International Journal of Computer, Network Security and Information System (IJCONSIST)
Vol. 5 No. 2, Maret 2024

A Survey On Causal Consistency Implementation In Geo-Replicated Cases

Muhamad Syukron
Institut Teknologi Sepuluh Nopember, Jawa Timur, Indonesia
muhamad.syukron@its.ac.id

Abdul Aziz
Institut Teknologi Sepuluh Nopember, Jawa Timur, Indonesia
abdl.azis348@gmail.com

Chilyatun Nisa
Institut Teknologi Sepuluh Nopember, Jawa Timur, Indonesia
nchilyatun@gmail.com

Royyana Muslim Ijtihadie
Institut Teknologi Sepuluh Nopember, Jawa Timur, Indonesia
roy@if.its.ac.id

***Abstract*— Distributed storage systems are a fundamental component of large-scale internet services. To meet the increasing needs of users regarding availability and latency, the design of data storage systems has developed into data replication techniques, one of which is geo-replication. Causal consistency is an attractive method for storing geo-replicated data because it is at the crucial point between ease of programming and resulting performance. This method also enables high availability and low latency. However, when implemented into cloud storage, there are limitations regarding throughput and costs. We surveyed several models using methods related to causal consistency in geo-replication cases designed by previous researchers. The models used were derived from papers on causal consistency in geo-replication cases published within the last five years. In this study, we compared the performance of previously designed models based on their performance results. The results of this study are grouping models based on throughput and latency performance obtained.**

Keywords*— Causal Consistency; Cloud Storage; Distributed Systems; Geo-Replication; Latency; Throughput

I. INTRODUCTION

CAUSAL CONSISTENCY (CC) is a consistency protocol positioned between strong and weak consistency. It effectively addresses the challenges of high latency and zero partition tolerance inherent in strong consistency, while avoiding the application-level anomalies often caused by weak consistency [1]. As such, it offers a practical solution that aligns with the demands of distributed environments and adheres to the CAP theorem, which states that a distributed system can only satisfy at most two of the following: consistency, availability, and partition tolerance [2].

Due to its distinctive advantages, CC has been widely adopted in the design of distributed storage systems. These systems typically synchronize data across geographically distributed data centers (DCs) using geo-replication strategies [3]. CC is particularly well-suited to managing fragmented and geo-replicated data in large-scale web applications [20], offering consistency guarantees without incurring the high latency associated with strong consistency models, and thus resolving many of the limitations found in eventual consistency [4]. This makes CC beneficial for both users and developers, as it ensures operation order while still providing consistency guarantees [5].

Geo-replication strategies can be categorized into full and partial replication [6]. In full geo-replication, updates are synchronized across all replicas, which can significantly increase synchronization costs as the number of DCs grows. In contrast, partial geo-replication avoids synchronizing updates to all replicas, thereby reducing synchronization overhead but increasing the cost of managing metadata for remotely stored data items [3].

Multiple CC models are employed in geo-replication contexts, each with distinct characteristics that make them suitable for specific use cases. This paper aims to complement existing discussions of causal consistency in geo-replicated environments, where numerous factors such as network quality and server speed influence both throughput and latency [1]. Motivated by these considerations, this study proposes a classification of CC models as applied to geo-replication scenarios. The classification also distinguishes models based on their operational contexts and use cases.

The grouping process is based on an analysis of whether each surveyed model falls under general causal consistency, causal consistency applied within geo-replication, or geo-replication alone. This categorization is informed by how the models are utilized in the literature. For instance, the CausalSpartan model, though rooted in eventual consistency, is considered part of the CC family, even when not applied in

geo-replicated settings. The final classification also considers the throughput and latency results reported for each model. Throughput varies by model depending on the underlying algorithm, while latency measured in milliseconds per operation also differs and is affected by various factors, including the number of DCs and client load.

Given the diverse range of proposed models tailored to different server requirements, a structured classification becomes essential. The objective of this paper is to identify and categorize existing CC models based on their application domains, as detailed in the survey results.

