

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Козіну Микиті Дмитровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Методи обробки даних в хмарних системах

затверджена наказом по університету від “ 21 ” квітня 2025 р. № 296 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2025 р.

3. Вхідні дані до роботи 1) мультихмарні системи; 2) георозподілені ресурси провайдерів хмарних послуг; 3) інтенсивність вхідного потоку 7-500пак/сек.

4. Перелік питань, що потрібно опрацювати у роботі _____

1) огляд найуживаніших методів обробки даних в хмарних системах;

2) вибір та обґрунтування методики та засобів дослідження;

3) програмна верифікація методу узгодження вхідних даних у георозподілених дата-центрах мультихмарних систем;

4) проведення експериментальних досліджень;

5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____

Слайд-презентація – 12 слайдів _____

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

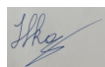
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	22.04.25-29.04.25	
2	Розробка алгоритмів обробки даних у георозподілених дата-центрах	30.04.25-05.05.25	
3	Реалізація алгоритмів	06.05.25-10.05.25	
4	Розробка структури програмних засобів	11.05.25-21.05.25	
5	Розробка програмних модулів	22.05.25-02.06.25	
6	Оформлення матеріалів кваліфікаційної роботи	03.06.25-05.06.25	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	06.06.25-10.06.25	
8	Подання кваліфікаційної роботи на рецензування	11.06.25-12.06.25	

Дата видачі завдання 22 квітня 2025 р.

Студент _____



(підпис)

Керівник роботи _____



(підпис)

проф. Волк М.О.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 69 с., 14 рис., 2 табл., 2 дод., 29 джерел.

МУЛЬТИХМАРНА СИСТЕМА, УЗГОДЖЕНІСТЬ ДАНИХ, РЕПЛІКАЦІЯ, ЗАТРИМКА, ПОСЛІДОВНІСТЬ ЗАГАЛЬНОГО ПОРЯДКУ, ІМІТАЦІЙНА МОДЕЛЬ.

Метою кваліфікаційної роботи є дослідження методів обробки даних в хмарних системах. Вирішувалась проблема розробки методу узгодження вхідних даних у георозподілених репліках мультихмарних систем.

У ході виконання кваліфікаційної роботи було розроблено алгоритм впорядкування номерів вхідних пакетів у послідовність загального порядку та метод узгодження пакетів за цими номерами у георозподілених мультихмарних систем. Результати програмної верифікації розробленого методу узгодження показали стійкість затримки узгодження вхідних запитів на запис при підвищенні інтенсивності вхідного потоку у 70 разів.

ABSTRACT

Master's thesis: 69 pages, 14 figures, 2 tables, 2 appendices, 29 sources.

MULTI-CLOUD SYSTEM, DATA CONSISTENCY, REPLICATION, DELAY, GENERAL ORDER CONSEQUENCE, SIMULATION MODEL.

The purpose of the qualification work is to study data processing methods in cloud systems.

The problem of developing a method for coordinating input data in geographically distributed replicas of multi-cloud systems was solved. During the qualification work, an algorithm for arranging the numbers of incoming packets in a general order sequence and a method for coordinating packets by these numbers in geographically distributed multi-cloud systems were developed. The results of software verification of the developed coordination method showed the stability of the delay in coordinating incoming write requests when the intensity of the incoming flow increases by 70 times.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ.....	8
ВСТУП	9
1 ОСОБЛИВОСТІ АРХІТЕКТУРИ МУЛЬТИХМАРНИХ СИСТЕМ	11
1.1 МХС як георозподілене середовище	11
1.2 Функціональні особливості ППЗ георозподіленої МХС	13
1.3 Роль ППЗ для забезпеченні несуперечності даних у георозподілених МХС	16
2 РЕПЛІКАЦІЯ ТА УЗГОДЖЕНІСТЬ ДАНИХ У МХС	20
2.1 Вимоги до спільної інфраструктури МХС	20
2.2 Особливості реалізацій моделей узгодженості даних через реплікацію у МХС.....	23
2.3 Протоколи реплікації кінцевого автомату.....	26
2.4 Алгоритмічні протоколи реплікації даних у МХС.....	29
3 ВИЗНАЧЕННЯ КОРЕКТНИХ НОМЕРІВ ВХІДНИХ ПАКЕТІВ МХС У ПОСЛІДОВНОСТІ ЗАГАЛЬНОГО ПОРЯДКУ	31
3.2 Інтервальні величини при узгодженні вхідних пакетів	33
3.3 Принцип формування ключа-ідентифікатора пакетів.....	37
3.4 Алгоритм Corr коректування номерів вхідних пакетів у послідовність загального порядку	39
4 МЕТОД УЗГОДЖЕННЯ ВХІДНИХ ДАНИХ У ГЕОРОЗПОДІЛЕНИХ МХС.....	44
4.1 Загальний опис методу узгодження вхідних пакетів до ДЦ МХС	44
4.2 Програмна верифікація методу узгодження вхідних пакетів у георозподілених репліках МХС	45
4.3 Аналіз експериментальних досліджень	47
ВИСНОВКИ.....	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	53

ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	57
ДОДАТОК Б Лістинг модулів імітаційної моделі.....	64
Б.1 Файл laws3.m.....	64

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

ДЦ – дата-центри

ППЗ – проміжне програмне забезпечення

МХС – мультимарні системи

СКВ – середнє квадратичне відхилення

ХП – хмарні послуги

CRUD – операції створення, читання, оновлення і видалення (англ., Create, Read, Update, Delete)

DBaaS – база даних як послуга (англ., Data Base As a Service)

PACS – група параметрів: продуктивність, гнучкість, вартість, безпека (англ., Performance, Agility, Cost, Security)

RTT – час кругової затримки (англ., Round-Trip Time)

SLA – згода про рівень обслуговування (англ., Service Level Agreement)

ВСТУП

Одночасне використання ресурсів декількох провайдерів ХП приносить все більше переваг для компаній з широкого кола галузей і різними бізнес-потребами. МХС надають можливість створювати високонавантажені георозподілені додатки та сховища даних з еластичним керуванням при оптимізації надійності та витрат на обслуговування.

