. Each providing deferent purpose	If the transaction executed without any errors, the system will commit the transaction. A transaction commit operation executes and saves all data manipulations performed within the transaction's boundaries to the cloud database.	In the event of an error during the transaction or if the user explicitly requests a rollback operation, the modifications made to the data within the transaction are not saved to the database.	No actual test and experiment were done in the real world only hypothesis.
[33] 2021	propose FaaSTCC, which enhances	FaaS middleware with Transactional Causal Consistency (TCC).	The researchers showed the relevance of using promises and snapshot intervals to maximize cache utilization in FaaSTCC, pecifically, promises contribute to 29% of the gains, while snapshot intervals bring an additional 23% reduction in latency.	FaaSTCC aims to combine the positive side of both approaches by using a caching layer to reduce unnecessary storage accesses and a TCC storage layer with mechanisms to facilitate consistency checking.	focused only on Transactional Causal Consistency but not other consistency criteria such as such as Snapshot Isolation.
[34] 2021	proposed a design of FaaSSI,	The system was evaluated on the Grid'5000 experimental platform using dedicated servers with specific hardware configurations (Each server is composed of 1	The results show that enforcing snapshot isolation can lead to up to 4× the latency of the eventual consistent solution. he performance of	The system aimed at providing strong transactional support in a Function as a Service (FaaS) environment while maintaining its	The result are not very promising and can not be properly be comperd with other researches since the the Grid'5000

		Intel Xeon Gold 5220 CPU with 18 cores, 96GB RAM and 480GB ofSSD storage).	FaaSSI degrades with an increase in skew, while the eventual consistency system shows a slight improvement in performance under the same conditions.	advantages, the system is designed to scale with executor nodes and impose minimal cost overheads.	is an experimental platform
[35] 2021	proposed JCLedger, a Blockchain-based distributed ledger for JointCloud,	The simulation in JCLedger involved 250 block heights, with 250 accounting nodes randomly sending transactions to each other	Results are not predictable	aims to enhance reliability and convenience in cloud resource and value exchange, proof of previous transaction (PoPT) as a more efficient alternative to proof of work (PoW) for JCLedger,	Results are not predictable
[36] 2022	to propose a model to encrypt data for cloud servers	consisting of three components: Data owners (DO), Intermediate cloud servers (ICS), and Frequent item-sets computing cloud server (FCCS).	result indicated achieving a high level of privacy without revealing users' data, contrasting this with less secure techniques.	a customized Apriori approach is presented to minimize the need for exploring the entire database, resulting in faster and correct mining results.	availability of more effective non-privacy algorithms
[37] 2022	Proposed the SEC-LearningChain,	Simulation parameter settings for a CPU-based computer system with specific hardware and software configurations(RAM of 8GB, 2GHz Intel core i7, and 256GB memory).	The system demonstrates improved performance in terms of mining size, utility for miners, and secure data transmission in cloud-based systems. The experimental analysis indicates better resource utilization and secure data	a secure framework for cloud computing that combines blockchain, machine learning (ML), and cloud computing technologies to ensure data integrity, availability, and efficiency.	The suggested work fails in unexpected circumstances because anonymous transactions prevent the node or user from tracing it outside the network.

			transmission compared to existing approaches.		
[38] 2022	presents integrating the Practical Byzantine Fault Tolerance (pBFT) model with the private Hyperledger Sawtooth blockchain.	The CPLEX optimization package was used to conduct the simulation in Python	The results show that P-pBFT reduces transaction time by 23.8% and 55.56% compared to private and public blockchains, respectively.	minimize transaction latency and predict transaction time. It offers flexibility in regulating node population and transaction size, allowing members to manage and control transaction time.	The pBFT system requires the number of malicious nodes to be less than one-third of the total node population. Any changes in these ratio will negatively effect the results
[39] 2023	introduced RedT, an RDMA-enhanced distributed transaction processing protocol	Each node in the system is equipped with Txn Workers and MemStore, RedT executes distributed transactions in two phases. experimental setups and benchmarks (YCSB and TPC-C) used in the study	The experimental result showed that in regards of Impact of Inter-DC Transaction Ratio, Impact of Inter-DC Networks, Impact of Read-Write Ratio and Impact of Contention Level RedT outperforms other protocols	designed to reduce inter-DC round-trips and improve system performance.	—
[40] 2023	Suggested a system with an emphasis on modeling tasks using Unified Modeling Language (UML) diagrams, including system, architectural, and database modeling.	Unified Modeling Language (UML) diagrams and Microsoft Azure	The results highlights the suitability of Azure IaaS for cloud computing due to its ease of use and power.	Specifically focusing on the application of Infrastructure as a Service (IaaS) to enhance cashier services	Relatively low security
[41] 2023	Proposed framework for IoTT (internet of things	Analysis simulation of three execution approaches	When an edge host does not experience a disconnection,	Studying concurrency control techniques in	Complicate d transaction manageme