The structure of this paper is as follows: Section II reviews recent studies on causal consistency and geo-replication; Section III outlines the methodology used in conducting the survey; Section IV presents the findings related to CC models in geo-replicated settings based on performance metrics; Section V provides discussion and analysis; and Section VI concludes the study.

II. Related Work

Causal consistency has been widely adopted in the development of distributed storage systems due to its advantages in achieving low latency and high performance. Today, storage architectures built on this principle have proliferated in various forms, each model distinguished by its unique characteristics. At the conceptual level, causal inference plays a crucial role in evaluating the impact of a cause on its effect, relying heavily on formal representations of the interactions among observed variables, typically expressed through causal graphs. While such graphical representations may be limited in conceptual rigor, they are highly effective in conveying causal relationships [7]. Across the literature, numerous studies have addressed challenges related to causal consistency, with many researchers conducting surveys of proposed models, often guided by their own specific evaluation criteria. Several key works have examined causal consistency and geo-replication in the context of distributed systems. A summary of these survey papers is as follows:

A. A Survey on Causal Discovery: Theory and Practice

This paper emphasizes that causal inference is specifically designed to quantify the fundamental relationships linking causes to their effects. Causal discovery, a subfield within the broader study of causality, reconstructs causal diagrams from data, thereby enabling identification and assessment of causal effects. The study highlights the applicability of causal inference across various domains, including energy production and consumption, the pathophysiology of Alzheimer's disease, unmet treatment needs in schizophrenia, and the relationship between alcohol use and anxiety disorders. However, overlapping and non-identical variable sets, federated learning, and streaming data learning are identified as current limitations in causal discovery [7].

B. Data Replication Schemes in Cloud Computing: A Survey

This survey highlights the lack of a comprehensive, systematic review of data replication in cloud environments, despite its maturity and widespread impact. The paper classifies data replication into categories such as deduplication, auditing, and replication management. It compares key factors such as usage context, replication types, deployment locations, evaluation tools, and the advantages and disadvantages of each method. Several challenges in cloud data replication are also discussed, including replication migration, placement, security, multi-environment support, fault tolerance, and replication schemes [8].

C. An Evaluation of Data Consistency Models in Geo-Replicated Cloud Storage

This paper evaluates various data consistency models used in geo-replicated cloud storage systems. The study employs a performance-oriented methodology to determine whether proposed solutions are effective and feasible. The algorithms assessed generally fail to fully meet all three key CAP properties: consistency, availability, and partition tolerance in geo-replicated data centers. The paper suggests that hybrid approaches, which combine strengths of different algorithms such as Fisheye, HBase, and prefetch mechanisms, may offer viable alternatives in future implementations [9].

D. A Brief Survey on Replica Consistency in Cloud Environments

This work explores the challenges in maintaining consistency in cloud storage systems with geographically distributed data replicas. It emphasizes the need for strategies that ensure all copies are updated consistently upon any data modification. Drawing from a broad body of literature, the authors identify three core consistency approaches: fixed consistency methods, configurable consistency methods, and consistency monitoring techniques. Enhancing these three dimensions is seen as key to addressing the core challenges in this space [10].

E. On Geo-Replication in Distributed Datacenters

This survey explores several themes in distributed data center architectures, offering an overview of general system models, data localization techniques, transaction enhancements, and the hybridization of consistency models. The review points out that no single system has yet succeeded in optimizing multiple dimensions simultaneously, suggesting the feasibility of developing systems that push the boundaries of the consistency–latency tradeoff [11].

Given the diverse range of proposed models tailored to different server requirements, a structured classification becomes essential. The objective of this paper is to identify and categorize existing CC models based on their application domains, as detailed in the survey results.

Previous studies have generally conducted surveys that classify the use of causal consistency in various application domains and describe the models employed for data replication techniques in cloud storage systems. Some of these studies also introduce more specific categorizations. A few of them group the models based on the level of data consistency performance by combining multiple algorithms.

