This study's object is the consistency of replication data in geo-distributed multicloud systems. The task under consideration is to devise a method for interval ordering of incoming requests by forming a sequence of general order numbers, according to which the order of writing data to geo-distributed replicas is executed.

A feature of the proposed method is the a priori determination of equally long non-intersecting intervals of adjustment of incoming packet numbers, during which the incoming packet numbers are ordered. Grouping users into conditional clusters according to geographical location around brokers makes it possible to determine the priorities of sorting incoming packets. Brokers should be located near the leading replica of each cloud service provider and accept write requests from users from the nearest conditional cluster. In addition, the ordering of incoming packets occurs in the order of their arrival at each broker during the specified intervals. These mechanisms make it possible to synchronize data write operations in one step of global communication between geo-distributed replicas of separate cloud service providers.

To estimate the latency of forming the total ordered sequence of incoming packet numbers, a simulation model has been built, whose feature is the ability to reproduce a different number and geographical location of replicas. The stability of a latency in coordinating incoming write requests in geo-distributed multicloud system with an increase in the intensity of the incoming flow by even 70 times has been shown experimentally.

The results make it possible not only to reduce the latency of writing data to replicas of existing multicloud systems but also to choose the best geographical location of cloud service provider resources when designing new ones

Keywords: multicloud systems, interval consistency method, geo-distributed data replication, latency model

UDC 004.67+004.75
DOI: 10.15587/1729-4061.2025.332189

DEVISING A METHOD FOR DATA CONSISTENCY AT REPLICATION IN MULTICLOUD SYSTEMS

Maksym Volk
Doctor of Technical Sciences, Professor*
Olha Kozina
Corresponding author
PhD, Associate Professor
AFOREHAND Studio
23 Serpnia str., 2, Kharkiv, Ukraine, 61072
E-mail: kozina@aforehand.com.ua
Andrii Buhrii
PhD*
Serhii Osiievskyi
PhD, Associate Professor
Department of Mathematics and Software of ACS**
Mykyta Kozin*
Darya Volk*
Dmytro Diachenko
Brainence – Software Development Vendor
Lemkivska str., 9A, Lviv, Ukraine, 79019
Yurii Turinskyi
Scientific-Organizational Section**
*Department of Electronic Computers
Kharkiv National University of Radio Electronics
Nauky ave., 14, Kharkiv, Ukraine, 61166
**Ivan Kozhedub Kharkiv National Air Force University
Poltavskyi Shliakh str., 123, Kharkiv, Ukraine, 61123

Received 02.04.2025
Received in revised form 14.05.2025
Accepted 06.06.2025
Published 29.08.2025

How to Cite: *Volk, M., Kozina, O., Buhrii, A., Osiievskyi, S., Kozin, M., Volk, D., Diachenko, D., Turinskyi, Y. (2025). Devising a method for data consistency at replication in multicloud systems. Eastern-European Journal of Enterprise Technologies, 4 (2 (136)), 14–22. https://doi.org/10.15587/1729-4061.2025.332189*

1. Introduction

The simultaneous utilization of resources from multiple cloud service (CS) providers is increasingly beneficial for companies across a wide range of industries and with different business needs [1]. Multicloud systems (MCSs) provide the ability to develop highly loaded geo-distributed applications and data warehouses with elastic management while optimizing reliability and maintenance costs [2].

Data replication between geo-distributed centers increases data availability in MCS but is also a major challenge for MCS developers [3]. The data replication mechanism must warrant the required level of consistency for the users of CSs. The need to compromise between performance and consistency has forced researchers to look for different models of data consistency in distributed systems. In general, the consistency model specifies what should be written and read at a given time by different processes or clients when there is simultaneous access to the same resource. All consistency models developed for distributed systems can be conditionally divided into three categories: data-oriented models, client-oriented models, and hybrid models [4].

Consistency models are guaranteed using various methods and protocols, the advantages of which vary depending on the architecture of both the business application for which they are implemented and the architecture and features of the network infrastructure of the cloud system itself. However, determining the resulting final level of consistency in MCS, unlike a single-provider cloud system, is quite difficult both theoretically and practically. This is due to the fact that the CS providers whose resources are involved in MCSs can support different levels of data consistency. In addition, middleware (MW) can be added to the MCS architecture, which usually performs the functions of management, synchronization, placement, and

data consistency between CS providers. At the same time, distributed services of such an MW may have their own data consistency models. Thus, from a theoretical and practical point of view, the consistency model of MCS is heterogeneous [5].

Under such circumstances, replication methods for MCSS should be based on specified write and read latencies, which would provide consistent and up-to-date data within a guaranteed time, and not on a consistency model. The level of latencies inherent in the structure of various replication methods is determined by the number of global communication steps between replicas, especially in MCS. Therefore, an effective data consistency method should take into account the functional features of using the resources of different CS providers, which will be reflected in the minimum number of global communication steps required for replication.

Thus, a relevant task is to devise a method for data consistency when replicating to the resources of different CS providers, the structure of which will contain the lowest level of latencies in replication operations.

2. Literature review and problem statement

In [6], the PigPaxos protocol is proposed as a modification of active consensus replication protocols, i.e., the Paxos family of leader determination algorithms. The random leader rotation procedure in PigPaxos protects relay nodes from turning into hot spots. The leader in PigPaxos broadcasts messages to a set of nodes based on a developed link combining mechanism. This approach results in a throughput improvement of more than 3 times compared to Paxos and EPaxos and an update visibility latency of about 5 ms. However, the authors considered a flow of incoming requests from only 120 users between 25 AWS nodes from a single network partition in a fully switched environment. This means that using PigPaxos in an MCS with heterogeneous network connectivity and not supporting a single network partition can result in significantly higher latency and scalability issues.