Розвиток і впровадження хмарних технологій є основним прогнозованим трендом у сфері ІТ останніх років. Аналітики компанії International Data Corporation, що спеціалізується на дослідженнях ринку інформаційних технологій, пов'язують цю тенденцію з поширеністю багатохмарності і гібридних хмарних систем, які дозволять більш повно і ефективно задовольняти постійно зростаючі потреби в обслуговуванні великих даних і глобальних веб-сервісів.

Мультихмарні системи надають можливість ІТ-підприємствам відповідно до потреб їхнього бізнесу використовувати ресурси та послуги декількох провайдерів хмарних послуг. 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 георозподілені 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.

Потреба у МХС може виникнути для реалізації різних вимог до рівня обслуговування, надійності, тривалості відгуку, вартості обслуговування або ступеня відповідальності за наявність оновлених достовірних даних кінцевих користувачів у складних інформаційних системах. Наприклад, безперебійне функціонування інформаційних медичних систем передбачає надійне

зберігання та безперервну доступність даних пацієнтів, що ефективно реалізується у вигляді обов'язкової реплікації даних пацієнтів у хмарних базах даних, що обслуговуються різними провайдерами. При цьому технічні чи фінансові проблеми одного із хмарних провайдерів медичної системи не призводять до збоїв в обслуговуванні пацієнтів.

МХС з географічно рознесеними репліками баз даних є оптимальним рішенням, наприклад, для інформаційних систем, у яких найважливішим параметром якості обслуговування кінцевих користувачів є мінімальна затримка відгуку. В інформаційних системах подібного виду чим ближче розташування дата-центрів хмарних провайдерів з даними кінцевих користувачів, які і є ініціаторами оновлення даних, тим швидше відображаються достовірні дані.

Використання кількох провайдерів хмарних послуг може бути корисно для підвищення надійності та доступності даних в георозподілених системах, проте це пов'язано з поки ще не до кінця вирішеними проблемами з управлінням доступом і узгодження даних в МХС. Потреба отримувати актуальні узгоджені дані у найшвидший термін з будь-якої репліки у георозподілених МХС породжує декілька науково-технічних задач, які на сьогодні не мають однозначного рішення.

1 ОСОБЛИВОСТІ АРХІТЕКТУРИ МУЛЬТИХМАРНИХ СИСТЕМ

1.1 МХС як георозподілене середовище

Використання комбінації хмарних служб від різних постачальників може бути використано для усунення обмежень одного постачальника та створення більш багатих сервісів, які можуть поєднувати послуги та ресурси на вимогу від різних постачальників. Під МХС мається на увазі хмарна система, що комбінує сервіси, які належать різним провайдерам, в той час як гібридна хмарна система може об'єднувати приватні та публічні хмари одного власника сервісів. У гібридній хмарі використання загальної оркестрації сервісів і додатків дозволяє переміщати навантаження між приватними та публічними хмарами одного провайдера.

МХС можуть інтегрувати багато різних типів ресурсів: від віртуальних апаратних ресурсів до сервісів на рівні програми. Для цього необхідно об'єднане керування ресурсами на сервері. Для впровадження та розгортання в реальному часі це має включати безпечне виявлення ресурсів, облік і аудит.

Залежно від типу ресурсу, що пропонується користувачам, хмарні сервіси, як правило, групуються в такі основні моделі: програмне забезпечення як послуга (SaaS), платформа як послуга (PaaS), інфраструктура як послуга (IaaS) та DBaaS, коли служба посилається на базу даних. Модель DBaaS забезпечує прозорі механізми для створення, зберігання, доступу та оновлення баз даних. Крім того, постачальник послуг бази даних бере на себе повну відповідальність за адміністрування бази даних, таким чином гарантуючи резервне копіювання, реорганізацію та оновлення версій. При цьому обчислювальні послуги, до яких мають доступ клієнти хмар, повністю абстрактні. З одного боку абстракція забезпечує ключові переваги хмарних систем, а саме спільний доступ до ресурсів, час фізичне розташування не має значення, та приховування фонових деталей надання ІТ-сервісу від клієнта.

Але, з іншого, отримання доступу до ресурсів лише у одного провайдера хмарних послуг має суттєві недоліки, особливо для компаній, що мають справу з великими обсягами даних з підвищеними вимогами до надійності та доступності, як наприклад, банківські установи чи державні медичні заклади.

Будемо розглядати МХС як георозподілене середовище, що використовує ресурси ДЦ різних провайдерів ХП. 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, але при цьому сервіс, що надається такою МХС для кінцевих користувачів, потребує надійного та швидкого доступу до даних.

Провідні хмарні провайдери, включаючи AWS, Azure, Google, IBM, Oracle, Mirantis, Red Hat і VMware, конкурують за клієнтів, які переносять свої робочі навантаження в гібридні хмарні середовища, проте все частіше озвучується необхідність опрацювання концепції мультихмарності та загальної екосистеми компаній, для якої потрібен не тільки більш загальний шар абстракції, що може поєднати різні середовища, а й спеціальні сервіси, здатні забезпечувати ефективне управління багатьма хмарами. Деякі компанії вже заявили про створення різних інструментів, призначених для поліпшення інтеграції хмарних сервісів різних власників. Так, наприклад, команда розробки Kubernetes, що розвиває платформу для оркестровки контейнерів з наданням більш високого рівня абстракції, ніж традиційні платформи управління хмарами, додала в нову версію свого програмного забезпечення функції, які дозволяють стороннім провайдерам зберігання додавати свої плагіни в кластери робочої конфігурації.

В роботі [2] експериментально показано, що розгортання веб-сервісів в дата-центрах паралельно у трьох провайдерів хмарних послуг – Amazon EC2, Google Compute Engine і Microsoft Azure – знижує затримку відгуку більш ніж на 20% в порівнянні з використанням цих же веб-сервісів в дата-центрах тільки якого-небудь одного хмарного провайдера. Автори вважають, що основною причиною такого зниження є більш ефективна маршрутизація між

IP адресами кінцевих користувачів і найближчим центром обробки даних.

1.2 Функціональні особливості ППЗ георозподіленої МХС

Георозподілені МХС можуть пропонувати послуги, наближені до користувачів, потенційно збільшить кількість хмарних центрів обробки даних, які можна об'єднати в мультихмару для виконання вимог додатків. У такому сценарії необхідні механізми для забезпечення більш динамічної інфраструктури, щоб впоратися з вимогами мобільних користувачів, які генерують і споживають дані на межі, включаючи динамічне об'єднання хмар для надання еластичних послуг відповідно до мобільного та змінного попиту.