In this survey of causal consistency models within geo-replication scenarios, we introduce a classification scheme that has not been addressed in earlier research. One key contribution of this work is the categorization of causal consistency models based on their throughput and latency outcomes. In addition, we classify data center models into two primary categories, namely those that support causal consistency and those that support geo-replication. We also construct a taxonomy based on different types of data center configurations. This framework is intended to support a more systematic analysis of the performance characteristics of existing data center models.

The resulting classification provides a foundation for identifying suitable models in future proposals. Related studies serve as additional references that illustrate the wide variety of existing model designs. An updated classification of causal consistency models in geo-replication systems is essential for current developments in the field. Each model, shaped by its algorithmic structure, offers distinct advantages and presents certain limitations. This paper also presents a comparative analysis of throughput and latency, which are among the most important performance criteria in distributed systems.

III. METHODOLOGY

In this survey paper, we present a comparative analysis of several models previously proposed by researchers, categorized according to a structured classification framework. Consistency in cloud storage systems is divided into several groups, namely: models implementing state-of-the-art Causal Consistency, those combining Causal Consistency with Geo-Replicated Partial systems, and models utilizing Geo-Replicated Partial mechanisms alone. The motivation for this categorization stems from the fact that partial geo-replication often incurs high latency due to remote access, necessitating functional groupings based on the operational objectives of each model [13].

This classification serves a dual purpose: first, to group models by their functional characteristics, and second, to facilitate comparative evaluation in terms of throughput and latency. These performance metrics are often implicitly defined by the original design goals of each algorithmic model, as observed in prior empirical studies. The allocation of models to specific categories is therefore aligned with their intended applications, enabling a clearer understanding of their suitability for particular use cases. This categorization is visually summarized in Figure 1.

Following this classification, our analysis focuses on throughput and latency as the primary comparative factors. These two metrics serve as critical benchmarks for assessing the efficacy of causal consistency models in the context of distributed cloud storage. In practice, systems must minimize latency while also achieving optimal throughput, making this trade-off central to model evaluation. The taxonomy proposed in this study, which organizes models based on these factors and their deployment characteristics, is illustrated in Figure 2.

This survey specifically concentrates on models implementing causal consistency as a subset of geo-replication frameworks, while also referencing alternative models for comparative purposes within the geo-replication context.

A. Model Classification

This section presents a detailed breakdown of the cloud storage models studied in prior work. The classification not only simplifies comparative analysis across different models but also provides a structured framework for grouping models based on shared characteristics. The focus of this paper is on models implementing causal consistency within geo-replicated environments.

The classification is constructed around three key dimensions: the functional role of the model, the architectural foundation on which it is built, and the context of application for which the model is designed. Based on these dimensions, the following categories are established:

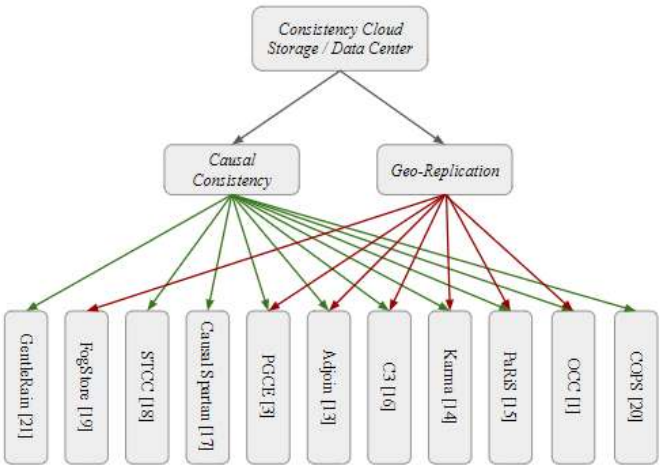

Fig. 1. Variations in Model Modifications

B. Model Categorization

This section presents a structured classification of the cloud storage models analyzed in previous studies. Grouping these models into distinct categories facilitates comparative analysis across different approaches. The studies reviewed in this paper are systematically categorized based on shared characteristics, enabling clearer insights into the distinctions and similarities among them.