The authors of [7] report the results of a comparison of five methods of geographic SMR: MultiPaxos, Mencius, FastPaxos, Domino, and EPaxos, but preliminarily note that the evaluation of all possible combinations of SMR protocols for different applications and replica placement options is a very difficult task. In order to evaluate the write and read latency for each protocol, simulation models were built that take into account different paths between replicas and the communication time between clients and replicas according to RTT data between Microsoft Azure regions. In this case, it is assumed that there is an asynchronous network with FIFO channels between replicas, based on the TCP protocol. The experimental results showed that a larger distance between replicas and different types of paths between replicas lead to a significant difference in latency between protocols: from 150 ms to 300 ms. This indicates the possibility of a significant increase in write latency when using these protocols in geo-distributed MCSs.

In [8], the GeoPaxos+ geographic SMR protocol is described, in which requests for data write are sent to the replica closest to the user, which distributes them to other replicas via FIFO channels. The ordering of requests is performed asynchronously according to unique Lamport timestamps, which are generated and distributed by the serializer replica, which is determined by a special optimization mechanism. The results of the implementation of GeoPaxos+ and EPaxos in three Amazon regions with different proportions of write and read operations showed that the latency of GeoPaxos+ can vary from 90 ms to 500 ms, while in EPaxos it is up to 2 s. However, the use of GeoPaxos+ in MCS will lead to the appearance of additional latency, which is due to the need to implement time synchronization protocols and the FIFO mechanism in data centers (DCs) of different CS providers.

In [9], the authors proposed the Weave method to reduce the data write latency that typically occurs in geo-distributed cloud systems using consensus methods. The Weave architecture is a modification of the Paxos algorithm, which uses passive consensus with multiple leaders and an optimized quorum size to ensure the ordering of guaranteed records. The authors consider the guaranteed record sequence as a reliable geo-replicated log that constantly stores changes to the program state and does not contain the actual results of data write operations, which is only a guarantee to the user regarding the further processing of the request in a specific sequence number of the distributed queue. This approach requires grouping replicas into so-called local availability zones to avoid the global communication stage between them. Experimental results in three AWS regions in Ohio, Frankfurt, and Sydney showed a guaranteed record enqueue time of about 21 ms and a dequeue time of 161 ms. However, these results may be very different in availability zones that will be grouped from replicas of different CS providers, which significantly reduces the advantages of the proposed method for MCS.

The authors of [5] prove that public APIs of MCSs almost do not guarantee the atomicity of operations, which is the basis of the functioning of existing SMR protocols. And consensus between replicas is also not mandatory to guarantee the requirements of data consistency in MCSs. Based on this, the authors of [5] conclude that it is possible to derive general algorithmic protocols that will ensure data consistency and are not necessarily based on consensus methods. As confirmation of these conclusions, the authors propose an algorithm for uploading parts of one file to DCs of different CPs, which does not require a central module between users and cloud servers, unlike methods that use the ZooKeeper protocol [10]. The proposed algorithm is focused only on working with files and does not take into account the geographical distance between DCs, which makes it impossible to use it in geo-distributed MCSs.

In [11], the VIP-Grab protocol was proposed, according to which data replication in MCS is performed based on the group communication primitive, according to which the broker closest to the user broadcasts the formed key-data pair from the write request to other brokers. Data reconciliation according to the VIP-Grab protocol is performed on each broker by forming a common sequence of key numbers during a priori defined intervals. However, in [12] it is shown that the write latency in this case can last longer than two RTT between the most distant DCs.

In [13–16], various replication algorithms and protocols that use time stamps to reconcile geo-distributed data were proposed. The authors of [13] reported a data replication algorithm for based on the global stabilization method. To achieve cause-and-effect consistency, the authors use time stamps of a physical clock, which is loosely synchronized between all servers using the NTP time synchronization protocol. The authors conducted experiments in a simulated distributed system, where servers connected in a ring can exchange packets only with neighboring servers. In this case, a constant of 100 ms was used to simulate network latencies, which does not reflect either the geographical location of replicas or the real random nature of global network latencies even in FIFO channels.

Therefore, according to the results of such experiments, the proposed algorithm cannot be considered optimal in terms of the latency in visibility of remote updates for MCS by clients.

The authors of [16] proposed the K2 method of natural consistency of incoming requests for data updates in one step of global distribution using time stamps of a hybrid logical clock with the PTP synchronization protocol. The paper assumes that usually naturally consistent transactions enter the cloud system, the processing of which must correspond to the order of their receipt in real time, which meets the requirements of strict serialization. At the same time, it was found that the use of time stamps leads to the inversion of real-time connections between transactions and the occurrence of write and read conflicts, to avoid which additional algorithms are proposed. The need for an additional time stamp coordinator server proves that the K2 method is optimized for intra-regional replication in an autonomous server cluster.

Thus, the methods of consistency of incoming write requests that we have reviewed do not take into account the structural features of MCS. In addition, studies in which data reconciliation is based on time stamps have significant limitations in the presence of uneven workload and network load in different regions. Therefore, it is advisable to devise methods for data reconciliation of replication in MCSs based on group distribution mechanisms and algorithmic methods for ordering write operations. This will lead to increased efficiency of functioning of geo-distributed data warehouses and reduced data write latencies in MCSs.

3. The aim and objectives of the study

The aim of our work is to devise a method for coordinating data for recording in the general sequence of incoming packet numbers during replication in multicloud systems. This will reduce the latency in recording data and will contribute to increasing the efficiency of functioning of geo-distributed cloud data storage and multicloud systems.

To achieve this aim, the following objectives were accomplished:

– to develop an algorithm for adjusting incoming packet numbers in the general sequence;

– to propose a structure for a method for interval coordination of data recording requests in geo-distributed MCS replicas.