	transaction) processing	including Cloud Host Execution Strategy (CHS), Edge Host Execution Strategy (EHS), and Both Execution Strategy (BHS).	the results are shown, It is clear that out of the three strategies, the CHS is the most effective, and the EHS is the least effective.	IoT transaction processing	nt algorithms have the potential to reduce system overhead and processing time, which will lower system performance.
[42] 2023	proposed a solution for enhancing the security of the payment process	Hideing the customer's password within cover text using text steganography	No actual test were conducted to see the effectiveness of the proposed method	introducing a certified authority between the customer and the merchant.	The technique only addresses customer data security and identity theft prevention.

E. Extracted Statistics

The extracted statistics from this research shed light on the diverse array of tools employed in investigating the intricacies of distributed transactions within the realm of cloud computing. The study extensively delves into various methodologies and frameworks leveraged by researchers to explore the nuances of reliability and consistency in distributed transactions. Notable tools include distributed databases, consensus algorithms, and transaction management systems, which collectively contribute to the comprehensive understanding of the challenges and solutions in this dynamic field. These statistics underscore the interdisciplinary nature of the research, highlighting the importance of a multifaceted approach that integrates a variety of tools to address the complexities inherent in ensuring the reliability and consistency of distributed transactions in cloud computing environments.

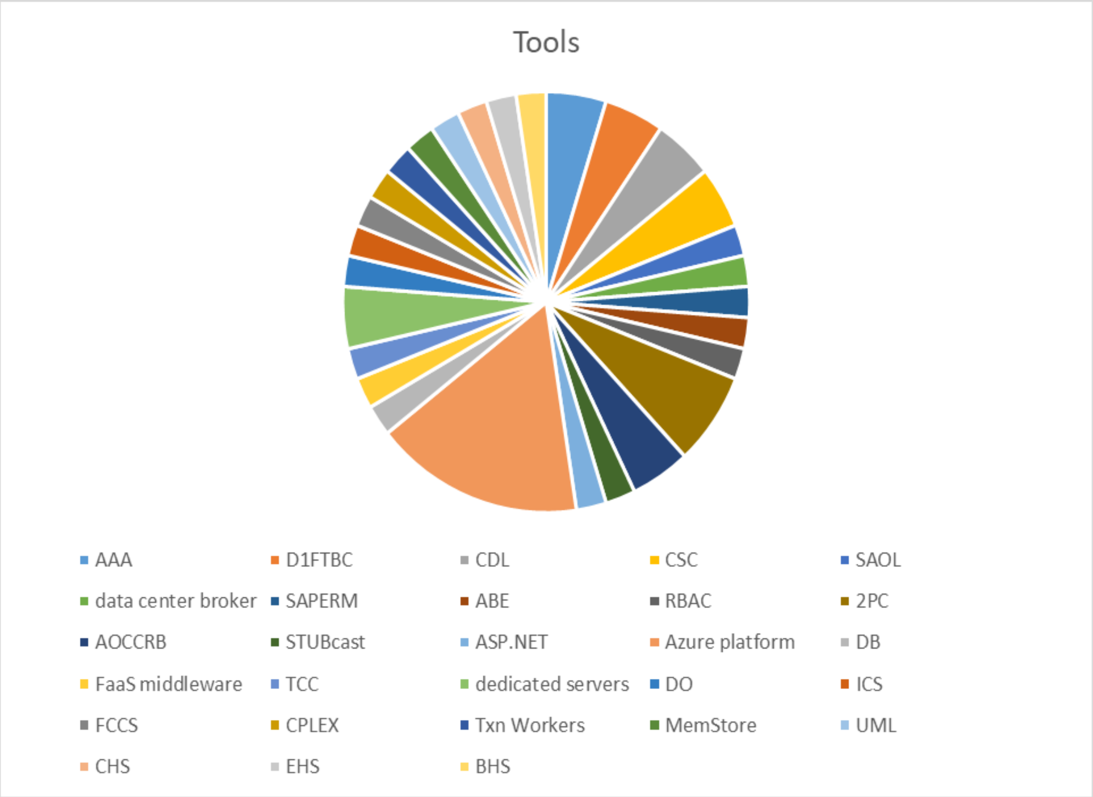

Figure 1. The most tools employed in investigating the intricacies of distributed transactions within the realm of cloud computing**.**

F. Recommendation

Utilize Sharding and Replication:

Leverage sharding and replication techniques in distributed databases to improve scalability and fault tolerance. These mechanisms distribute data across multiple nodes and replicate it to ensure availability and durability, contributing to the reliability of distributed transactions.