Під час вибору конфігурації та створенні архітектури МХС оцінка продуктивності мережі стає першочерговою проблемою. Той факт, що МХС перетинатимуть різні адміністративні домени, підкреслює необхідність управління федерацією ресурсів різних провайдерів. Ідентифікація та авторизація повинні бути належним чином застосовані в мультихмарному ППЗ, щоб бути придатним для використання в реальному часі. Користувачі МХС можуть належати до різних організацій і надавати різні типи ідентифікаційних даних. Це вимагатиме керування ідентифікацією на клієнтській стороні.

Мультихмарне ППЗ відіграє центральну роль у забезпеченні поєднання ресурсів і послуг від різних хмарних постачальників, завдяки створенню платформи розподіленої системи, яка може їх ефективно координувати. Для надання клієнтам прозорих сервісів для великих масштабованих проектів ДЦ, що належить декільком власників, потрібно узгоджувати свою роботу. Координація між ресурсами залучених провайдерів у МХС виконується не засобами окремих провайдерів, а реалізується повністю на клієнтській стороні розподілених додатків чи сервісів у ППЗ.

Автори багатьох робіт вважають, що в загальному випадку ППЗ МХС

може виконувати наступні типи завдань в залежності від специфіки галузі використання мультимарних систем:

- об'єднання різнорідних ресурсів з різних адміністративних доменів;
- забезпечення взаємодії програмного забезпечення різних провайдерів, включаючи використання механізмів аутентифікації і авторизації;
- аналіз і комбінування різних моделей білінгу в різних сервісах;
- управління місцем розташування, узгодженням і несуперечністю даних;
- безперервний моніторинг параметрів і тривалості виконання завдань.

Функціонально ППЗ можна розглядати як централізований блок управління мультимарних систем, а структурно – як підсистему управління мікросервісами, наприклад, мікросервісом для збору і моніторингу параметрів в кожній хмарі; мікросервісом узгодження даних; мікросервісом координування користувальницького трафіку; мікросервісом маршрутизації операцій читання/запису даних користувачів та інші.

Узагальнена схема інтеграції ресурсів різних провайдерів ХП за допомогою ППЗ наведено на рисунку 1.1.

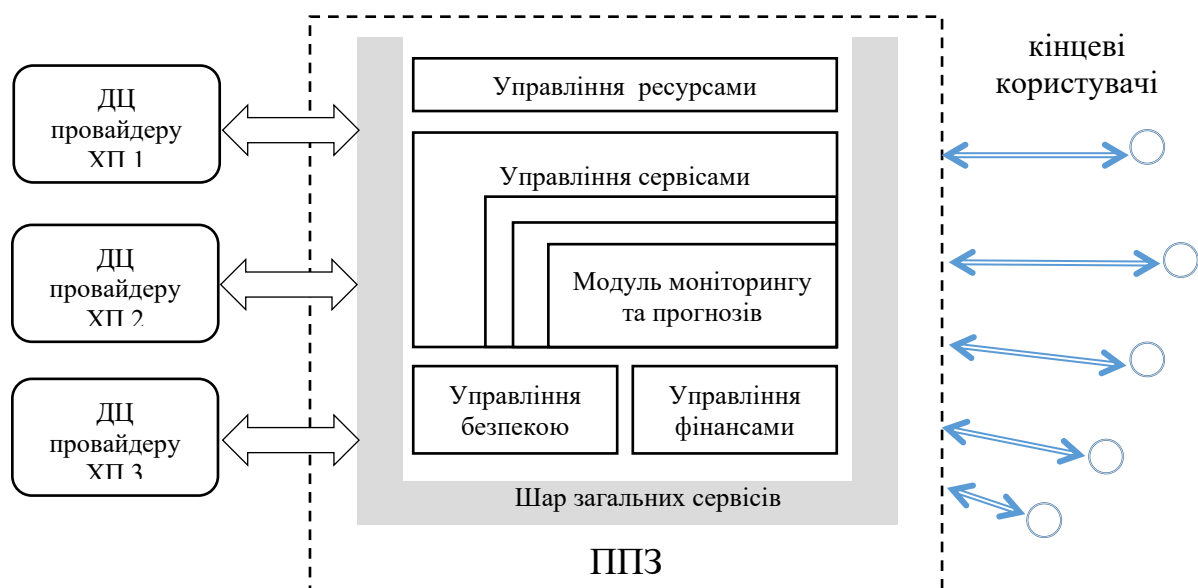


Рисунок 1.1 – Узагальнена схема інтеграції ресурсів різних провайдерів ХП за допомогою ППЗ

В роботі [3] ППЗ розглядається як адаптивна платформа, яка виконує багатоцільову оптимізацію мультихмарного сховища даних, яке може постійно розвиватися. Для вирішення завдання адаптивного ефективного розміщення даних в декількох хмарах автори пропонують виділити 4 рівні в архітектурі ППЗ: рівень мультиарендності, рівень додатків, рівень адаптивного управління даними і рівень масштабування/зміни середовища мультихмарного сховища. Рівень мультиарендності та рівень додатків відповідають за конфігурацію серверних частин сервісів мультихмарних систем і налаштування специфікацій угоди про SLA. Рівень адаптивного управління даними автономно ідентифікує поведінку окремих систем зберігання і зміни в операційному середовищі, а останній рівень ППЗ надає єдиний програмний інтерфейс взаємодії з драйверами різних баз даних і систем зберігання, реалізованих у кожного хмарного провайдера.

У роботі [4] проведено порівняльний компромісний аналіз архітектур ППЗ для менеджменту даних у мультихмарних сховищах різного типу. Через величезну складність, неоднорідність і нестійкість конфігурацій МХС автори запропонували групування вимог, які повинні враховуватися та підтримуватися додатками, що розгортаються, керуються та виконуються у хмарних сховищах. На їхній базі було розроблено декілька архітектур ППЗ для обраних сценаріїв динаміки обробки даних клієнтами МХС, у тому числі, фреймворк з можливостями самонастроювання для керування даними в МХС. Розроблений авторами фреймворк постійно адаптується до безперервних змін у робочому середовищі та на основі зібраних показників та SLA приймає рішення, щодо управління даними та автономної реконфігурації архітектури МХС. Слід зазначити, що сформульовані авторами групи вимог, за якими були розроблені архітектури ППЗ, мають здебільшого якісний характер, тому не можуть бути аналітичною базою об'єктивного порівняння МХС та, відповідно, вибору оптимального складу провайдерів ХП.