4. The study materials and methods

The object of our study is the coordination of replication data based on active synchronization in geo-distributed MCSs. In this case, the method for interval ordering of incoming packets and the assessment of latencies in the formation of a common sequence of their numbers, which determine the order of data recording in geo-distributed replicas, is the subject of this study.

The principal hypothesis of the study assumes that the implementation of a new method for coordinating incoming requests by common sequence numbers in the MCS MW could reduce the latency in writing data to geo-distributed replicas. The method is based on an algorithm for ordering incoming requests during equally long non-intersecting number adjustment intervals, in which a special chronology of incoming packet processing is formed. This would ensure increased data relevance and, accordingly, the efficiency of functioning of geo-distributed multicloud web applications and data stores.

The following conditions were used when devising the method:

Condition 1. MCS is considered as geographically distributed DCs of different CS providers, coordination between which is implemented by brokers that are part of MW.

Condition 2. The geographical location of the brokers coincides with the master DC (MDC) containing data replicas in the regions of each CS provider.

Condition 3. Users are grouped into conditional clusters by geographical location and their requests are sent exclusively to the nearest broker, which acts as an active replication node. With such a replication scheme, each broker in MCS can act as both a master and a slave. Data replication between DCs belonging to the same CS provider can be performed by the provider's means in accordance with the replication scheme with one master node.

Condition 4. Despite possible differences in the architecture and principles of data management of different CS providers, in all brokers, synchronization of data write and update operations is implemented by identical mechanisms. In this case, a separate mechanism is used for the read operation.

It is assumed that the MCS implements active replication with several master nodes and CS providers themselves guarantee the failure resistance of their DCs. Therefore, the effectiveness of the MCS architecture under consideration should be evaluated not by the probability of failure of any DC but from the point of view of the reliability of transporting replication packets between brokers through the global network. That is why, our work assumes the use of the TCP protocol for packet transmission between brokers, which makes it possible to obtain the following properties of atomic transmission [13]:

– validity – all slave brokers will eventually receive a packet broadcast by the active broker no later than the maximum allowed latency;

– unified agreement – if one slave broker receives a packet from the active broker, then all other slave brokers will also eventually receive this packet;

– integrity – each slave broker receives a packet from the active broker at most once, and only if it has been broadcast before, but the delivery of packets between brokers in the general order is not guaranteed. For example, if a broker sent packet A before sending packet B, then each slave broker may receive packet A before B or vice versa, packet B before A.

The basis of the ordering of requests in the Latord method is the assumption that their agreement must occur in each broker within equally long non-intersecting adjustment intervals I with the same value in all brokers.

Depending on the area of the conditional cluster of users, the average delivery time of packets from users to the nearest broker can differ significantly in different clusters, which affects the chronology of fixing incoming requests. For example, if user A from the periphery of a large-area cluster sent a request to his broker on the physical clock earlier than user B from a small cluster, which was very close to his broker. In such a situation, there is a high probability that the request from user B will reach MW earlier than the request from user A. This means that the chronology of fixing incoming packets differs from the actual chronology of sending these requests by users. To avoid such situations and to equalize the chances of fixing the actual chronology of generating requests by users in different clusters, within the adjustment intervals I_x, each broker Br_x defines its own availability window W_x. The duration

of such an availability window must be no less than the average delivery time of packets from cluster users to its Br_x. Therefore, the own adjustment interval of each broker Br_x is equal to

$$I_x = W_x + \Delta I_x, \quad (1)$$

where ΔI_x is the residual window, the duration of which must be no less than the average packet transmission time from the active broker to the most distant slave broker of MCS. When coordinating the order of write requests, the MCS uses the single unified value of the adjustment interval

$$I = \max_{x=1..BrMax} I_x. \quad (2)$$

Also, our study assumes that incoming packets receive a sorting priority $Prior_x$, which is determined by the availability window W_x: the longer W_x in Br_x, the higher the sorting priority of packets received from it to other brokers, i.e., if $W_1 > W_2$, then $Prior_1 > Prior_2$ in slave brokers.

It was assumed that the write latency $Lat_{wr}(Br_x(to), j)$ of a data packet $PR(Br_x(to), j)$, with number j during active replication in a geo-distributed MCS is defined as the time difference between the time $t_{wr}(Br_x(to))$ of obtaining permission to write data in the slave broker $Br_x(to)$ and the time $t_u(Br_a(from))$ of the packet from the user to the nearest broker $Br_a(from)$ (Fig. 1)

$$Lat_{wr}\left(Br_x(to), j\right) = t_{wr}\left(Br_x(to)\right) - t_u\left(Br_a(from)\right). \quad (3)$$

Thus, for each write request in MCS with four brokers, there will be four separate values of the packet write latency for each replica. In this case, by the time $Lat_{wr}(Br_x(to), j)$ coincides, the ordering of packet numbers must be completed in all brokers, the duration of which $T_{consist}(Br_x(to), j)$ is defined as the time difference between time $t_n(Br_x(to))$ of the last correction of packet number $PR(Br_x(to), j)$ and the delivery time $t_i(Br_x(to))$ of this packet to $Br_x(to)$

$$T_{consist}\left(Br_x(to), j\right) = t_n\left(Br_x(to)\right) - t_i\left(Br_x(to)\right). \quad (4)$$

