ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

УДК 004.75 **DOI: 10.20998/2411-0558.2025.01.06**

М. О. ВОЛК, д-р техн. наук., проф., ХНУРЕ, Харків,
О. А. КОЗІНА, канд. тех. наук, AFOREHAND Studio, Харків,
М. Д. КОЗІН, магістр, ХНУРЕ, Харків

МЕТОД СИНХРОНІЗАЦІЇ ЗАПИТІВ НА ЗАПИС ДАНИХ У ФЕДЕРАТИВНИХ ХМАРНИХ СИСТЕМАХ

У статті запропоновано метод синхронізації даних у гео-розподілених федеративних хмарних системах (ФХС), які не використовують засоби провайдерів хмарних послуг для керування спільними даними. У розробленому методі синхронізація запитів на запис даних від користувачів, на відміну від моделей, що використовують консенсуси реплік, забезпечується шляхом формування номерів вхідних пакетів загального порядку протягом фіксованого апріорно визначеного терміну, що є однаковим для усіх реплік. У роботі удосконалена процедура загального упорядкування вхідних пакетів з протоколу VIP-Grab, використання якої для синхронізації дозволить вирішити потенційні конфлікти порядків, запобігти багаторазовому запису вхідних даних у репліках та знизити латентність запису ФХС. Іл.: 4. Бібліогр.: 20 назв.

Ключові слова: федеративні хмарні системи; синхронізація операцій; реплікація.

Постановка задачі. Мультихмарні системи (МХС) надають можливість ІТ-підприємствам відповідно до потреб їхнього бізнесу використовувати ресурси та послуги декількох провайдерів хмарних послуг (англ. *Cloud Service Providers*, CSP). Згідно зі звітом Flexera 2024 State of the Cloud Report, 89% ІТ-підприємств використовують мультихмарні технології, а 73% обирають гібридний хмарний підхід, який поєднує публічні та приватні хмари [1]. Такі системи забезпечують гео-розподілені додатки та сховища даних клієнтів додатковою гнучкістю, надійністю та підвищеною керованістю при оптимізації рівня резервів та операційних витрат на обслуговування, але створюють проблеми, пов’язані з розподілом ресурсів, оптимізацією продуктивності та несуперечливістю даних [2].

Федеративні хмарні системи (ФХС), як підклас МХС, надають змогу ІТ- підприємствам керувати спільними даними, що розташовані у різних CSP за відсутності чіткої угоди між CSP. Координація між ресурсами

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

залучених CSP у ФХС виконується не засобами окремих CSP, а повністю реалізується на клієнтській стороні розподілених додатків чи сервісів [3] у проміжному програмному забезпеченні (ППЗ). У обов'язки ППЗ ФХС зазвичай входять також функції з розміщення, керування, синхронізації та узгодженості даних.

Синхронізація даних у ФХС має забезпечувати несуперечливість та актуальність даних, що зберігаються у дата-центрах (ДЦ) різних CSP. При цьому реалізована у ФХС модель узгодженості даних має гарантувати, що у випадку, коли кілька користувачів намагатимуться змінити один й то й же запис одночасно, жодна інформація ні в якому CSP не буде втрачена. Отже, треба розробити такий метод синхронізації даних, який, з одного боку, задовольнить усім специфічним вимогам бізнес-логіки хмарного додатку чи сервісу, а з іншого, буде відповідати функціонально-структурним особливостям гео-розподілених ФХС.

Аналіз літератури. Потреба отримувати актуальні узгоджені дані у найшвидший термін з будь-якої репліки у гео-розподілених ФХС породжує декілька науково-технічних задач, які на сьогодні не мають однозначного рішення. Існуючі методи реплікації та методи синхронізації даних у хмарних системах вирішують пов'язані задачі, але фокусуються на їх різних аспектах.

У роботах [4 – 6] автори розглядають особливості та можливості застосування численних моделей узгодженості даних у розподілених та хмарних системах. У зв'язку з тим, що кожен CSP має власну модель узгодженості даних, для кожній ФХС треба визначати загальну модель узгодженості даних, яка може бути неоднорідною [5]. Тим не менш, кожна надійна ФХС повинна реалізовувати такий метод синхронізації даних, який забезпечить узгодженість даних, незважаючи на різні гарантії узгодженості у кожному окремому CSP.

Для вирішення питань стосовно того у який спосіб, де та коли створювати чи підтримувати у актуальному стані копії баз даних чи їх частини, тобто, стосовно реплікації даних у хмарних системах широко використовуються протоколи реплікації кінцевого автомату [7], наприклад, Raft [8], Zab [9], та численні варіанти протоколу Paxos: MultiPaxos [10], NOPaxos [11], Mencius [12], FastPaxos [13], Domino [14] і EPaxos [15]. Протоколи реплікації кінцевого автомату базуються на примітиві

атомарної трансляції та визначають за кворумом лідера-секвенсора реплікації, до якого кожна репліка має надсилати усі запити на обробку даних від кінцевих користувачів. Фундаментальними відмінностями у протоколах реплікації кінцевого автомату є різні кількості реплік, що створюють кворум та різні латентності відгуку, тобто, час від моменту, коли кінцевий користувач надіслав запит до репліки, до отримання відповіді. Так, наприклад, у базовому варіанті протоколу Paxos латентність відгуку складається з не менш ніж двох RTT (англ. *round-trip time*), тому, що складається з часу кругової мережевої доставки пакетів від користувача до лідера та від лідера до інших реплік, а у протоколі NOPaxos – до одного RTT.