The scope of this paper is specifically focused on causal consistency models within geo-replicated systems. The

categorization framework is derived from three primary aspects: the functionality of each model, the architectural principles underlying its design, and the intended usage context. These criteria serve as the foundation for constructing a taxonomy of models, which is used throughout the analysis that follows:

1) *Causal Consistency*
   - CausalSpartan [16]
   - GentleRain [20]
   - Strict Timed Causal Consistency (STCC) [17]
   - Cluster of Order-Preserving Servers (COPS) [19]
2) *Geo-Replication*
   - FogStore [1]
3) *Causal Consistency Partial Geo-Replication*
   - Optimistic Causal Consistency (OCC)5
   - PaRiS [14]
   - Karma [13]
   - C3 [15]
   - Adjoin [12]
   - Partial Geograpical replication and Cloud-Edge (PGCE) [3]

C. Performance-Based Comparison

The comparative results are divided into two key dimensions: throughput and latency. This section evaluates each model based on these performance indicators, as both are critical to ensuring consistency in replicated cloud storage systems. The rationale for selecting these metrics lies in the need to handle high data loads efficiently while maintaining minimal latency across distributed data centers (DCs).

A model is considered to exhibit high throughput when it achieves a performance rate exceeding 10,000 operations per second (op/s). Models with performance below this threshold are classified as having low throughput. Latency, on the other hand, is assessed relative to various influencing factors, but fundamentally reflects response time. Lower latency values indicate superior performance, as they imply faster synchronization and reduced delays in data propagation [3, 12, 13, 14, 15].

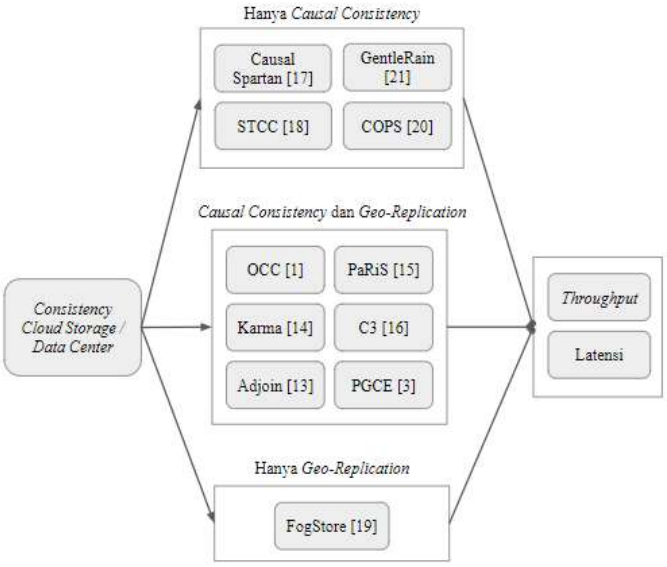

Fig. 2. Simplified Taxonomy Classification

IV. SURVEY RESULT

The experiments examined in the surveyed literature employ a wide range of evaluation metrics to assess the effectiveness of cloud storage models. Among these, throughput and latency, particularly in relation to data load, are the most widely adopted benchmarks and serve as the primary evaluation criteria in this study. Although other parameters such as the number of clients, data centers, or edge nodes are occasionally used to measure performance, these factors are often inconsistent across studies due to differences in model architectures and application scenarios. Consequently, they are not emphasized in our comparative analysis.

The two core performance factors, throughput and latency, serve as the basis for comparing the various models. In addition to the functional differences among the models, these quantitative outcomes are treated as the primary indicators in our evaluation. The comparison between causal consistency models and causal consistency models with partial geo-replication is presented in terms of both model categorization and performance outcomes.