For MCSs, where communication means over a global network must be used to synchronize data recording, the packet delivery times between brokers, which are defined as

$$t_{tr}\left(Br_a(from), Br_x(to)\right) = t_i\left(Br_x(to)\right) - t_u\left(Br_a(from)\right), \quad (5)$$

make a significant contribution to the latency in the formation of corrected numbers in the overall sequence of packets $Lat_{consist}(Br_x(to), j)$ and should be considered as independent random processes. Therefore, $Lat_{consist}(Br_x(to), j)$ for each broker is defined as the sum of the duration of the formation of ordered packet numbers and the delivery time of packets from the leading broker

$$\begin{aligned} &Lat_{consist}\left(Br_x(to), j\right) = \\ &= T_{consist}\left(Br_x(to), j\right) + t_{tr}\left(Br_a(from), Br_x(to)\right) = \\ &= t_n\left(Br_x(to)\right) - t_i\left(Br_x(to)\right) + \\ &+ t_i(Br_x(to)) - t_u\left(Br_a(from)\right) = \\ &= t_n\left(Br_x(to)\right) - t_u\left(Br_a(from)\right), \end{aligned} \quad (6)$$

and is considered as a random process, which, in addition to the procedure of interval matching of the general sequence numbers, is determined by probability values t_u, t_i and t_n

$$\begin{aligned} &\forall_{a,x=1..BrMax} Lat_{consist}\left(Br_x(to), j\right) = \\ &= f\left(t_u\left(Br_a(from)\right), t_i\left(Br_x(to)\right), t_n\left(Br_x(to)\right)\right). \end{aligned} \quad (7)$$

Our study suggested that it is necessary to determine a priori the maximum allowable period of latency

$$\begin{aligned} &MaxLate = \\ &= \max_{a,x=1...BrMax} t_{tr}\left(Br_a(from), Br_x(to)\right) = K_{Late} \cdot I, \end{aligned} \quad (8)$$

which must be no less than the maximum average packet delivery time between any two brokers. In this case, the lateness index K_{Late} is calculated as a ceiling from the division

$$K_{Late} = \left\lceil \frac{MaxLate}{I} \right\rceil. \quad (9)$$

Let $MaxLate$ be equal to twice the adjustment interval, hence the lateness index $K_{Late} = 2$.

The time of arrival of write requests from users to each $Br_x(to)$ is considered to be independent random variables with the same function of type $PDF_{U,Br_a(from)}(t_u)$. The parameters of each individual function $PDF_{U,Br_a(from)}(t_u)$ can be determined based on theoretical assumptions, historical data for similar systems, or based on data from the MW module for collecting and analyzing incoming traffic.

Our study assumed that the structure of MCS has a predetermined architecture, that is, the scalability of resources of such a system should change not by changing the composition of CS providers but by adding or balancing the internal resources of the corresponding CS provider with the same location of brokers. This means that the statistical characteristics of probability functions $PDF_{U,Br_a(from)}(t_u)$ of packet delivery times between brokers in pairs should be estimated at the MCS design stage.

Simulation experiments, numerical calculations, and visualization of the results of our work were performed in the MATLAB R2023b computing software environment. Each individual simulation experiment uses one law $PDF_{U,Br_a(from)}(t_u)$ for all brokers and one type of density distribution $PDF_{Br_a(from),Br_x(to)}(t_i)$ between any pair of brokers.

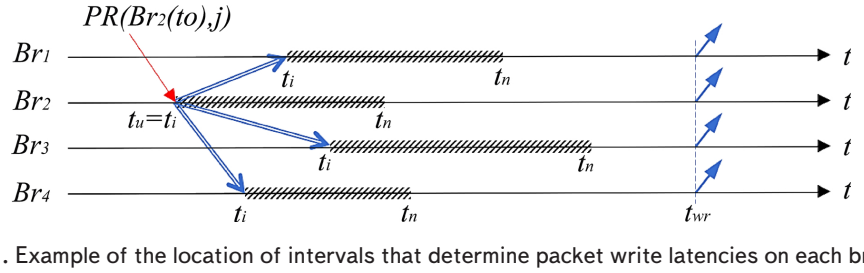

Fig. 1. Example of the location of intervals that determine packet write latencies on each broker

5. Results related to devising the Latord method for data consistency in geo-distributed multicloud systems

5. 1. Algorithm for adjusting the numbers of incoming packets to a general order sequence

For each broker, in each interval $I(to)$, the value of the free number of the general sequence $N_{free}(I(to))$ is determined. When each incoming packet $PR(Br_x(to), j)$ arrives at $Br_x(to)$, a tuple of windows $Br_x(to)$ is formed. The packet numbers from the tuple of these windows M_{late} can be adjusted in connection with the appearance of a new packet. The tuple M_{late} contains all intervals and windows, starting from the source window $Br_x(from)$ of the current incoming packet, and up to $Br_x(to)$. For example, if the incoming packet $PR(Br_3(to), j)$ arrived during interval $I(to)$ from the residual window of broker Br_1 with lateness index $K_{Late} = 2$, i.e., two adjustment intervals before the current window (Fig. 2), then the tuple M_{late} should include all source windows from the interval from $\Delta I_1(from) = \Delta I_1(to - K_{Late}) = \Delta I_1(to - 2)$ and to $\Delta I_1(to)$, packets from which have already arrived at broker Br_3, i.e., $|M_{late}| = 5$ and

$$M_{late} = \left\langle \begin{matrix} \Delta I_1(to-2), W_1(to-1), \\ \Delta I_1(to-1), W_1(to), \Delta I_1(to) \end{matrix} \right\rangle. \quad (10)$$

If no other packets have reached $Br_x(to)$ within $MaxLate$, i.e., $|Mcor| = 0$, or the correct location j_x should be after all existing sequence numbers, then the next number is added to the end of the existing sequence of numbers

$$j_x = N_{free}\left(I(to)\right), \quad (11)$$

where $N_{free}(I(to))$ is the first free number of the general sequence of current $I(to)$ in broker $Br_x(to)$. After that, $N_{free}(I(to))$ is increased by 1. If $|Mcor| \neq 0$ and the correct position of j_x is among $\forall_{i=1...|Mcor|}\exists_{x=1...BrMax} PR(Br_x(to), j_x)$, then the number is inserted into the existing general sequence so as not to disturb the previous numbers of packets from $Mcor$, and the numbers of all subsequent packets are increased by 1.