У георепліkованих системах консенсусні протоколи реплікації значно збільшують затримку відгуку, тому для реплікації даних у таких системах було запропоновано декілька спеціалізованих протоколів, наприклад, Weave [16] та CURP [17], де використовується групування реплік і оптимізований розмір кворуму для забезпечення гарантованого запису. Однак, характеристики ФХС представляють собою специфічну архітектуру для зберігання та керування даними, яка порушує багато припущень навіть спеціалізованих консенсусних методів. Зокрема, ФХС не можуть ефективно обмінюватися пакетами для того, щоб координувати між собою виконання протоколу консенсусу, як це можливо між серверами у приватній хмарі чи у одному дата-центрі одного CSP. Крім того, теоретичні припущення, що лежать в основі систем кворуму, ігнорують багато практичних міркувань, таких як неоднорідність обчислювальних ресурсів в різних CSP, нерівномірність робочого навантаження та навантаження на мережу у різних регіонах, які сумісно унеможливлюють виконання атомарної трансляції, що необхідна для коректного функціонування протоколів реплікації кінцевого автомата [18]. Тому, методи синхронізації запитів, що базуються на механізмах групової розсилки та загальної впорядкованої послідовності операцій є дуже перспективними для ФХС.

У роботі [19] було запропоновано протокол VIP-Grab, за яким синхронізація даних у МХС виконується на основі примітиву групового зв’язку без голосування реплік таким чином, що кожна репліка може приймати запити на запис даних та у свою чергу поширювати оновлені дані на інші репліки. Для вирішення розбіжностей у репліках у VIP-Grab

запропоновано локально формувати загальну послідовність номерів ключів, які додаються до вхідних даних. Алгоритм впорядкування таких монотонно зростаючих номерів гарантовано сходиться до ідентичних значень, але, як показано у роботі [20], в деяких ситуаціях може тривати довше, ніж два RTT. У методі синхронізації реплік на основі протоколу VIP-Grab відсутні недоліки консенсусних механізмів синхронізації, що проявляються у гео-розподілених ФХС, але наявні місця, удосконалення яких гарантовано знизить латентність відгуку хмарного додатку чи сервісу.

Мета статті. Розробка методу синхронізації запитів на запис даних без голосування реплік у гео-розподілених ФХС шляхом удосконалення процедури формування повністю впорядкованої послідовності номерів вхідних пакетів протоколу VIP-Grab.

Функціональні особливості ППЗ ФХС для обробки запитів на запис даних. Будемо розглядати ФХС як гео-розподілене середовище, що використовує ресурси ДЦ не менш ніж чотирьох різних CSP. Такі ресурси можуть мати різні внутрішні архітектури, механізми внутрішньої реплікації баз даних чи підтримки несуперечності та цілісності даних, різні гарантії угоди про рівні обслуговування SLA (англ. *Service-Level-Agreement*), тощо, але при цьому сервіс, що надається такою ФХС для кінцевих користувачів, потребує надійного та швидкого доступу до даних. Будемо розглядати ФХС, де для досягнення цих цілей не використовуються засоби та сервіси самих CSP, натомість у кожному CSP використовується ідентичний модуль ППЗ, так званий брокер $Br_x, x \in [1, \text{Max}]$, який, окрім функцій з координації ресурсів різних CSP, ініціює операції CRUD (англ. *Create, Read, Update, Delete* – створення, читання, оновлення і видалення) для даних у відповідному ДЦ. Також будемо вважати, що у кожній ДЦ є репліка, і у ППЗ кожного брокеру реалізовані ідентичні механізми для кожної окремої операції CRUD, тобто, наприклад, результати читання даних з реплік двох будь-яких брокерів повернуть однакові актуальні дані не змінюючи їх при цьому.

Для зниження латентності операції запису даних припустимо, що запити від кінцевих користувачів, що умовно згруповані у кластери, потрапляють до найближчого брокеру (рис.1).

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

Рис. 1. Схема взаємодії ППЗ брокерів $Br_1 - Br_4$ ФХС при обробці запитів на запис даних

При цьому трансляція пакетів між брокерами має відповідати наступним вимогам:

– якщо з брокера-джерела транслюється пакет до інших брокерів, то всі брокери зрештою отримають цей пакет, не пізніше максимально дозволеного терміну запізнення;

– якщо один брокер отримує пакет від іншого брокеру, то всі інші брокери також зрештою отримають цей пакет;

– кожен брокер отримує пакет від іншого брокеру щонайбільше один раз, і тільки якщо він раніше транслювався.

– причинний порядок пакетів при отриманні відсутній, тобто, якщо брокер-джерело надіслав пакет А перед тим, як надіслати пакет Б, то кожен інший брокер може отримати пакети А і Б у довільному порядку: А раніше Б чи Б раніше А.

Отже, будемо розглядати ППЗ ФХС з активною асинхронною реплікацією даних з багатьма лідерами на основі загальної впорядкованої послідовності пакетів на запис даних.

Принцип обробки запитів на запис даних у ППЗ ФХС на основі загальної послідовності номерів пакетів. Особливістю синхронізації пакетів за методом активної реплікації незалежно від методу впорядкування вхідних даних є обов'язкова передача пакетів з даними для синхронізації між ДЦ, тому при обробці запитів на оновлення чи на запис нових даних у ФХС пропонується розглядати окремо методи обміну метаданими, що необхідні для синхронізації, та методи обміну корисними даними для запису в бази даних.