Optimize for Cloud Environments:

Tailor distributed transaction solutions to the specific characteristics of cloud environments. Consider factors such as dynamic resource provisioning, varying network latencies, and the heterogeneity of cloud infrastructure when designing and implementing reliability and consistency mechanisms.

Address Network Latency:

Mitigate the impact of network latency on distributed transactions by optimizing communication protocols and employing techniques such as asynchronous messaging. Implementing strategies to handle varying latencies across geographically dispersed nodes is crucial for maintaining transaction reliability.

Explore Eventual Consistency Models:

Assess the suitability of eventual consistency models for specific use cases. While strong consistency is desirable, eventual consistency provides a pragmatic approach in scenarios where immediate consistency is not a strict requirement, offering improved availability and fault tolerance.

Monitor and Analyze Performance:

Establish robust monitoring and analysis tools to continuously assess the performance of distributed transactions in the cloud. Monitor key metrics, identify bottlenecks, and conduct thorough performance evaluations to optimize the efficiency and reliability of the implemented solutions.

By incorporating these recommendations and researchers can contribute to the ongoing improvement of distributed transactions in cloud computing, ensuring higher levels of reliability and consistency in the face of the evolving challenges posed by distributed and dynamic cloud environments.

G. Conclusion

The paper discusses the challenges of managing distributed transactions in cloud computing, emphasizing the critical balance between reliability and consistency in the face of complexities such as hardware failures, network outages, and varying latencies. It highlights the delicate balance required to ensure that transactions are both reliable and consistent in cloud environments, which are prone to hardware glitches and network disruptions. The paper also delves into innovative approaches, aiming to foster a resilient and dependable computing ecosystem amidst the evolving demands of cloud computing.

H. References

[1] R. Makhlouf, "Cloudy transaction costs: a dive into cloud computing economics," *Journal of Cloud Computing*, vol. 9, pp. 1–11, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:210166542

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

[3] A. P. Marathe, "DBMS Performance Troubleshooting in Cloud Computing Using Transaction Clustering," in *International Conference on Extending Database Technology*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232283641

[4] R. Makhlouf, "Cloudy transaction costs: a dive into cloud computing economics," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–11, 2020.

[5] D. A. Hasan, B. 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," *International Journal of Science and Business*, vol. 5, no. 6, pp. 33–44, 2021.

[6] P. Fan, J. Liu, W. Yin, H. Wang, X. Chen, and H. Sun, "2PC*: a distributed transaction concurrency control protocol of multi-microservice based on cloud computing platform," *Journal of Cloud Computing*, vol. 9, no. 1, Dec. 2020, doi: 10.1186/s13677-020-00183-w.

[7] T. Jin and B. Zhang, "Intermediate data fault-tolerant method of cloud computing accounting service platform supporting cost-benefit analysis,"

Journal of Cloud Computing, vol. 12, pp. 1–10, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:255517299

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

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

[10] 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)*, IEEE, 2021, pp. 161–166.

[11] S. R. 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.

[12] S. R. 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.

[13] S. R. M. Zeebaree *et al.*, "Multicomputer multicore system influence on maximum multi-processes execution time," *TEST Engineering & Management*, vol. 83, no. 03, pp. 14921–14931, 2020.

[14] 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)*, IEEE, 2019, pp. 76–81.

[15] 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.

[16] H. Malallah *et al.*, "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.

[17] P. Y. Abdullah, S. R. 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.

[18] D. M. Abdulqader, S. R. M. Zeebaree, R. R. Zebari, S. A. L. I. Saleh, Z. N. Rashid, And M. A. M. 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.

[19] S. R. M. 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.

[20] H. M. Zangana and S. R. M. 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.

[21] Z. S. Ageed and S. R. M. 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.

[22] 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)*, IEEE, 2022, pp. 38–43.

[23] 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)*, IEEE, 2022, pp. 44–48.

[24] J. Antony John Prabu and S. Britto Ramesh Kumar, "Performance of proposed architecture for data transactions in cloud using D1FTBC," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 5390–5395, Jul. 2019, doi: 10.35940/ijrte.B3296.078219.

[25] H. Chen and M. Migliavacca, "StreamDB: A unified data management system for service-based cloud application," in *Proceedings - 2019 IEEE International Conference on Services Computing, SCC 2019* , Institute of Electrical and Electronics Engineers Inc., Sep. 2019, pp. 169–176. doi: 10.1109/SCC.2018.00029.

[26] P. Fan, J. Liu, W. Yin, H. Wang, X. Chen, and H. Sun, "2PC*: a distributed transaction concurrency control protocol of multi-microservice based on cloud computing platform," *Journal of Cloud Computing*, vol. 9, no. 1, Dec. 2020, doi: 10.1186/s13677-020-00183-w.