When determining the correct number of an incoming packet in the existing ordered sequence of packets from the $Mcor$ array, it is necessary to maintain the correct order of the windows from which the packets arrived. This means that when sorting packet numbers with $Mcor$, packets arriving from the previous window, for example, $\Delta I_x(from) = \Delta I_x(to - K_{Late})$ should precede packets arriving from windows $W_x(from) = W_x(to - (K_{Late} - 1))$ and $\Delta I_x(from) = \Delta I_x(to - (K_{Late} - 1))$. In addition, packets arriving during the same window from brokers with a higher priority should have lower numbers than packets arriving from brokers with a lower priority. The Corr algorithm for determining the correct incoming packet numbers among the packets of tuple $Mcor$ is shown in Fig. 3.

For packets that exited from the same window, for example, $W_1(from)$, ordering in $Br_x(to)$ should be performed according to the order of their exit from the source window, i.e., the packet that exited first from $W_1(from)$ will have a number lower than the packet that exited second, regardless of the order in which they arrived at $Br_x(to)$ (Fig. 4).

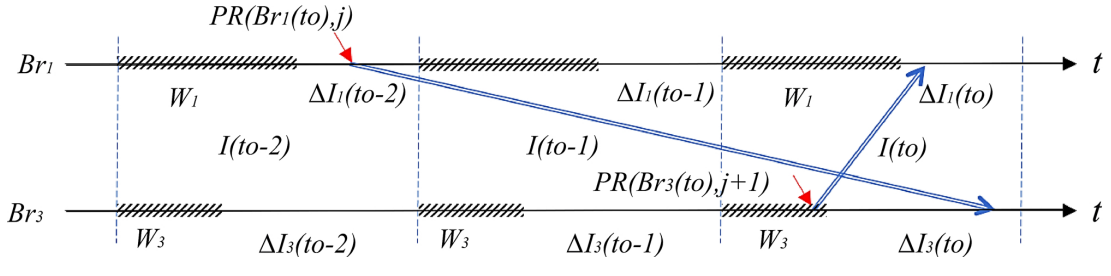

Fig. 2. Example of forming a tuple of windows for adjusting packet numbers $PR(Br_1(to), j)$ and $PR(Br_3(to), j+1)$

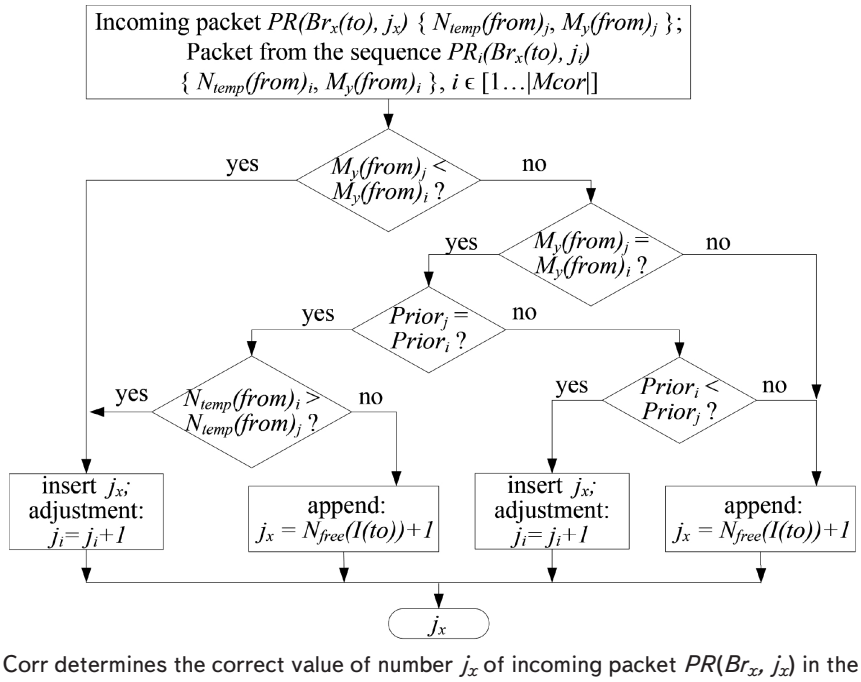

Fig. 3. Algorithm Corr determines the correct value of number j_x of incoming packet $PR(Br_x, j_x)$ in the general sequence

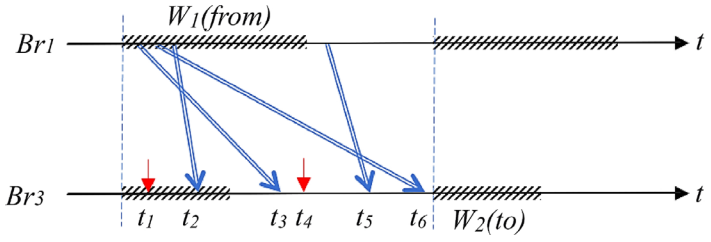

Fig. 4. Example of determining the correct location of packet $PR(Br_2, t_6)$ in the overall sequence

As an example, in Fig. 4, it is assumed that $Prior_1 > Prior_2$, therefore the correct sequence of incoming packets to Br_2, in terms of the moments of their arrival, takes the form $\langle t_3, t_6, t_2, t_1, t_5, t_4 \rangle$.

Thus, the chronology of packet arrival to slave brokers is determined not by absolute time but by the order $N_{temp}(to)$ of packet arrival within each window $\forall_{y=1...|Mlate|} M_y(to)$, and to the master brokers $Br_x(from)$ – by the order $N_{temp}(from)$ of packet arrival from users during the source window $M_y(from)$, $y = 1...|M_{late}|$.