Процедура синхронізації вхідних пакетів з даними для оновлення чи для запису, що виконується на основі монотонно зростаючих номерів у загальній послідовності операцій, що запропонована у методі VIP-Grab, базується на використанні для кожного вхідного пакету ключа-ідентифікатора *key*. Такий ключ формується у брокері, який отримав пакет від користувача. У методі VIP-Grab до такого ключа-ідентифікатора додається відкоригований номер пакету N_{gl} із загальної послідовності та пара $\left(key, N_{gl}\right)$ використовується для визначення часу, коли дані можуть бути записані у репліки ФХС. Таким чином, пакет від користувача породжує два паралельних механізми: упорядкування пакетів у брокерах та передачі корисних даних від одного брокеру до усіх інших. Після виконання цих механізмів у ППЗ виконується процедура отримання дозволу на запис корисних даних у кожному брокері, відповідно до результатів виконання обох попередніх механізмів.

Процедура отримання дозволу на запис даних у методі VIP-Grab вимагає відправляти з кожного брокеру усім іншим брокерам повідомлення-підтвердження про формування коректного номеру пакету, отже, запис корисних даних можливий тільки після отримання кожним брокером таких підтверджень від усіх інших брокерів. Для того, щоб знизити обумовлену такою процедурою високу загальну латентність запису даних у ФХС пропонується, по-перше, удосконалити метод VIP-Grab таким чином, щоб строк формування останніх корекцій номерів у

послідовності загального порядку у кожному брокері був апріорі визначеним та рівним для усіх брокерів. Такий принцип формування послідовності номерів загального порядку для вхідних пакетів зробить широкомовне відправлення повідомлень-підтверджень про формування остаточно виправлених номерів пакетів усіма брокерами надмірним та необов'язковим, що буде означати зміну моделі синхронізації та відповідно зниження загальної латентності запису даних на один RTT.

По-друге, пропонується замість пари $\left(key, N_{gl}\right)$ використовувати ключ $PR\left(Br_x, N_{gl}\right)$ спеціального формату, у якому вже буде міститися відкоригований за апріорно визначений термін номер N_{gl}, ідентичний для цього пакету у всіх брокерах.

По-третє, зміни попередніх двох пунктів призводять до формулювання нового правила з отримання дозволу на запис корисних даних, а саме: запис даних можливий лише тоді, коли до ППЗ кожного брокеру вже потрапили і корисні дані, і відповідний ключ $PR\left(Br_x, N_{gl}\right)$ з відкоригованим номером N_{gl}, отже, вхідний пакет з ключем $PR\left(Br_x, N_{gl}\right)$ буде записано у репліку поточного брокеру Br_x за умови, що вхідний пакет з номером $\left(N_{gl}-1\right)$ вже було записано у цю репліку.

Запропоновані зміни у принципах обробки запитів на запис даних відобразяться на методах мережевої передачі пакетів у ФХС. Обсяги корисних даних від кінцевих користувачів до ППЗ можуть бути суттєво більшими, ніж розміри метаданих-ключів, які необхідні для синхронізації запитів та мають бути доставленими усім іншим CSP. Для гарантованої передачі пакетів з ключами $PR\left(Br_x, N_{gl}\right)$ між брокерами пропонується використовувати протокол TCP.

Особливості формування послідовності номерів пакетів загального порядку протягом апріорно визначених інтервалів. Нехай для впорядкування вхідних пакетів часова лінія розглядається як безперервна послідовність рівнотривалих непересічних інтервалів коригування *I*. У загальному випадку тривалість інтервалу коригування

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

має бути пропорційна середньому часу доставки пакетів між максимально віддаленими брокерами ФХС, тобто, якщо з одного брокеру був відправлений пакет, то протягом максимально допустимого інтервалу він гарантовано буде доставлений до будь-якого іншого брокеру. Усі брокери мають один початковий номер інтервалу коригування.

Будемо вважати, що від початку кожного інтервалу коригування для кожного брокеру визначене власне вікно доступності $W_x, x \in [1 \ldots Max]$. Тривалість вікна доступності пропорційна середньому часу доставки пакетів від географічного кластеру кінцевих користувачів до цього брокеру, як до найближчого закріпленого за цим кластером користувачів. Термін після спливу вікна доступності W_x до початку наступного інтервалу коригування назвемо залишковим інтервалом ΔI_x, тобто, $I_x = W_x + \Delta I_x, x \in [1 \ldots Max]$

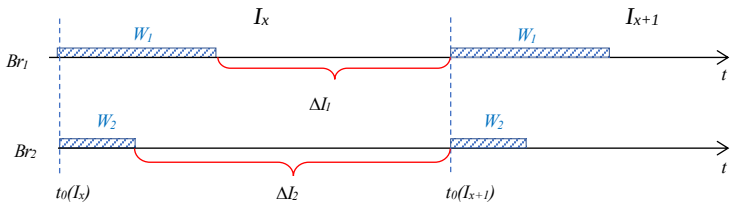

Рис.2. Тривалість вікон доступності та залишкових інтервалів для різних брокерів

Також визначимо, що максимально дозволений термін запізнення будь-якого пакету при транспортуванні між брокерами дорівнює подвоєному інтервалу коригування, тобто, $2I$, отже, сортуванню підлягатиме фактично обмежена вибірка з генеральної сукупності, що складається з фіксованої кількості монотонно зростаючих номерів пакетів, які надійшли до брокеру протягом попередніх двох вікон доступності та двох залишкових інтервалів. При цьому сортування має відбуватися послідовно у кожному інтервалі окремо, враховуючи номери пакетів починаючи з самого раннього.