A. Model Categorization

In general, the implementation of causal consistency involves the use of metadata that is appended to each operation. When an operation is received, the associated metadata ensures that its execution adheres to causal consistency constraints. If the conditions are not yet met, the operation is delayed until it is safe to execute according to the consistency rules.

As summarized in Table I, models classified under causal consistency typically evolve from principles of eventual consistency. Causal consistency ensures that if two operations, labeled a and b, are causally related such that a occurs before b, then b must observe the effects of a if the system is functioning

correctly. This behavior can be formally described through the following conditions:

1) Operations a and b are issued by the same client, with a occurring prior to b.

2) Operation a usually performs an update, while b performs a read operation that should reflect the changes made by a.

3) If a third operation, c, is issued afterward, it will maintain the causal relationship established by the previous operations.

Many storage systems also incorporate support for partial geo-replication, where each data center maintains knowledge of the replication state for specific data objects. This allows the system to selectively propagate updates only to the data centers where the objects are replicated. While some systems such as Cassandra natively support this feature, others may achieve it by explicitly restricting replication scopes per object.

The application of geo-replication is typically evident from the deployment structure of data centers in the system. Cloud storage models that support eventual consistency and geo-replication often meet two main criteria. First, they replicate data across widely distributed infrastructures. Second, they place replicas near end users to reduce access latency. Nevertheless, numerous real-world applications require stronger consistency guarantees due to context-aware requirements, where the correctness of data flow depends on preserving consistent system states.

TABLE I. CATEGORIZATION OF CAUSAL CONSISTENCY AND GEO-REPLICATION MODELS

Model	*Causal Consistency Section*	*Geo-Replication Section*
PGCE	Yes	Yes
Adjoin	Yes	Yes
Karma	Yes	Yes
PaRiS	Yes	Yes
C3	Yes	Yes
Causal Spartan	Yes	No
STCC	Yes	No
FogStore	No	Yes
COPS	Yes	No
GentleRain	Yes	No
OCC	Yes	Yes

B. Throughput and Latency

Several algorithms that adopt strong consistency models require that all operations be executed using a serial method. This execution pattern necessitates coordination among replicas to maintain consistency, which often results in high latency and may reduce system availability, potentially leading to operational failures. An important issue arises in time anomaly scenarios, particularly in the amplification of a single query. For example, a social media update may trigger hundreds or even thousands of GET and PUT operations. The delay introduced by each of these operations cumulatively increases the total execution time and ultimately affects the client's response latency. This indicates that updating data efficiently requires a high throughput rate, measured in operations per second. When a data center fails to achieve adequate throughput, requests may accumulate rapidly, leading to system bottlenecks.

Table II presents a comparative summary of the surveyed models, evaluating their throughput and latency performance. The comparison highlights the trade-offs encountered in achieving consistent performance across different architectures.

Strong and linear consistency models demand significant coordination across geographically distributed replicas. As a result, users often experience elevated network latency, which contradicts the design goals of cloud storage systems that aim to offer fast and seamless data access. Conversely, weaker consistency models may compromise data availability and coherence. In such systems, consistency mechanisms are often bypassed unless required to meet specific client demands. This includes use cases such as distributed mailboxes, web page updates, software libraries, or user newsgroups. These challenges raise a fundamental question for system designers: how can high throughput be maintained while ensuring acceptably low latency? While application requirements vary widely, the selection of an appropriate model must ultimately reflect the operational and performance needs of the end users.

V. DISCUSSION

This section presents a synthesis of the findings outlined in the previous sections, with particular attention to how the surveyed models align with the proposed classification framework. The analysis is centered on causal consistency within the context of geo-replication. This focus is chosen because causal consistency represents a more recent advancement in consistency models and serves as a relevant basis for comparison with earlier approaches.