1.3 Роль ППЗ для забезпеченні несуперечності даних у георозподілених МХС

Багато провайдерів ХП генерують та підтримують у своїх клієнтів прив'язаність, тобто економічний стан, за якого клієнт залежить від технології, продуктів або послуг конкретного постачальника настільки, що перехід до іншого конкурента стає неможливим або досить дорогим. Різні провайдерам ХП підтримують різні технології зберігання даних, формати та протоколи доступу до даних, змушуючи клієнтів хмари уникати міграції між провайдерами хмарних послуг через таку несумісність. У зв'язку з цим необхідні активні зусилля зі стандартизації для забезпечення переносимості даних між різнорідними хмарними системами.

Створення та використання МХС з георозподіленими репліками даних зводить до мінімуму прив'язаність до провайдера ХП, а також значно знижує ризик втрати даних при збою у одному провайдера ХП гарантуючи при цьому високу доступність та відновлення даних. Перехід до таких МХС пов'язаний зі складним вибором провайдерів та архітектури усєї розподіленої системи, який потребує наявності об'єктивних метрик порівняння та кількісних оцінок властивостей ХП, що можуть отримувати клієнти.

Автори роботи [5] запропонували метод порівняльного аналізу масштабованості хмарних додатків, що базується на метриці попиту на ресурс та метриці завантаження ресурсу. Розроблений комплекс допомагає швидко оцінити, чи можуть певні обсяги розподілених ресурсів впоратися з інтенсивністю навантаження, але такі метрики не можуть бути повноцінними оцінками МХС загалом.

Дослідження [6] демонструє переваги використання теорії нечіткої логіки у вирішенні проблеми оптимального вибору ресурсів провайдерів ХП на основі продуктивності, гнучкості, вартості та безпеки, тобто, так званої

групи параметрів PACS. Автори запропонували модель вибору мультимарних ресурсів з точки зору клієнта, який бажає оптимальних рішень для розгортання своїх високо навантажених додатків. Ця модель містить два рівні механізму нечіткого логічного висновку для оцінки провайдерів хмарних послуг: оцінки потрібних клієнту факторів PACS та оцінки провайдерів хмарних послуг відповідним факторам PACS. Розроблений оптимізатор попередньо рекомендує інтелектуальний набір налаштованих рішень для хмарних шаблонів, оптимізованих на основі критеріїв PACS для мультимарних ресурсів. Однак, використання лише параметрів PACS звужує потенціал комплексного порівняння МХС та не дає змоги оцінити, наприклад, узгодженість та доступність даних у різних конфігураціях георозподілених МХС.

Незалежно від бізнес специфіки МХС з розподіленими репліками баз даних, її ефективне управління залежить від архітектури ППЗ та методів реалізації обраної моделі несуперечності даних у системі. Використання георозподілених МХС породжує складності з узгодженістю даних та поширенням оновлень, ініційованих клієнтом, тобто підвищується ризик отримання користувачами застарілих або невірних даних, наприклад, якщо частина неструктурованого контенту зберігається у декількох репліках.

При цьому, змагання за нових клієнтів хмарних сервісів, яким пропонується високий ступінь узгодженості даних, призводить до підвищення кількості операцій з надання високої доступності і, відповідно до високої собівартості ХП.

Вимоги до ППЗ МХС продовжуватимуть постійно змінюватися, і виникатиме більше проблем, які дослідницьке співтовариство має вирішити. Балансування продуктивності і доступності, як і підтримка адаптивного рівня узгодженості в цих масштабованих об'єктно-орієнтованих георозподілених МХС залишається проблемою. Відсутністю стандартів і загальноприйнятих підходів в проектуванні ППЗ, наряду з великою кількістю різноманітних завдань, які повинні виконуватися ППЗ, є на сьогоднішній день відомим

фактором, який обмежує широке впровадження георозподілених МХС. Ця проблема загострюється, коли ПЗЗ також потребує інтеграції внутрішніх платформ, рішень, ресурсів, спеціалізованого апаратного забезпечення та застарілих програм для повного рішення, і раптово з'являється безліч нових потреб, що ускладнює розробку ПЗЗ для багатьох МХС.

Для виконання завдань управління ресурсами георозподіленої МХС ПЗЗ, як об'єднання взаємодіючих мікросервісів, повинно мати: з одного боку – швидкий канал передачі даних між керованими об'єктами, тобто ресурсами кожного провайдера, з іншого боку – кінцеві користувачі повинні отримувати доступ до актуальних та несуперечливих даних МХС через мікросервіси ПЗЗ. Саме тому, для георозподіленої МХС структура і фізичне місце розташування ПЗЗ грає важливу роль в забезпеченні допустимої затримки відгуку.

Необхідність компромісу між продуктивністю та несуперечністю даних змусила дослідників шукати різні моделі узгодження даних у розподілених системах. Загалом, модель узгодженості визначає що має бути записано та прочитано у передбачений час різними процесами чи клієнтами при наявності одночасного доступу до одного ресурсу. Усі моделі узгодженості, що розроблено для розподілених систем, можна умовно поділити на три категорії: моделі, що орієнтовані на дані, моделі, що орієнтовані на клієнтів, та гібридні моделі.

Моделі узгодженості даних реалізуються за допомогою різноманітних методів та протоколів, переваги яких змінюються залежно від архітектури як бізнес-додатку, для якого вони розроблені, так і від архітектури та особливостей мережевої інфраструктури самої хмарної системи. Однак, визначити результуючий остаточний рівень узгодженості у МХС, на відміну від хмарної системи одного провайдера, достатньо складно як теоретично, так і практично. Це пов'язано з тим, що провайдери ХП, чії ресурси залучаються до МХС, можуть підтримувати різні рівні узгодженості даних. Крім цього, у архітектуру МХС може бути додано ПЗЗ, яке зазвичай виконує

функції керування, синхронізації, розміщення та узгодженості даних між провайдерами ХП. При цьому розподілені сервіси такого ППЗ можуть мати власні моделі узгодженості даних. Таким чином, з теоретичної та практичної точки зору модель узгодженості МХС є неоднорідною [7]. За таких обставин прийнятний рівень узгодженості даних при реплікації у різних МХС можна оцінювати за затримкою відгуку, тобто, за зменшенням непередбачуваності часу відповіді при операціях запису та читання з будь-якої репліки георозподіленої МХС. Тому, розробка та дослідження методів, що дозволяють отримати прийнятний рівень затримки операцій запису та читання даних у георозподілені репліки МХС є вкрай важливими та потрібними багатьом користувачам хмарних платформ саме зараз.