Визначимо ключ $PR\left(Br_x, N_{gl}\right)$ як кортеж елементів:

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

$$PR\left(Br_x(to), N_{gl}(to)\right) =$$
$$= \left\langle Br_x(from), N^W_{temp}(from), N_{temp}(from), N_{gl}(from), I(from), \right.$$
$$\left. N^W_{temp}(to), N_{temp}(to), N_{gl}(to), I(to) \right\rangle$$

де $Br_x(from)$ – брокер-джерело, до якого надійшов пакет з даними від користувача, $x \in [1 \ldots Max]$;

$N^W_{temp}(from)$ – порядковий номер поточного пакету серед інших, що надійшли до брокера-джерела $Br_x(from)$ протягом вікна доступності W_x;

$N_{temp}(from)$ – порядковий номер поточного пакету серед інших, що надійшли до брокера-джерела $Br_x(from)$ протягом залишкового інтервалу ΔI_x;

$N_{gl}(from)$ – номеру пакету у загальній послідовності у брокері-джерелі $Br_x(from)$ на момент формування ключа;

$I(from)$ – номер інтервалу коригування, протягом якого до брокера-джерела $Br_x(from)$ надійшов пакет від користувача;

$N^W_{temp}(to)$ – порядковий номер поточного пакету серед інших, що надійшли до брокера $Br_x(to)$ протягом вікна доступності W_x;

$N_{temp}(to)$ – порядковий номер поточного пакету серед інших, що надійшли до брокера $Br_x(to)$ протягом залишкового інтервалу ΔI_x;

$N_{gl}(to)$ – поточна версія номеру пакету у загальній послідовності у брокері $Br_x(to)$;

$I(to)$ – номер інтервалу коригування, протягом якого до брокеру $Br_x(to)$ надійшов пакет з ключем.

У брокері, який отримав пакет від користувача, елементи ключа $N^W_{temp}(from)$, $N_{temp}(from)$, $N_{gl}(from)$, $I(from)$ залишаються порожніми, а $Br_x(from) = Br_x(to)$.

Запропонований формат ключа до запиту на запис корисних даних містить групу параметрів про джерело пакету та групу параметрів про місце отримання пакету. Такий формат ключа не дозволяє повторне використання протягом визначеного терміну узгодженості запитів 2^I

ключів з однаковим складом групи параметрів про джерело пакету, а отже дозволяє підтримувати лише одноразове застосування даних від користувачів в інших брокерах.

Відповідно до тривалості вікон доступності будемо визначати пріоритет брокерів при упорядкуванні вхідних пакетів, тобто, серед пакетів, що надійшли до одного й того ж брокеру протягом одного інтервалу сортування, найменші номери у загальній послідовності отримають ті, що надійшли від брокеру з найдовшим вікном доступності незалежно від дійсної хронології потраплянь цих пакетів. Цей принцип надає змогу "пропустити вперед" віддалених користувачів та підлаштуватися до дійсної хронології запитів від користувачів, які теоретично могли би бути розташовані далі від закріпленого за їхнім кластером брокеру у момент надсилання запиту на запис даних. Окрім цього, наявність пріоритетів обслуговування запитів від різних клієнтських кластерів, що надійшли протягом одного й того ж інтервалу чи навіть одночасно та мають на меті змінити один й той же запис, унеможливлює появу конфліктів порядків для конкурентних записів.

Вплив особливостей методу формування загальної послідовності номерів вхідних пакетів на синхронізацію отримання дозволів на запис даних у ФХС. Розглянемо ФХС з чотирма брокерами $Br_1 - Br_4$, для яких визначено вікна доступності $W_1 - W_4$. Розглянемо ситуацію, коли визначена тривалість вікон доступності для кожного брокеру співвідносяться між собою так: $W_1 > W_2 > W_3 > W_4$, що визначає відповідні пріоритети у сортуванні пакетів у межах одного брокера як: $Pr_1 > Pr_2 > Pr_3 > Pr_4$. Також припустимо, що найменшим вільним номером у загальній послідовності вхідних пакетів є 8.

Нехай протягом одного інтервалу коригування I_{10} до брокеру Br_4 у момент $t_8(Br_4)$, тобто, в межах вікна доступності W_4 у I_{10}, надійшов пакет з даними від користувача, де для нього сформовано ключ $PR(Br_4, 8) = \langle Br_4, \varnothing, \varnothing, \varnothing, \varnothing, 1, \varnothing, 8, 10 \rangle$, та до брокеру Br_2 у момент $t_8(Br_2)$ на інший пакет з даними від користувача сформовано ключ

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

$PR(Br_2,8)=\langle Br_2,\varnothing,\varnothing,\varnothing,\varnothing,\varnothing,1,8,10\rangle$, що пакети з цими ключами надійшли до інших трьох брокерів у моменти часу, як показано на рис.3.

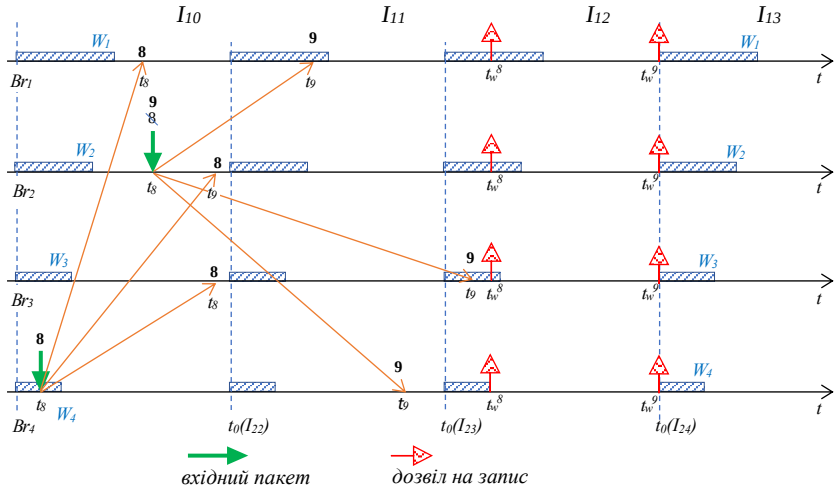

Рис. 3. Схема синхронізації отримання дозволів на запис даних з двох пакетів, що надійшли до різних брокерів протягом одного інтервалу коригування