[27] A. N. Gnana Jeevan and M. A. Maluk Mohamed, "DyTO: Dynamic Task Offloading Strategy for Mobile Cloud Computing Using Surrogate Object Model," *Int J Parallel Program*, vol. 48, no. 3, pp. 399–415, Jun. 2020, doi: 10.1007/s10766-018-0563-0.

[28] B. Balusamy and P. V. Krishna, "Simplified and efficient framework for managing roles in cloud-based transaction processing systems using attribute-based encryption," *Int. J. Computational Science and Engineering*, vol. 14, no. 2, pp. 135–149, 2020.

[29] X. Wang, T. Tawose, F. Yan, and D. Zhao, "Distributed Nonblocking Commit Protocols for Many-Party Cross-Blockchain Transactions," *arXiv* , 2020.

[30] A. Al-Qerem, M. Alauthman, A. Almomani, and B. B. Gupta, "IoT transaction processing through cooperative concurrency control on fog–cloud computing environment," *Soft comput*, vol. 24, no. 8, pp. 5695–5711, Apr. 2020, doi: 10.1007/s00500-019-04220-y.

[31] J. Antony, J. Prabu, S. Britto, and R. Kumar, "An Integrated Architecture For Secured Cloud Data Transactions With Consistency Enhancements," *Annals of R.S.C.B.*, vol. 25, pp. 18328–18339, 2021, [Online]. Available: http://annalsofrscb.ro

[32] M. S. Rajhans, "Enhancing Data Securing In Cloud Using Scalable Transactions," *International Journal of Innovative Research in Computer and Communication Engineering (An ISO*, vol. 3297, no. 11, 2021, [Online]. Available: www.ijircce.com

[33] T. Lykhenko, R. Soares, and L. Rodrigues, "FaaSTCC: Efficient Transactional Causal Consistency for Serverless Computing," in *Proceedings of the 22nd*

International Middleware Conference, 2021, Association for Computing Machinery (ACM), Dec. 2021, pp. 159–171. doi: 10.1145/3464298.3493392.

[34] R. Soares, "An Architecture to Offer Transactional Strong Consistency for FaaS Applications," 2021.

[35] F. Xiang, W. Huaimin, and S. Peichang, "Proof of Previous Transactions (PoPT): An Efficient Approach to Consensus for JCLedger," *IEEE Trans Syst Man Cybern Syst*, vol. 51, no. 4, pp. 2415–2424, Apr. 2021, doi: 10.1109/TSMC.2019.2913007.

[36] D. Dhinakaran, P. M. Joe Prathap, D. Selvaraj, D. Arul Kumar, and B. Murugeshwari, "Mining Privacy-Preserving Association Rules based on Parallel Processing in Cloud Computing," *International Journal of Engineering Trends and Technology*, vol. 70, no. 3, pp. 284–294, Mar. 2022, doi: 10.14445/22315381/IJETT-V70I3P232.

[37] P. Pon and V. Kavitha, "Blockchain based cloud service security architecture with distributed machine learning for smart device traffic record transaction," *Concurr Comput*, vol. 34, no. 3, Feb. 2022, doi: 10.1002/cpe.6583.

[38] M. O. Okoye and H. M. Kim, "Optimized User-Friendly Transaction Time Management in the Blockchain Distributed Energy Market," *IEEE Access*, vol. 10, pp. 34731–34742, 2022, doi: 10.1109/ACCESS.2022.3162214.

[39] Q. Zhang *et al.*, "Efficient Distributed Transaction Processing in Heterogeneous Networks," in *Proceedings of the VLDB Endowment*, VLDB Endowment, 2023, pp. 1372–1385. doi: 10.14778/3583140.3583153.

[40] E. Yumami, I. Irfansyah, M. K. Anam, and H. Hamdani, "Implementation of Cloud Computing Based on Infrastructure as a Service (IaaS) to Improve Transaction Quality (Case Study Shop of Central Mart Pekanbaru)," *JURNAL TEKNOLOGI DAN OPEN SOURCE*, pp. 86–97, Jun. 2023, doi: 10.36378/jtos.v6i1.3127.

[41] A. Al-Qerem *et al.*, "Transactional Services for Concurrent Mobile Agents over Edge/Cloud Computing-Assisted Social Internet of Things," *Journal of Data and Information Quality*, vol. 15, no. 3, Sep. 2023, doi: 10.1145/3603714.

[42] V. S. Kshirsagar and Prof. N. M. Sawant, "Privacy Protection for Cloud Based Online Transaction Using Steganography & Visual Cryptography," *International Journal of Innovations in Engineering and Science*, vol. 8, no. 5, Apr. 2023, doi: 10.46335/ijies.2023.8.5.9.