2 РЕПЛІКАЦІЯ ТА УЗГОДЖЕНІСТЬ ДАНИХ У МХС

2.1 Вимоги до спільної інфраструктури МХС

Реплікація даних у МХС має забезпечувати несуперечливість та актуальність даних, що зберігаються у ДЦ різних провайдерів ХП. При цьому реалізована у МХС модель узгодженості даних має гарантувати, що у випадку, коли кілька користувачів намагатимуться змінити один й той же запис одночасно, жодна інформація ні в якому провайдері ХП не буде втрачена.

Реплікація даних між георозподіленими ДЦ підвищує доступність даних у МХС, але є і основним викликом для розробників МХС [8]. Механізм реплікації даних має гарантувати користувачам ХП необхідний рівень узгодженості даних, тобто, надавати несуперечливі та актуальні дані на запити до реплік різних провайдерів ХП у гарантований термін.

Використання моделі DBaaS дозволяє постачальникам ХП копіювати та налаштовувати свої дані на кількох серверах, які можуть бути фізично розділені, навіть розміщені в різних центрах обробки даних. Роблячи це, вони можуть задовольнити зростаючі вимоги, спрямовуючи користувачів на найближчий або останній доступ до сервера. Надійна спільна інфраструктура зберігання даних є ключовим аспектом для забезпечення такого способу зберігання даних у МХС, щоб усі ресурси могли ефективно використовуватися спільно зі зменшенням проблем узгодженості. Найважливіші вимоги, які повинні враховуватися моделлю спільної інфраструктури:

- автоматизація: зберігання даних має бути автоматизованим, щоб мати можливість швидко вносити зміни в інфраструктуру, необхідні для підтримки узгодженості репліки без втручання людини;
- доступність: зберігання даних має гарантувати, що дані

продовжують бути доступними на необхідному рівні продуктивності в ситуаціях, що варіюються від нормальних до несприятливих;

- еластичність: сховище даних повинне мати можливість не тільки масштабуватися із збільшенням навантаження, але й пристосуватися до зменшення навантаження шляхом вивільнення хмарних ресурсів, гарантуючи при цьому дотримання необхідного рівня SLA;

- відмовостійкість: сховище даних має мати можливість відновлюватися у разі збою, наприклад, забезпечуючи резервну копію програми, яка буде готова взяти на себе роботу без збоїв;

- низька затримка; сховище даних має вирішувати проблеми затримки шляхом вимірювання та тестування затримки мережі, перш ніж зберігати дані, які змінила програма, і перед тим, як зробити ці дані доступними для інших програм;

- продуктивність: зберігання даних має забезпечувати інфраструктуру, яка підтримує швидкий і надійний доступ до даних, оновлення та відновлення;

- надійність: зберігання даних має забезпечувати можливість відновлення даних у разі аварії;

- масштабованість: сховище даних має швидко масштабуватися відповідно до вимог робочого навантаження, забезпечуючи таким чином горизонтальну та вертикальну масштабованість. Горизонтальна масштабованість стосується здатності збільшувати потужність шляхом додавання більшої кількості машин. Вертикальна масштабованість, з іншого боку, стосується збільшення потужності шляхом додавання додаткових ресурсів, наприклад, більше пам'яті або додатковий процесор.

Теорема CAP була запропонована Брюером як гіпотеза і згодом доведена у обмеженій формі Гілбертом і Лінчем [9]. Ця теорема стала важливою концепцією в хмарних системах, яка встановлює, що при розгляді бажаних властивостей узгодженості, доступності та стійкості до розділення у розподілених системах можна одночасно досягти не більше двох із них.

Очевидно, що теорема CAP створює конфлікти та накладає ряд проблем на розподілені системи та постачальників ХП. Серед конфліктів, враховуючи, що розділення мережі є неминучим у територіально розподіленому середовищі, ми виділяємо компроміс між узгодженістю та доступністю, приклад якого наведено на рисунку 2.1: користувач 2 виконує запит на читання для елемента даних D1 у репліці R3 у ДЦ2, після того, як користувач 1 оновив елемент даних D1 у репліці R1 у ДЦ1 за наявності розділу мережі, який ізолює два центри обробки даних. Враховуючи, що мережевий розділ означає, що оновлення, виконане користувачем 1, не було розповсюджено на репліку R3, є два можливі сценарії: обидві репліки можуть бути доступними без затримки, але користувач 2 читатиме застарілі дані, порушуючи таким чином узгодженість, або користувач 2 повинен чекати, доки репліку R3 буде оновлено, таким чином порушуючи безперервну доступність.

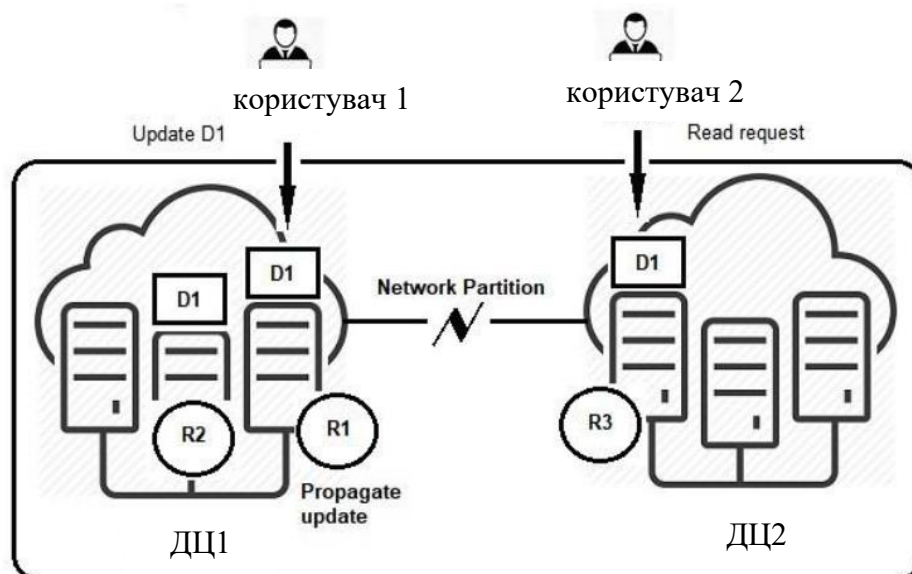


Рисунок 2.1 – Компроміс між узгодженістю та доступністю

Потреба у високій доступності та відмовостійкості спонукає великі підприємства підтримувати кілька копій своїх баз даних на кількох машинах у вигляді реплікам. Отже, реплікація дозволяє досягти таких функцій, як швидкий доступ, покращена продуктивність і вища доступність. Таким

чином, реплікація є важливою особливістю МХС.