Коли до брокеру Br_2 у момент $t_9(Br_2)$ надходить пакет з ключем $PR(Br_4,8)$, то відбувається корегування номерів обох пакетів, що надійшли протягом I_{10} до Br_2, припускаючи, що протягом попередніх інтервалів коригування $I_5 \dots I_9$ вхідних пакетів до ФХС не було. За результатами перерахунку для пакету, що надійшов у $t_9(Br_2)$, формується новий ключ $PR(Br_2,8)=\langle Br_4,1,\varnothing,8,10,\varnothing,2,8,10\rangle$, та змінюються елементи ключа до пакету, що надійшов у $t_8(Br_2)$, на $PR(Br_2,9)=\langle Br_2,\varnothing,1,8,10,\varnothing,1,9,10\rangle$. Відповідно до методу інтервального впорядкування номерів послідовності загального порядку протоколу VIP-Grab при сортуванні пакетів у цьому залишковому інтервалі спочатку мають отримати номери пакети, що було надіслано з попередніх вікон доступності. Саме тому, пакет від Br_4, який було надіслано до Br_2 ще з вікна доступності W_2 у I_{10}, отримав менший номер, навіть якщо за хронологією він потрапив до Br_2 пізніше, ніж пакет від користувача, який надійшов до Br_2 протягом ΔI_2.

Пакет від Br_2, що надійшов до Br_1 протягом наступного інтервалу I_{11}, отримає ключ $PR(Br_1,9)=\langle Br_2,\varnothing,1,8,10,1,\varnothing,9,11\rangle$, а пакет від Br_2 що надійшов до Br_3 протягом W_3 у I_{12}, буде мати ключ $PR(Br_1,9)=\langle Br_2,\varnothing,1,8,10,1,\varnothing,9,11\rangle$.

Відповідно правил сортування номерів протягом апріорно визначеного терміну $2\cdot I$, для пакету, що надійшов від користувача у Br_4 протягом W_4, усі можливі корегування $N_{gl}(to)$ мають завершитися після спливу W_4 у I_{12} та починаючи з моменту $t_w^8(Br_4)=t_0(I_{12})+W_4$ дані з цього пакету можуть бути записані, якщо дані пакету з номером $N_{gl}(to)-1$ вже отримали дозвіл на запис до поточної репліки. Аналогічно для пакету, що надійшов від користувача у Br_2 протягом ΔI_2, усі можливі корегування $N_{gl}(to)$ мають завершитися після спливу ΔI_2 у I_{12} та починаючи з моменту $t_w^9(Br_2)=t_0(I_{12})+\Delta I_2=t_0(I_{13})$ дані з цього пакету можуть бути записано в бази даних усіх CSP.

У випадку, коли протягом одного інтервалу сортування до брокеру надходять пакети від різних брокерів та користувачів, як наприклад, на рис.4 до брокеру Br_2 протягом ΔI_2 у I_{21}, формування номерів послідовності загального порядку має виконуватися відповідно до пріоритетів брокерів, від яких ці пакети надійшли.

Нехай найменший вільний номер у загальній послідовності вхідних пакетів до кожного брокеру на рис.4 є 7. У момент $t_7(Br_2)$ пакету від Br_3 буде надано номер $N_{gl}(to)=7$ та у момент $t_8(Br_2)$ після надходження пакету від Br_1 відбудеться корекція номерів: обидва вхідні пакети надійшли з попередніх вікон доступності, але $W_1>W_3$, отже, $Pr_1>Pr_3$ та пакет від Br_1 має отримати номер $N_{gl}(to)=7$, а пакет від Br_1 номер $N_{gl}(to)=8$.

У момент $t_9(Br_2)$ вхідний пакет отримає номер $N_{gl}(to)=9$, тому, що він надійшов протягом ΔI_2 від користувача, тобто, джерело пакету знаходилося у межах ΔI_2, а попередні два пакети надійшли з попередніх інтервалів сортування: з W_1 та з W_3.

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

Процедура визначення моментів t_w, починаючи з яких вхідні пакети від користувачів на рис.4 мають дозвіл на запис даних, аналогічна процедурі описаної для пакетів з рис. 3:

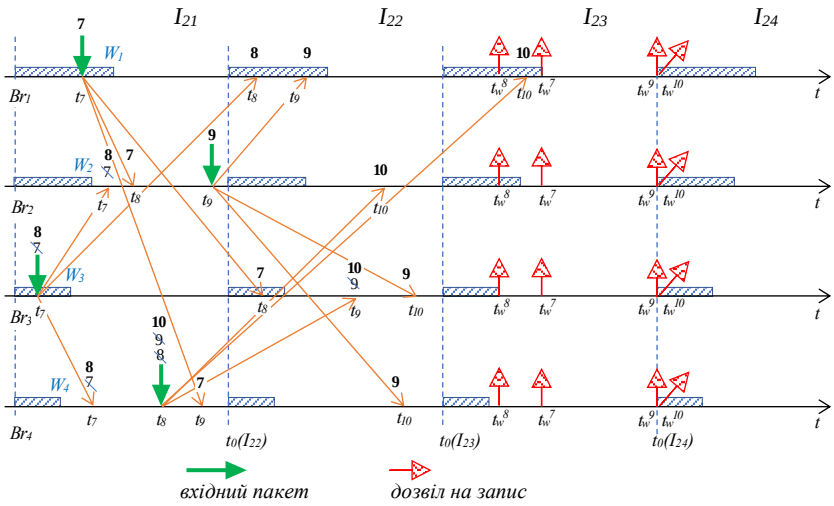

Рис. 4. Схема вирішення конфлікту порядків для записів, що надійшли протягом одного інтервалу сортування