TABLE II. CATEGORIZATION OF CAUSAL CONSISTENCY AND GEO-REPLICATION MODELS

Model	*Throughput*	*Latency*
PGCE	Low	Low
Adjoin	High	High
Karma	High	High
PaRiS	High	High
C3	High	High
Causal Spartan	Low	Low
STCC	Low	Low
FogStore	Low	Low
COPS	High	High

Model	*Throughput*	*Latency*
GentleRain	High	High
OCC	High	High

A. Optimistic Casual Consistency Model

The Optimistic Causal Consistency model, abbreviated as OCC, is designed to prioritize data freshness during updates by focusing on modifications to GET and PUT operations. In this model, servers return the most recent version of a data item available, even when certain causal consistency dependencies have not yet been replicated locally. The system maintains sufficient metadata to detect and block subsequent reads that depend on such unreplicated causal links, until those dependencies are locally fulfilled.

This mechanism enables each data center to perform freshness checks during transmission, ensuring that any data received reflects the most recent version available. OCC is evaluated in contrast to pessimistic consistency models, particularly addressing their limitations regarding data freshness. Empirical results demonstrate that OCC achieves significantly higher throughput, indicating increased data flow within data centers. To accommodate the high volume of operations, experimental evaluations segmented the OCC performance across various partition sizes. The model reached a throughput peak of approximately 120 thousand operations per second, classifying it among high-performance consistency protocols.

B. C3 Model

The C3 model was evaluated using real-world cloud configurations consisting of nine globally distributed data centers. Results indicate that C3 achieves an improved balance between throughput, client-perceived latency, and data freshness when compared to current state-of-the-art models that implement causal-plus consistency in geo-replicated systems. These findings are particularly significant in relation to baseline configurations using the Cassandra system, which typically provides only eventual consistency guarantees.

The C3 framework incorporates modifications to the Cassandra architecture by introducing a partitioned data flow model, tailored to the parameters defined by the system designer. This design is benchmarked against the C-SAT model, or Cassandra Saturn, where load distribution is machine-managed. In contrast, C3 dynamically adjusts Cassandra's internal mechanisms to reduce the burden on the coordination layer. C3 employs specialized algorithms across different architectural layers to enhance scalability and support evolving system demands.

Performance comparisons reveal that the remote execution model of C3 outperforms C-SAT in terms of throughput, while maintaining lower latency. These outcomes emphasize the efficacy of C3's layered design in supporting scalable and low-latency geo-replication.

C. Adjoin Model

The Adjoin model extends the capabilities of CausalSpartan by providing support for true partial geo-replication while maintaining high throughput and update visibility. It employs an adjacency list structure to represent replica storage relationships between data centers. During update propagation, data is transmitted directly to the appropriate data centers, which reduces communication overhead.

Adjoin enhances the vector timestamp technique used in CausalSpartan by incorporating metadata that reflects the most recent stable state of the current data center and its directly connected peers. This method ensures update visibility and reduces metadata storage requirements. Clients report the maximum timestamp of the versions they have read and maintain a set of adjacent dependencies. Servers then validate causal consistency when processing update operations.

In throughput evaluations, the number of partitions was set to eight, and clients continuously issued read and write requests to their local data centers using a round-robin strategy. Data centers periodically synchronized with related peers. As the number of data centers increased, the size of the version vector metadata also grew, leading to higher computational costs per data center and, consequently, a reduction in overall throughput. Despite this, Adjoin achieved a throughput improvement of 7.66 percent over CausalSpartan, highlighting its effectiveness in balancing performance and consistency in geo-replicated environments.

D. Karma Model

The design of the Karma model emerges from the observation that organizing client key-value operations in a ring structure is sufficient to maintain causal consistency in modern systems. Partial geo-replication introduces remote data access patterns that often result in elevated latency. To mitigate this, Karma employs per-data-center caching and a write buffer mechanism. While such techniques are effective in reducing latency, maintaining consistency across multiple layers of caching and buffering requires careful design. Karma guarantees ordering integrity as data flows through the write buffer, the ring-based storage structure, and local caches.