Особливо складна проблема, яка виникає в контексті хмарних систем зберігання з георозподіленою реплікацією даних, полягає в тому, як досягти узгодженого стану в усіх репліках.

2.2 Особливості реалізацій моделей узгодженості даних через реплікацію у МХС

Модель узгодженості в розподілених середовищах визначає, які гарантії можна очікувати для операції оновлення, а також для доступу до оновленого об'єкта. Отримання правильного балансу між вищими рівнями узгодженості та доступності є однією з відкритих проблем при розробці методів реплікації, використання яких має забезпечити наявність бажаної моделі узгодженості даних у хмарних системах. Компроміси через теорему CAP вимагають вибору з ряду моделей для вирішення різних рівнів узгодженості. У цьому контексті існують дві різні перспективи, які слід розглядати в МХС зберігання даних: модель узгодженості, що орієнтована на дані та модель, орієнтована на клієнта.

З точки зору даних, розподілена система зберігання даних синхронізує операції доступу до даних з усіх процесів, щоб гарантувати правильні результати. З точки зору клієнта, система лише синхронізує операції доступу до даних того самого процесу, незалежно від інших, щоб гарантувати їх узгодженість. Така точка зору є виправданою, оскільки часто буває так, що спільні оновлення є рідкісними та мають доступ переважно до приватних даних. Отже, реалізована у МХС модель узгодженості даних має гарантувати, що у випадку, коли кілька користувачів намагатимуться змінити один й той же запис одночасно, жодна інформація ні в якому провайдері ХП не буде втрачена.

Розрізняють активні та пасивні методи реплікації у розподілених системах. При активній реплікації дані синхронізуються за допомогою

атомарної трансляції, яка гарантує, що клієнтські команди доставляються у тому порядку всім реплікам.

Навпаки, при пасивній реплікації команди виконуються однією реплікою-лідером, яка поширює оновлення стану, викликані командами, інші репліки. Цей підхід дозволяє реплікувати транзакції з недетермінованими операціями, наприклад, що залежать від тайм-аутів або переривань. Але реалізація цього механізму вимагає поширення оновлень від лідера до послідовників з використанням сильнішого примітиву, ніж класична атомарна трансляція. Це пов'язано з тим, що за пасивної реплікації оновлення стану є інкрементним стосовно стану, у якому було згенеровано. Отже, для забезпечення узгодженості між репліками кожне оновлення має застосовуватись послідовником до того ж стану, в якому воно було згенеровано лідером.

Хмарні середовища зберігання даних, які вимагають моделі узгодженості в кінцевому режимі [10], припускають, що всі зміни в реплікованих даних в кінцевому підсумку досягають усіх реплік. Однак застосування синхронної реплікації для забезпечення надійної узгодженості в такому середовищі призводить до значних накладних витрат на продуктивність через збільшену мережеву затримку між центрами обробки даних або навіть до недоступності сервісу. Це призвело до розробки методів асинхронної реплікації, які дозволяють регулювати рівень узгодженості під час виконання, щоб покращити продуктивність або зменшити витрати, зберігаючи при цьому відсоток застарілих зчитувань. При цьому для координації процесу передачі пакетів між репліками можуть бути використані наступні моделі гарантій порядку: трансляція FIFO, причинна трансляція, трансляція загального порядку, трансляція за часом, трансляція загального порядку FIFO та ширококомовна або атомарна трансляція повного причинного порядку.

Причинна трансляція – це надійна трансляція причинного порядку, тобто, якщо ширококомовне повідомлення А причинно передує трансляції

повідомлення Б, тоді жоден правильний процес не доставляє до репліки повідомлення Б, якщо він раніше не доставив повідомлення А.

Трансляція загального порядку є надійним засобом обміну повідомленнями з наступною властивістю: для будь-яких двох повідомлень А і Б, якщо будь-який процес доставляє А, не доставляючи Б, тоді жоден процес доставляє Б до А.

Трансляція за часом є надійною трансляцією зі збереженням своєчасності: якщо повідомлення А надсилається в момент часу t , та визначена константа δ , то до жодної репліки повідомлення А не буде доставлено після часу $t + \delta$.

Особливістю моделі трансляції загального порядку FIFO є забезпечення механізму трансляції, при якому якщо повідомлення А надсилається перед тим, як транслюється повідомлення Б, то до жодної репліки повідомлення Б не має бути доставленим, якщо раніше до неї не було доставлено повідомлення А.

Атомарна трансляція загального порядку представляє собою примітив, який дозволяє реалізувати протоколи реплікації кінцевого автомату. Атомарна трансляція загального порядку дозволяє впорядкувати повідомлення між репліками, щоб забезпечити надійну та послідовну систему даних. Уніфікована широкомовна трансляція загального порядку має наступні властивості:

- дійсність: усі репліки зрештою отримають повідомлення, що транслювалося;
- цілісність: кожна репліка отримує повідомлення від активної репліки щонайбільше один раз, і тільки якщо це повідомлення раніше транслювалося;
- уніфікована угода: якщо одна репліка отримує повідомлення від активної репліки, то всі інші репліки також зрештою отримають це повідомлення;
- загальний порядок: для будь-яких двох повідомлень А і Б, якщо будь-який процес доставляє повідомлення А, не доставляючи повідомлення

Б, то жодна репліка не отримає повідомлення Б до отримання повідомлення А.