– для пакету, що надійшов у Br_1, $t_w^7(Br_1) = t_0(I_{23}) + W_1$;

– для пакету, що надійшов у Br_2, $t_w^9(Br_2) = t_0(I_{23}) + \Delta I_2 = t_0(I_{24})$;

– для пакету, що надійшов у Br_3, $t_w^8(Br_3) = t_0(I_{23}) + W_3$;

– для пакету, що надійшов у Br_4, $t_w^{10}(Br_4) = t_0(I_{23}) + \Delta I_2 = t_0(I_{24})$.

Але у зв'язку з тим, що $W_1 > W_3$, у момент $t_w^8(Br_3)$ пакет, що надійшов до Br_1, ще не буде записаний у бази даних, тому отримання дозволу на запис даних для пакету, що надійшов до Br_3, настане лише після моменту $t_w^7(Br_1)$.

Обидва пакета, що надійшли до брокерів Br_2 та Br_4, мають однаковий термін отримання дозволу на запис даних, але дані з пакету, що надійшли до Br_4 та мають $N_{gl}(to) = 10$, будуть записані у бази даних усіх брокерів

лише після завершення запису даних з пакету, що надійшов до Br_2 та мають $N_{gl}(to) = 9$.

Таким чином, синхронізація запитів на запис у запропонованому методі забезпечується тим, що кожен брокер отримує інформацію стосовно інтервалу коригування, протягом якого поточний пакет надійшов від користувача та має принцип визначення кінцевого терміну формування відсортованого номеру цього пакету у загальній послідовності. Удосконалений метод формування загальної послідовності номерів пакетів протягом апріорно визначених інтервалів дозволить значно зменшити загальну латентність запису даних у гео-розподілених ФХС, що відповідно знизить вартість хмарних послуг при підвищенні їхньої якості для користувачів.

Висновки. В роботі удосконалено метод формування послідовності номерів загального порядку для вхідних пакетів у гео-розподілених ФХС. Запропонована процедура дозволяє вирішувати потенційні конфлікти порядків протягом сортування вхідних пакетів за апріорно визначений термін та запобігає помилковому багаторазовому запису вхідних даних у репліках.

Вперше розроблено метод синхронізації запитів на запис даних у ФХС, що базується на удосконаленому методі формування загальної послідовності номерів вхідних пакетів. Особливістю розробленого методу синхронізації є відсутність голосування реплік та групування користувачів за географічними регіонами для більш швидкого доставки пакетів до CSP.

Розроблений метод синхронізації запитів на запис може бути використаний для низько латентного фреймворку керування даними у гео-розподілених ФХС.

Список літератури:

1. *Maliye S.K.* Multi-Cloud Automation: A Strategic Approach to Cloud Infrastructure Management / *S.K. Maliye //* International Journal of Scientific Research in Computer Science Engineering and Information Technology. – 2024. – Vol. 10, Issue 6. – P. 183-190. DOI: 10.32628/CSEIT24106167

2. *Johnson O.B., Olamijuwon J., Cadet E., Osundare O.S., Samira Z.* Designing multi-cloud architecture models for enterprise scalability and cost reduction / *O.B. Johnson, J. Olamijuwon, E. Cadet, O. S. Osundare, Z. Samira* // Open Access Research Journal of Engineering and Technology. – 2024. – 07(02). –P. 101-113.

3. *Rafique A.* Middleware for Data Management in Multi-Cloud: disser. PhD: Computer Science / *A. Rafique* – Leuven (Belgium), 2019. – 222 p.

4. *Aldin H.N.S., Deldari H., Moattar M.H. and Ghods M.R.* Consistency models in distributed systems: A survey on definitions, disciplines, challenges and applications / *H.N.S. Aldin, H. Deldari, M.H. Moattar, M.R. Ghods,* 2019. [Електронний ресурс]. URL: https://arxiv.org/pdf/1902.03305v1.pdf – Дата доступа 05.02.2025 р.

5. *Campêlo R.A., Casanova M.A., Guedes D.O., Laender A.H.* A brief survey on replica consistency in cloud environments / *Campêlo R.A., Casanova M.A., Guedes D.O., Laender A.H.* // Journal of Internet Services and Applications. – 2020. – Vol.11, Article No.: 1. [Електронний ресурс]. URL: https://doi.org/10.1186/s13174-020-0122-y – Дата доступа 04.02.2025 р.

6. *Viotti P.* Consistency in Non-Transactional Distributed Storage Systems / *P. Viotti and M. Vukolić //* ACM Computing Surveys. – 2016. – Vol.49(1). [Електронний ресурс]. URL: https://doi.org/10.1145/2926965 – Дата доступа 27.01.2025 р.

7. *Sutra P.* Contributions to the practice and theory of state-machine replication / P.*Sutra* // Distributed, Parallel, and Cluster Computing. Institut Polytechnique Paris, 2024. [Електронний ресурс]. URL: https://theses.hal.science/tel-04588230v1 – Дата доступа 01.02.2025 р.

8. *Ongaro D., Ousterhout J.K.* In search of an understandable consensus algorithm / *D.Ongaro, J.K. Ousterhout* // Proceedings of USENIX ATC ‘14: 2014 USENIX Annual Technical Conference. June 19–20, 2014. – P. 305-319.