5. 2. Structure of the Latord method for interval data consistency in geo-distributed replicas of multicloud systems

The replication mechanism in MCS determines that the master broker, having received a packet with data for writing from a user of its cluster, must immediately send the packet to all other brokers. The packet received from another broker becomes the input packet $PR(Br_x(to), j_x)$ for current broker Br_x. All input packets receive numbers on their brokers and are idempotently reconciled in a general order sequence according to the Latord method.

The input data for the Latord method are:

– the maximum number of brokers in the system *BrMax*, which is equal to the number of user clusters;

– the average packet delivery time from users to the broker of their cluster and the matrix of the average packet delivery time between brokers $t_{tr}(Br_a(from), Br_x(to))$;

– the value of the starting free number of the general sequence $N_{free}(I_0)$ for all brokers.

Based on the a priori determined input data, the availability windows W_x and the residual windows ΔI_x are determined, which are used to calculate the own adjustment intervals I_x for each broker according to expression (1) and the unified value of the adjustment interval I according to expression (2). According to availability windows W_x, the sorting priority $Prior_x$ of packets from each broker is determined, and the lateness index K_{Late} is determined according to expression (9).

After preparing these values, an array of packets *Mcor* is formed that arrived from the tuple of windows M_{late}, which is formed according to the example in expression (10). The incoming packet $PR(Br_x(to), j)$ is assigned number j_x in the overall packet sequence, and other packet numbers in the *Mcor* array are adjusted as needed according to the algorithm shown in Fig. 5.

The moment $t_{wr}(Br_x(to))$ of obtaining permission to write data of packet $PR(Br_x, j_x)$ to all replicas according to the Latord algorithm occurs after the expiration of the maximum allowed latency time M_{late}, during which the ordering of packet numbers occurred. For example, a packet that arrived at the broker during window $W_x(to)$ will receive permission to write at the end of the availability window $W_x(to + K_{Late})$. In this case, the procedure for writing the data packet $PR(Br_x, j_x)$ can only occur if this packet has already received permission to write and a packet with the previous number has already been written to the current replica.

To assess the effectiveness of the proposed method, a series of three experiments was conducted on the constructed simulation model, which considers MCS of four CS providers, therefore $BrMax = 4$. In each experiment, each broker received three separate streams of 50,000 incoming packets from users with random functions $PDF_{U,Br_a(from)}(t_u)$ of the distribution of intervals between incoming packets according to the Pareto law, truncated exponential, and uniform laws. The average values for all three types of functions $PDF_{U,Br_a(from)}(t_u)$ for each broker during each experiment were the same (Table 1).

Fig. 6 shows the results of simulation experiments on calculating the cumulative probability distribution function $CDF(Lat)$ of the latency in forming the correct positions of packet numbers in the general sequence for each broker separately.

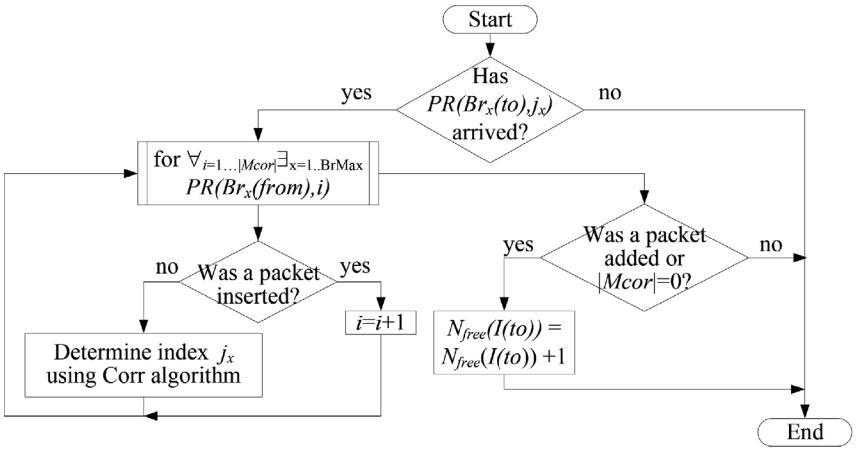

Fig. 5. Algorithm for interval ordering of incoming packet numbers in the *Mcor* array

Table 1

Average values of intervals between incoming packets, ms

Experiment No.	to Br_1	to Br_2	to Br_3	to Br_4
1	148	97	163	112
2	37	24	41	28
3	2	2	2	2