2.3 Протоколи реплікації кінцевого автомату

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, наприклад, Raft [11], Zab [12], та численні варіанти протоколу Paxos: MultiPaxos [13], NOPaxos [14], Mencius [15], FastPaxos [16], Domino [17] і EPaxos [18]. 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, 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 роботі [19] запропоновано протокол PigPaxos як модифікацію протоколів реплікації з активним консенсусом, тобто, групи алгоритмів визначення лідера Paxos. Техніка довільного чергування лідерів у PigPaxos захищає вузли ретрансляції від перетворення їх у гарячі точки. Лідер у PigPaxos транслює повідомлення деякій сукупності вузлів на основі розробленого механізму комбінування зв'язків. Такий підхід призводить до покращення пропускної здатності більш ніж у 3 рази порівняно з Paxos і EPaxos та затримки видимості оновлень біля 5 мс. Однак автори розглядали

потік вхідних запитів лише від 120 користувачів між 25 вузлами AWS з одного мережевого розділу у повністю комутованому середовищі. Це означає, що використання PigRaхos у МХС, вузли яких характеризуються неоднорідним мережевим підключенням та не підтримують один мережевий розділ, може дати значно більші затримки та проблеми з масштабованістю.

Автори роботи [20] наводять результати порівняння п'яти методів географічної реплікації кінцевого автомату: MultiRaхos, Mencius, FastRaхos, Domino і ERaхos, але попередньо відзначають, що оцінка всіх можливих комбінацій протоколів реплікації кінцевого автомату для різних додатків та варіантів розміщення реплік є дуже складною задачею. Для того, щоб оцінити затримку запису та читання для кожного протоколу було створено імітаційні моделі, які враховують різні шляхи між репліками та час зв'язку між клієнтами та репліками за даними RTT між регіонами Microsoft Azure. Результати експериментів показали, що більша відстань між репліками та різні типи шляхів між репліками призводять до значної різниці в затримці між протоколами: від 150 мс до 300 мс. Це свідчить про можливість значного збільшення затримки запису при використанні цих протоколів у георозподілених МХС.

У роботі [21] представлено протокол географічної реплікації кінцевого автомату GeoRaхos+, у якому запити на запис даних надходять до найближчої до користувача репліки, яка через канали FIFO розповсюджує їх на інші репліки. Впорядкування запитів виконується асинхронно відповідно до унікальних міток часу Лампорту, які генеруються і розповсюджуються реплікою-серіалізатором, що визначається за спеціальним механізмом оптимізації. Результати імплементації GeoRaхos+ та ERaхos у трьох регіонах Amazon при різних пропорціях операцій запису та читання показали, що затримка GeoRaхos+ може варіюватися від 90 мс до 500 мс, тоді як у ERaхos до 2 с. Однак застосування GeoRaхos+ у МХС призведе до появи додаткової затримки, яка обумовлена необхідністю впровадження протоколів синхронізації часу і механізму FIFO у ДЦ різних провайдерів ХП.

У роботі [22] автори запропонували метод Weave для зменшення затримки запису даних, яка зазвичай виникає у георозподілених хмарних системах при використанні консенсусних методів. Архітектура Weave є модифікацією алгоритму Paxos, у якій для забезпечення впорядкованості гарантованих записів використовується пасивний консенсус з кількома лідерами та оптимізованим розміром кворуму. Автори розглядають послідовність гарантованих записів як надійний геореплікований журнал, який постійно зберігає зміни до стану програми і не містить фактичних результатів операцій запису даних, що є лише гарантією для користувача стосовно подальшої обробки запита у конкретний порядковий номер розподіленої черги. Такий підхід вимагає групування реплік у так звані локальні зони доступності для уникнення етапу глобального зв'язку між ними. Результати експериментів у трьох регіонах AWS в Огайо, Франкфурті та Сіднеї показали час постановки гарантованих записів у чергу близько 21 мс та час виключення з черги 161 мс. Однак ці результати можуть бути вельми іншими у зонах доступності, які будуть згруповано з реплік різних провайдерів ХП, що значно зменшує переваги запропонованого методу для МХС.

У геореплікованих системах консенсусні протоколи реплікації значно збільшують затримку відгуку, тому для реплікації даних у таких системах було запропоновано декілька спеціалізованих протоколів, наприклад, Weave та CURP [23], де використовується групування реплік і оптимізований розмір кворуму для забезпечення гарантованого запису. Однак, характеристики МХС представляють собою специфічну архітектуру для зберігання та керування даними, яка порушує багато припущень навіть спеціалізованих консенсусних методів. Зокрема, МХС 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 провайдера ХП. Крім того, теоретичні припущення, що лежать в основі систем кворуму, ігнорують багато практичних міркувань,

таких як неоднорідність обчислювальних ресурсів в різних провайдерів ХП, нерівномірність робочого навантаження та навантаження на мережу у різних регіонах, які сумісно унеможливають виконання атомарної трансляції, що необхідна для коректного функціонування протоколів реплікації кінцевого автомата. Тому, методи реплікації вхідних запитів на запис та оновлення даних, що базуються на механізмах групової розсилки та загальної впорядкованої послідовності операцій є дуже перспективними для МХС.

2.4 Алгоритмічні протоколи реплікації даних у МХС

Автори роботи [24] доводять, що публічні API МХС майже не гарантують атомарність операцій, яка є основою функціонування існуючих протоколів реплікації кінцевого автомата. Консенсус між репліками також не є обов'язковим для гарантування вимог узгодженості даних у МХС. Базуючись на цьому, автори роботи [24] роблять висновок, що можна вивести загальні алгоритмічні протоколи, які будуть забезпечувати узгодженість даних та не обов'язково ґрунтуватися на методах консенсусу. У якості підтвердження цих висновків автори пропонують алгоритм завантаження частин одного файлу до ДЦ різних ХП, який не потребує центрального модуля між користувачами та серверами хмар на відміну до методів, що використовують протокол ZooKeeper [25]. Запропонований алгоритм орієнтований тільки на роботу з файлами та не враховує географічну відстань між ДЦ, що унеможливає його використання у георозподілених МХС.