9. *Junqueira F.P., Reed B.C., Serafini M.* Zab: High-performance broadcast for primary-backup systems / *F.P. Junqueira, B.C.Reed, M. Serafini* // Proceedings of the 2011 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2011, Hong Kong, China, June 27-30 2011. IEEE Compute Society. – 2011. –P. 245-256. DOI: 10.1109/DSN.2011.5958223

10. *Renesse R.V., Altinbuken D.* Paxos made moderately complex / *R. Van Renesse, D. Altinbuken* // ACM Computing Surveys (CSUR). – 2015. – Vol. 47(3), Article No.: 42. – P. 1-36.

11. *Li J., Michael E., Sharma N.K., Szekeres A., Ports D.R.K.* Just Say NO to Paxos Overhead: Replacing Consensus with Network Ordering / *J. Li., E. Michael, N.K. Sharma, A. Szekeres, D.R.K. Ports* // Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI ‘16). – 2016. – P. 467-483.

12. *Mao Y., Junqueira F.P., Marzullo K.* Mencius: building efficient replicated state machines for WANs / *Y. Mao, F.P. Junqueira, K. Marzullo* // Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI ‘08). – 2008. – P. 369-384.

13. *Lamport L.* Fast Paxos / *L. Lamport* // Distributed Computing. 2006. – Vol. 19(2). – P. 79-103. DOI: 10.1007/s00446-006-0005.

14. *Yan X., Yang L., Wong B.* Domino: using network measurements to reduce state machine replication latency in WANs / *X. Yan, L. Yang, B. Wong* // Proceedings of the 16th International Conference on emerging Networking Experiments and Technologies (CoNEXT '20). – 2020. – P 351-363. DOI: 10.1145/3386367.3431291

15. *Moraru I., Andersen D.G., Kaminsky M.* There is more consensus in Egalitarian parliaments / *I. Moraru, D.G. Andersen, M. Kaminsky* // SOSP '13: Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles. – 2013. – P. 358-372. DOI: 10.1145/2517349.25173

16. *Eischer M., Straßner B., Distler T.* Low-latency geo-replicated state machines with guaranteed writes / *M. Eischer, B. Straßner, T. Distler* // PaPoC '20: Proceedings of the 7th Workshop on Principles and Practice of Consistency for Distributed Data – 2020. – Article No.: 13. – P. 1-9. DOI:10.1145/3380787.3393686

17. *Park S.J., Ousterhout J.* Exploiting Commutativity For Practical Fast Replication / *S.J. Park, J. Ousterhout* // NSDI'19: Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation. – 2019. – P. 47-64.

18. *Mhaisen N., Malluhi Q.M.* Data Consistency in Multi-Cloud Storage Systems With Passive Servers and Non-Communicating Clients / *N. Mhaisen, Q.M. Malluhi* // IEEE Access. – 2020. – Vol.8. – P. 164977-164986. DOI: 10.1109/ACCESS.2020.3022463

19. *Kozina O., Panchenko V., Kolomiitsev O., Stratiienko N., Usik V., Safoshkina L., Kucherenko Y.* Data consistency protocol for multicloud systems / O. *Kozina, V. Panchenko, O. Kolomiitsev, N. Stratiienko, V. Usik, I. Safoshkina, Y. Kucherenko* // International Journal of Cloud Computing. – 2024. – Vol. 13, No.1. – P. 42-61. DOI: 10.1504/IJCC.2024.136284

20. *Kozina O., Kozin M.* Simulation Model of Data Consistency Protocol for Multicloud Systems / O. *Kozina O., M. Kozin* // Proceedings of IEEE 3rd KhPI Week on Advanced Technology (KhPI Week). – 2022. – P. 730-733. DOI: 10.1109/KhPIWeek57572.2022

References

1. Maliye S.K. (2024) 'Multi-Cloud Automation: A Strategic Approach to Cloud Infrastructure Management', *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, vol. 10, Issue 6, pp. 183-190. DOI: 10.32628/CSEIT2410616

2. Johnson O.B., Olamijuwon J., Cadet E., Osundare O.S., Samira Z. (2024) 'Designing multi-cloud architecture models for enterprise scalability and cost reduction', *Open Access Research Journal of Engineering and Technology*, Vol. 07(02), pp. 101-113.

3. Rafique A. (2019) '*Middleware for Data Management in Multi-Cloud: disser. PhD: Computer Science*', Leuven (Belgium), 222 p.

4. Aldin H.N.S., Deldari H., Moattar M.H. and Ghods M.R. (2019) 'Consistency models in distributed systems: A survey on definitions, disciplines, challenges and applications', [online] https://arxiv.org/pdf/1902.03305v1.pdf (Accessed 05 February 2025)

5. Campêlo R.A., Casanova M.A., Guedes D.O., Laender A.H. (2020) 'A brief survey on replica consistency in cloud environments', *Journal of Internet Services and Applications*,

Vol.11, Article No.: 1, [online] https://doi.org/10.1186/s13174-020-0122-y (Accessed 04 February 2025).

6. Viotti P. and Vukolić M. (2016) 'Consistency in Non-Transactional Distributed Storage Systems', *ACM Computing Surveys,* Vol. 49(1) [online] https://doi.org/10.1145/2926965 (Accessed 27 January 2025)

7. Sutra P. (2024) 'Contributions to the practice and theory of state-machine replication', *Distributed, Parallel, and Cluster Computing. Institut Polytechnique Paris*, [online]. https://theses.hal.science/tel-04588230v1 (Accessed 01 February 2025)

8. Ongaro D., Ousterhout J.K. (2014) 'In search of an understandable consensus algorithm', *Proceedings of USENIX ATC '14: 2014 USENIX Annual Technical Conference*. *June 19–20*, pp. 305-319.