In the Karma architecture, each data center organizes its internal storage areas into a circular grouping, which feeds into an intermediary coordination point, referred to as the Middle Man, before reaching the primary server. Experimental evaluation indicates that Karma achieves an average throughput improvement of approximately 43 percent compared to COPS-PR, with significantly reduced latency at comparable cost. The reduced throughput of COPS-PR is largely attributed to the absence of caching mechanisms. Additionally, COPS suffers from high response time due to its reliance on explicit dependency checking and tracking. While the latency values of Karma and COPS may appear similar under certain workloads, this is due to COPS operating under a higher thread count, which affects latency characteristics under heavy loads.

E. PaRiS Model

The PaRiS model addresses challenges associated with causal consistency tracking protocols through the implementation of Universal Stable Time, or UST. This mechanism identifies a consistent snapshot of the dataset that is installed across all data centers. As a result, transactions in any data center can read from this snapshot without incurring blocking delays. Complementing UST, PaRiS equips clients with private caches to store recent updates that have not yet been included in the UST-determined snapshot. Together, UST and client-side caching enable the realization of transactional causal consistency with non-blocking reads in a partially geo-replicated environment.

PaRiS simplifies the implementation of replication by supporting non-blocking operations, allowing improved system responsiveness. Experimental results demonstrate that, when causal consistency guarantees are enforced, PaRiS delivers approximately 20 percent higher throughput for read-dominant workloads and up to 37 percent improvement for write-intensive scenarios compared to its counterparts without such consistency guarantees.

F. PCGE Model

The PGCE model implements a distributed storage framework built upon a cloud-edge collaborative architecture. This design extends existing causal consistency mechanisms by allowing each edge node to store a selective subset of the global dataset, thereby enabling partial geo-replication. A matrix-based approach is employed to manage replica placement, ensuring that updates are propagated exclusively to nodes storing relevant data. This strategy effectively minimizes metadata processing and reduces synchronization overhead.

By extending cloud storage and computation capabilities to edge nodes located near end-user terminals, PGCE facilitates localized data processing, intelligent decision-making, and efficient resource management. Unlike models such as Adjoin, PGCE does not require metadata processing at every data center. Instead, client requests are handled directly by proximal edge nodes, thereby reducing both communication latency and metadata bandwidth consumption.

Under equivalent experimental conditions, PGCE achieves throughput improvements of 15.39 percent over CausalSpartan, 11.79 percent over STCC, and 6.31 percent over Adjoin. These results confirm its effectiveness in leveraging edge-cloud cooperation to enhance causal consistency performance in distributed systems

VI. Conclusion

This paper has presented a comprehensive examination of causal consistency models within the context of geo-replication, focusing on their performance in terms of throughput and latency. A taxonomy has been proposed to classify existing models based on the architectural scope of their data center configurations, specifically those implementing only causal consistency, those combining causal consistency with geo-replication, and those relying solely on geo-replication. From the body of surveyed literature, a performance comparison was conducted across the identified models. Among the eleven models analyzed, four exhibited relatively low throughput and high latency, while seven demonstrated both high throughput and low latency. These findings suggest a generally proportional relationship between throughput and latency among the surveyed models. The analysis highlights a key challenge for future research: the development of causal consistency protocols that can simultaneously achieve high throughput and minimal latency. Meeting this objective would enhance the practical viability of distributed systems and better support the growing demand for efficient, responsive data center services in contemporary applications.

References

[1] Spirovska, K., Didona, D., & Zwaenepoel, W. (2021). Optimistic causal consistency for geo-replicated key-value stores. *IEEE Transactions on Parallel and Distributed Systems, 32*(3), 527–542. https://doi.org/10.1109/TPDS.2020.3026778

[2] Braun, S., & Deßloch, S. (2020). A classification of replicated data for the design of eventually consistent domain models. *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 33–40. https://doi.org/10.1109/ICSA-C50368.2020.00014

[3] Junfeng, T., Wenqing, B., & Haoyi, J. (2022). PGCE: A distributed storage causal consistency model based on partial geo-replication and cloud-edge collaboration architecture. *Computer Networks, 212*, 109065. https://doi.org/10.1016/j.comnet.2022.109065

[4] Hellerstein, J. M., & Alvaro, P. (2020). Keeping CALM: When distributed consistency is easy. *Communications of the ACM, 63*(9), 72–81. https://doi.org/10.1145/3369736

[5] Mehdi, S. A., Littley, C., Crooks, N., Alvisi, L., Bronson, N., & Lloyd, W. (2017). I can't believe it's not causal! scalable causal consistency with no slowdown cascades. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)* (pp. 453–468). USENIX Association.