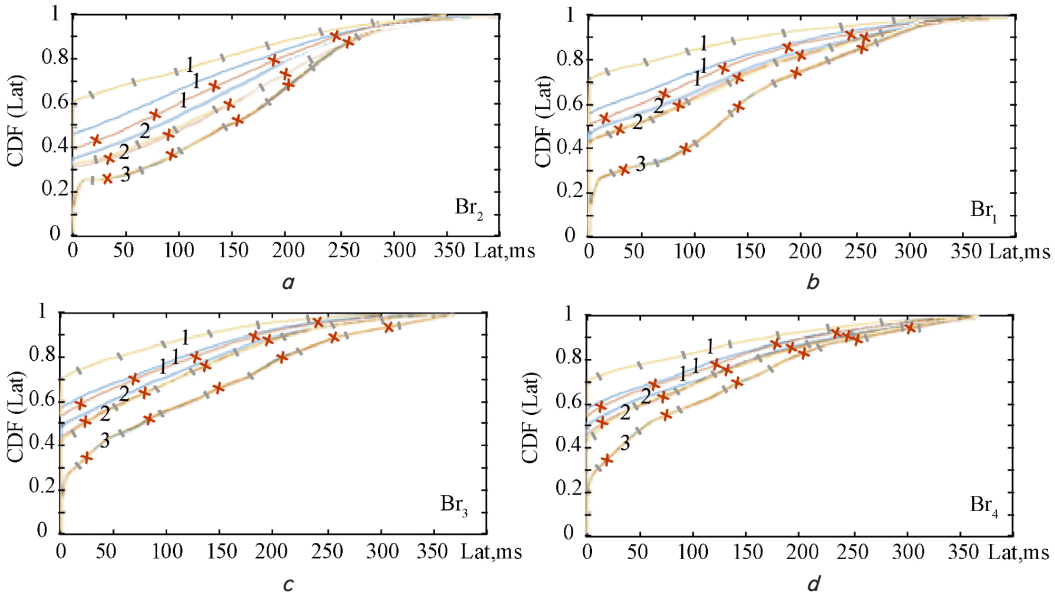

Fig. 6. Cumulative probability distribution functions *CDF*(*Lat*) of the latency in forming correct numbers of the general sequence for experiments 1–3: *a* – at Br_1, *b* – at Br_2, *c* – at Br_3, *d* – at Br_4; — Pareto law; —1— – experiment 1; — uniform law; —2— – experiment 2; — truncated exponential law; —3— – experiment 3

In each experiment, the packet delivery time between brokers was generated according to a uniform law with the following mean values *MeanBrToBr* (ms)

$$MeanBrToBr = \begin{array}{c|cccc} & Br_1 & Br_2 & Br_3 & Br_4 \\ Br_1 & 0 & 156 & 82 & 59 \\ Br_2 & 156 & 0 & 130 & 107 \\ Br_3 & 82 & 130 & 0 & 118 \\ Br_4 & 59 & 107 & 118 & 0 \end{array} \quad (12)$$

with SD values equal to 8 ms for all brokers.

For all simulation experiments, the duration of adjustment interval $I = 295$ ms, and the vector of availability windows for brokers Br_1–Br_4, respectively, $W_x = |90\ 76\ 30\ 19|$ (ms), $x = [1...4]$.

6. Discussion of results based on the devised Latord method for data consistency at replication in multicloud systems

The interval approach to ordering incoming packets according to the Corr algorithm, which is illustrated in Fig. 2, 4 and given by expressions (1) and (2), allows us to solve the well-known problem of sequences of general order [17]. The essence of this problem is that it is impossible to sort a set formed from an infinite stream of elements, that is, a stream of packets.

The Latord method, unlike data replication methods that use physical clocks for synchronization [8], uses a monotonically increasing sequence of numbers, as shown in Fig. 2, 4. Packet numbers for a priori defined interval M_{late} are idempotently ordered on all brokers according to a certain chronology, which in content is analogous to a tuple of logical clock labels and provides a model of consistency in the end. In the VIP-Grab protocol, there is no maximum allowed latency for incoming packets. This leads to a latency in forming correct numbers in about 4% of incoming packets, which is 10 times longer than the round trip time of packet transmission between MCS brokers [11]. The presence of the M_{late} interval makes it impossible for such a situation to occur when using the Latord method.

A feature of the devised Latord method is the need for only one global broadcast step, as shown in Fig. 1. This is less than in consensus replication methods [6, 7, 9], in which two or more global broadcast messages must be exchanged between geo-distributed DCs to implement a single order of writing packets to all replicas. The proposed algorithm for ordering incoming requests and the principle of locating MW brokers relative to users and geo-distributed replicas of different CS providers reduces the latencies that are determined from formulas (3) and (6). Therefore, using the Latord method will make it possible not only to reduce the latency of writing data to replicas of existing MCSs but also select the best geographical location of CS provider resources when designing new MCSs.

The simulation model built allows us to estimate latency $Lat_{consist}(Br_x(to), j)$, which is determined from formula (6) and is equivalent in essence to the visibility latency in work [6]. The experimental results showed that for the maximum allowed packet latency $MaxLate = 2 \cdot I = 2 \cdot 295 = 590$ ms, the latencies in forming correct general sequence numbers on all four brokers do not exceed 400 ms (Fig. 6). Our result is attributed to the fact that the number adjustment occurs between packets that have arrived at the brokers during the double adjustment interval *I*. The value of *I* depends on maximum value *MeanBrToBr*, which is equal to 156 ms and reflects only one step of global communication between geographically distributed brokers, and on the maximum availability window, which is equal to 90 ms. A comparison of the results of experiments 1–3 in Fig. 6 shows that the maximum latency in matching incoming packets does not change when the intensity of incoming requests increases even 70 times (Table 1). This demonstrates the stability of the Latord method to peak input loads.

The results of our simulation experiments showed that increasing the intensity of incoming packets leads to such an increase in the number of packets in each adjustment interval *I* that the difference between the forms of laws $PDF_{Br_a(from),Br_x(to)}(t_i)$ disappears. This is seen in Fig. 6, where in experiment 3, with different $PDF_{Br_a(from),Br_x(to)}(t_i)$ forms,

the *CDF*(*Lat*) curves merge into almost one on all brokers. It is known that for the flow of incoming packets, intensity fluctuations are natural [18] in accordance with the jumps in the popularity of services or the schedule of working hours. Therefore, the application of our results is very useful in the field of designing feedback modules in MCS MW for dynamic configuration of the replication latency depending on the intensity of the flow of incoming requests.

It should be noted that the choice of the value of adjustment interval $I = 295$ ms can be considered a limitation for the application of our results. An additional limitation of this study is determining the latency $K_{Late} = 2$.

The disadvantage of our study is the complexity of a reasonable a priori determination of intervals I_x and W_x for MCS, which are defined at the design stage. Incorrect selection of these intervals may become a place that makes it impossible to use our method in practice.

This study in the future should look at analyzing the influence of values of the adjustment interval, the latency period, the availability windows, as well as relationships between them, on acceptable levels of data recording latency in geographically distributed replicas. The formation of criteria for selecting these intervals could create conditions for the practical application of our method when designing the MCS structure and will be considered in subsequent studies.