9. Junqueira F.P., Reed B.C., Serafini M. Zab: (2011) 'High-performance broadcast for primary-backup systems', *Proceedings of the 2011 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2011, Hong Kong, China, June 27-30 2011*. *IEEE Compute Society*, pp. 245-256.

10. Renesse R.V., Altinbuken D. (2015) 'Paxos made moderately complex', *ACM Computing Surveys (CSUR)*,Vol. 47(3), Article No.: 42., pp. 1-36.

11. Li J., Michael E., Sharma N.K., Szekeres A., Ports D.R.K. (2016) 'Just Say NO to Paxos Overhead: Replacing Consensus with Network Ordering', *Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, pp. 467-483.

12. Mao Y., Junqueira F.P., Marzullo K. (2008) 'Mencius: building efficient replicated state machines for WANs', *Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '08),* pp. 369-384.

13. Lamport L. (2006) 'Fast Paxos', *Distributed Computing*, Vol. 19(2), pp. 79-103.

14. Yan X., Yang L., Wong B. (2020) 'Domino: using network measurements to reduce state machine replication latency in WANs', *Proceedings of the 16th International Conference on emerging Networking Experiments and Technologies (CoNEXT '20),* pp. 351-363.

15. Moraru I., Andersen D.G., Kaminsky M. (2013) 'There is more consensus in Egalitarian parliaments', *SOSP '13: Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. pp. 358-372.

16. Eischer M., Straßner B. and Distler T. (2020) 'Low-latency geo-replicated state machines with guaranteed writes', *PaPoC '20: Proceedings of the 7th Workshop on Principles and Practice of Consistency for Distributed Data*, Article No.: 13, pp.1–9.

17. Park S.J., Ousterhout J. (2019) 'Exploiting Commutativity For Practical Fast Replication', *NSDI'19: Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation*, pp. 47-64.

18. Mhaisen N., Malluhi Q.M. (2020) 'Data Consistency in Multi-Cloud Storage Systems With Passive Servers and Non-Communicating Clients', *IEEE Access*, Vol.8, pp. 164977-164986.

19. Kozina O., Panchenko V., Kolomiitsev O., Stratiienko N., Usik V., Safoshkina L., Kucherenko Y. 'Data consistency protocol for multicloud systems', *International Journal of Cloud Computing*, Vol. 13, No.1. pp. 42-61.

20. Kozina O., Kozin M. (2022) 'Simulation Model of Data Consistency Protocol for Multicloud Systems', *Proceedings of IEEE 3rd KhPI Week on Advanced Technology (KhPI Week)*, pp. 730-733.

Статтю представив д.т.н., проф. Харківського національного технічного університету радіоелектроніки А.А. Коваленко.

Надійшла (*received*) *08.01.2025*

Volk Maksym, Dr. Tech. Sci., Professor
Kharkiv National University of Radio Electronics,
Nauky Ave. 14, Kharkiv, Ukraine, 61166
Tel.:+38-050-343-42-90, e-mail: maksym.volk@nure.ua;
ORCID ID: 0000-0003-4229-9904

Kozina Olha, PhD Tech., CEO
AFOREHAND Studio,
Rudika str., 6, Kharkiv, Ukraine, 61070,
Tel.:+38-050-526-46-60, e-mail: kozina4common@gmail.com;
ORCID ID: 0000-0003-0740-7068

Kozin Mykyta, Master student
Kharkiv National University of Radio Electronics,
Nauky Ave. 14, Kharkiv, Ukraine, 61166
Tel.:+38-050-409-96-65, e-mail: nlkita.kozin51@gmail.com;
ORCID ID: 0009-0003-5965-9491

УДК 004.75

Метод синхронізації запитів на запис даних у федеративних хмарних системах / Волк М.О., Козіна О.А., Козін М.Д. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2025. – № 1 (13). – С. 80 – 98.

У статті запропоновано метод синхронізації даних у гео-розподілених федеративних хмарних системах (ФХС), які не використовують засоби провайдерів хмарних послуг для керування спільними даними. У розробленому методі синхронізація запитів на запис даних від користувачів, на відміну від моделей, що використовують консенсуси реплік, забезпечується шляхом формування номерів вхідних пакетів загального порядку протягом фіксованого апріорно визначеного терміну, що є однаковим для усіх реплік. У роботі удосконалена процедура загального упорядкування вхідних пакетів з протоколу VIP-Grab, використання якої для синхронізації дозволить вирішити потенційні конфлікти порядків, запобігти багаторазовому запису вхідних даних у репліках та знизити латентність запису ФХС. Іл.: 4. Бібліогр.: 20 назв.

Ключові слова: федеративні хмарні системи; синхронізація операцій; реплікація.

UDC 004.75

Method for synchronizing data write requests in federated cloud systems / Volk M.O., Kozina O.A., Kozin M.D. // Herald of the National Technical University "KhPI". Series of "Informatics and Modeling". – Kharkov: NTU "KhPI". – 2025. – № 1 (13). – P. 80 – 98.

A method for data synchronization in geo-distributed federated cloud systems (FCSs) that do not use cloud service providers tools to manage shared data is proposed in the article. In the developed method, synchronization of data write requests from users contrary to models that use replicas consensus, is ensured by forming fully ordered numbers of incoming packets within a fixed apriori defined period equal for all replicas. The procedure for general ordering of incoming packets from the VIP-Grab protocol, the use of which for synchronization will allow resolving potential order conflicts, preventing multiple writing of incoming data in replicas, and reducing the latency of FCS writing is improves in the paper. Figs.: 4. Refs.: 20 titles.

Keywords: federated cloud systems; synchronization of operations; replication.