[6] Agrawal, D., Abbadi, A. E., & Salem, K. (2015). A taxonomy of partitioned replicated cloud-based database systems. *IEEE Data Engineering Bulletin, 38*(1), 4–9.

[7] Zanga, A., Ozkirimli, E., & Stella, F. (2022). A survey on causal discovery: Theory and practice. *International Journal of Approximate Reasoning, 151*, 101–129. https://doi.org/10.1016/j.ijar.2022.09.004

[8] Shakarami, A., Ghobaei-Arani, M., Shahidinejad, A., Masdari, M., & Shakarami, H. (2021). Data replication schemes in cloud computing: A survey. *Cluster Computing, 24*(3), 2545–2579. https://doi.org/10.1007/s10586-021-03283-7

[9] Mkandla, R., & Chikohora, E. (2021). An evaluation of data consistency models in geo-replicated cloud storage. *2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 1–5. https://doi.org/10.1109/IMITEC52926.2021.9714674

[10] Campêlo, R. A., Casanova, M. A., Guedes, D. O., & Laender, A. H. F. (2020). A brief survey on replica consistency in cloud environments. *Journal of Internet Services and Applications, 11*(1), 1. https://doi.org/10.1186/s13174-020-0122-y

[11] Prehn, L. (2017). Seminar report: On geo-replication in distributed datacenters.

[12] Tian, J., & Pang, Y. (2020). Adjoin: A causal consistency model based on the adjacency list in a distributed system. *Concurrency and Computation: Practice and Experience, 32*(22). https://doi.org/10.1002/cpe.5835

[13] Mahmood, T., et al. (2021). Karma: Cost-effective geo-replicated cloud storage with dynamic enforcement of causal consistency. *IEEE*

Transactions on Cloud Computing, 9(1), 197–211. https://doi.org/10.1109/TCC.2018.2842184

[14] Spirovska, K., et al. (2019). PaRiS: Causally consistent transactions with non-blocking reads and partial replication. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 304–316). IEEE. https://doi.org/10.1109/ICDCS.2019.00038

[15] Fouto, P., et al. (2018). Practical and fast causal consistent partial geo-replication. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)* (pp. 1–10). IEEE. https://doi.org/10.1109/NCA.2018.8548067

[16] Roohitavaf, M., et al. (2017). CausalSpartan: Causal consistency for distributed data stores using hybrid logical clocks. In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)* (pp. 184–193). IEEE. https://doi.org/10.1109/SRDS.2017.27

[17] Gupta, H., & Ramachandran, U. (2018). FogStore: A geo-distributed key-value store guaranteeing low latency for strongly consistent access. In *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems* (pp. 148–159). ACM. https://doi.org/10.1145/3210284.3210297

[18] Roohitavaf, M., & Kulkarni, S. (2018). DKVF: A framework for rapid prototyping and evaluating distributed key-value stores. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (pp. 912–915). ACM. https://doi.org/10.1145/3238147.3240476

[19] Xiang, Z., & Vaidya, N. H. (2020). Global stabilization for causally consistent partial replication. In *Proceedings of the 21st International Conference on Distributed Computing and Networking* (pp. 1–10). ACM. https://doi.org/10.1145/3369740.3369795

[20] Lesani, M., Bell, C. J., & Chlipala, A. (2016). Chapar: Certified causally consistent distributed key-value stores. *ACM SIGPLAN Notices, 51*(1), 357–370. https://doi.org/10.1145/2914770.2837622