7. Conclusions

1. An algorithm for adjusting incoming packet numbers into a general order sequence has been developed. Grouping users according to their geographical location around DCs included in the MCS architecture into conditional clusters makes it possible to determine the priorities of source brokers according to the average packet delivery time from cluster users to the nearest broker. Using such broker priorities helps reflect the chronology of user request generation on each broker. In addition to priorities, incoming packet numbers are ordered according to the order of their arrival at each broker during specified intervals. This algorithm allowed us to develop an algorithm for interval ordering of incoming packets in geo-distributed replicas of multicloud systems.

2. The structure of the Latord method for interval coordination of data write requests in geo-distributed MCS replicas has been proposed. The main difference of this structure from the structures of consensus methods is the use of equally long non-intersecting adjustment intervals with a single value for all brokers. During these intervals, data write requests are coordinated on each broker using the algorithm for adjusting the incoming packet numbers into a sequence of general order without using a physical clock. This has made it possible to perform replication data coordination in one step of global communication between geo-distributed MCS DCs, while the structure of the Latord method takes into account the geographical location of resources of different CS providers. The results of our study proved the dependence of latency in the formation of correct packet numbers on the forms of distributions of incoming streams of data write requests with low intensity. The stability of latency in the coordination of incoming write requests in geo-distributed MCSs has also been proven even with an increase in the intensity of the incoming stream by as much as 70 times.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available, either in numerical or graphical form, in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Mamchych, O., Volk, M. (2022). Smartphone Based Computing Cloud and Energy Efficiency. 2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT). https://doi.org/10.1109/dessert58054.2022.10018740
2. Mamchych, O., Volk, M. (2024). A unified model and method for forecasting energy consumption in distributed computing systems based on stationary and mobile devices. Radioelectronic and Computer Systems, 2024 (2), 120–135. https://doi.org/10.32620/reks.2024.2.10
3. Tricomi, G., Merlino, G., Panarello, A., Puliafito, A. (2020). Optimal Selection Techniques for Cloud Service Providers. IEEE Access, 8, 203591–203618. https://doi.org/10.1109/access.2020.3035816
4. Aldin, H. N. S., Deldari, H., Moattar, M. H., Ghods, M. R. (2019). Consistency models in distributed systems: A survey on definitions, disciplines, challenges and applications. arXiv. https://arxiv.org/abs/1902.03305
5. Mhaisen, N., Malluhi, Q. M. (2020). Data Consistency in Multi-Cloud Storage Systems With Passive Servers and Non-Communicating Clients. IEEE Access, 8, 164977–164986. https://doi.org/10.1109/access.2020.3022463
6. Charapko, A., Ailijiang, A., Demirbas, M. (2021). PigPaxos: Devouring the Communication Bottlenecks in Distributed Consensus. Proceedings of the 2021 International Conference on Management of Data, 235–247. https://doi.org/10.1145/3448016.3452834
7. Shiozaki, K., Nakamura, J. (2024). Selection Guidelines for Geographical SMR Protocols: A Communication Pattern-Based Latency Modeling Approach. Stabilization, Safety, and Security of Distributed Systems, 344–359. https://doi.org/10.1007/978-3-031-74498-3_25

8. Coelho, P., Pedone, F. (2021). GeoPaxos+: Practical Geographical State Machine Replication. 2021 40th International Symposium on Reliable Distributed Systems (SRDS), 233–243. https://doi.org/10.1109/srds53918.2021.00031
9. Eischer, M., Straßner, B., Distler, T. (2020). Low-latency geo-replicated state machines with guaranteed writes. Proceedings of the 7th Workshop on Principles and Practice of Consistency for Distributed Data, 1–9. https://doi.org/10.1145/3380787.3393686
10. Petrescu, M. (2023). Replication in Raft vs Apache Zookeeper. Soft Computing Applications, 426–435. https://doi.org/10.1007/978-3-031-23636-5_32
11. Kozina, O. A., Panchenko, V. I., Kolomiitsev, O. V., Usik, V. V., Stratiienko, N. K., Safoshkina, L. V., Kucherenko, Y. F. (2024). Data consistency protocol for multicloud systems. International Journal of Cloud Computing, 13 (1), 42–61. https://doi.org/10.1504/ijcc.2024.136284
12. Kozina, O., Kozin, M. (2022). Simulation Model of Data Consistency Protocol for Multicloud Systems. 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek), 1–4. https://doi.org/10.1109/khpiweek57572.2022.9916343
13. Xiang, Z., Vaidya, N. H. (2020). Global Stabilization for Causally Consistent Partial Replication. Proceedings of the 21st International Conference on Distributed Computing and Networking, 1–10. https://doi.org/10.1145/3369740.3369795
14. Kakwani, D., Nasre, R. (2020). Orion. Proceedings of the 7th Workshop on Principles and Practice of Consistency for Distributed Data, 1–6. https://doi.org/10.1145/3380787.3393676
15. Song, H., Wang, Y., Chen, X., Feng, H., Feng, Y., Fang, X. et al. (2025). K2: On Optimizing Distributed Transactions in a Multi-region Data Store with TrueTime Clocks (Extended Version). arXiv. https://doi.org/10.48550/arXiv.2504.01460
16. Lu, H., Mu, S., Sen, S., Lloyd, W. (2023). NCC: Natural Concurrency Control for Strictly Serializable Datastores by Avoiding the Timestamp-Inversion Pitfall. arXiv. https://doi.org/10.48550/arXiv.2305.14270
17. Kleppmann, M. (2017). Designing Data-Intensive Applications. O'Reilly Media, 614.
18. Gracia-Tinedo, R., Junqueira, F., Zhou, B., Xiong, Y., Liu, L. (2023). Practical Storage-Compute Elasticity for Stream Data Processing. Proceedings of the 24th International Middleware Conference: Industrial Track, 1–7. https://doi.org/10.1145/3626562.3626828