У роботах [26-29] запропоновано різні алгоритми та протоколи реплікації, що використовують позначки часу для узгодження георозподілених даних. Автори роботи [26] представили алгоритм реплікації даних для на основі методу глобальної стабілізації. Для досягнення причинно-наслідкової узгодженості автори використовують часові мітки фізичного годинника, який слабо синхронізований між усіма серверами за

допомогою протоколу синхронізації часу NTP. Автори провели експерименти у імітаційній розподіленій системі, де зв'язані у кільце сервери можуть обмінюватися пакетами лише з сусідніми серверами. При цьому для імітації мережеских затримок використана константа у 100 мс, що не відображає ні географічне розташування реплік, ні реальну випадкову природу затримок глобальної мережі навіть у FIFO-каналах. Отже, за результатами таких експериментів запропонований алгоритм не можна вважати оптимальним з точки зору затримки видимості клієнтами віддаленого оновлення для МХС.

Таким чином, розглянуті методи узгодження вхідних запитів на запис не враховують структурні особливості МХС та мають значні обмеження застосування при наявності нерівномірності робочого навантаження та навантаження на мережу у різних регіонах. Тому доцільною є розробка методів узгодження даних реплікації у МХС, що базуються на механізмах групової розсилки, алгоритмічних методах впорядкованості операцій запису та відповідає функціонально-структурним особливостям георозподілених МХС. Це призведе до підвищення ефективності функціонування георозподілених МХС та зниження рівня затримок запису даних у них.

3 ВИЗНАЧЕННЯ КОРЕКТНИХ НОМЕРІВ ВХІДНИХ ПАКЕТІВ МХС У ПОСЛІДОВНОСТІ ЗАГАЛЬНОГО ПОРЯДКУ

3.1 Елементи МХС з узгодженням вхідних запитів за номерами загальної послідовності

Нехай МХС поєднує ресурси чотирьох провайдерів ХП. З кожного провайдера ХП обрано один ДЦ, у якому розміщено ППЗ, тобто розташування брокерів $Br_x, x = [1; BrMax] = [1; 4]$ відповідає географічному розташуванню ведучих ДЦ. Кожний ДЦ має репліку даних кінцевих користувачів. При цьому кінцеві користувачі згруповані в умовні кластери за географічним розташуванням та їхні запити надходять до найближчого брокеру Br_x . При обробці вхідних запитів на оновлення чи запис даних від кінцевих користувачів такий брокер перенаправляє інформацію до всіх інших ведених брокерів, а значить, виступає при цьому активним вузлом реплікації (рисунок 3.1).

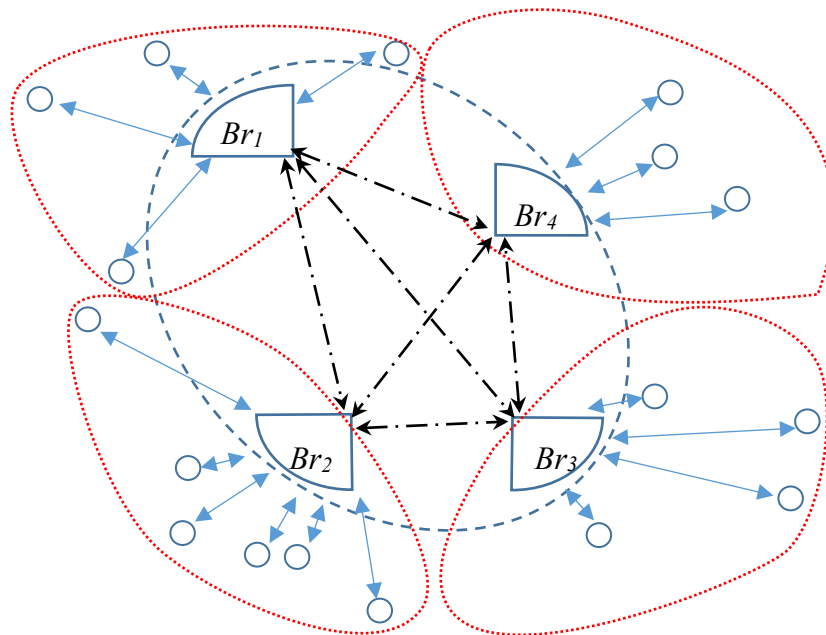


Рисунок 3.1 – Схема взаємодії ППЗ брокерів МХС при обробці запитів на запис даних

Будемо вважати, що 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. При такій схемі реплікації кожен брокер у МХС може виступати як ведучим, так і веденим. Реплікація даних між ДЦ, що належать одному провайдеру ХП, може виконуватися засобами провайдера відповідно схеми реплікації з одним ведучим вузлом.

Отже, будемо розглядати ППЗ МХС з активною асинхронною реплікацією даних з багатьма лідерами на основі загальної впорядкованої послідовності пакетів на запис даних. При цьому для гарантованої передачі пакетів між брокерами має застосовуватися протокол TCP, а 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 один брокер надіслав пакет X перед тим, як надіслати пакет Y, то кожен інший брокер може отримати пакети X і Y у довільному порядку.

Процедура реплікації вхідних пакетів з даними для оновлення чи для запису, що виконується на основі монотонно зростаючих номерів у загальній послідовності операцій, базується на використанні для кожного вхідного пакету ключа-ідентифікатора *key*. Отже, у ведучому брокері, до якого надійшов вхідний пакет від кінцевого користувача його кластеру, має бути сформований ключа-ідентифікатора *key*, який разом з даними пакету надсилається до усіх інших брокерів. Пакет, що було отримано від іншого брокеру, становиться вхідним пакетом $PR(Br_x(to), j_x)$ для поточного брокеру

Br_x . Таким чином в усіх брокерах вхідні пакети отримують номери та мають бути узгоджені у послідовність загального порядку за їхніми номерами.

3.2 Інтервальні величини при узгодженні вхідних пакетів

В залежності від площі умовного кластеру користувачів середній час доставки пакетів від користувачів до найближчого брокеру може суттєво відрізнятися в різних кластерах, що впливає на хронологію фіксації вхідних запитів. Припустимо, що користувач А з периферії кластеру великої площі надіслав запит до свого брокеру за фізичним годинником раніше, ніж користувач Б з кластеру маленького розміру, який знаходився дуже близько до свого брокеру. У такій ситуації існує велика ймовірність, що запит від користувача Б потрапить до ППЗ раніше, ніж запит від користувача А. Це означає, що хронологія фіксації моментів надходження запитів вхідних пакетів до брокерів відрізняється від дійсної хронології відправлення цих запитів користувачами. Для уникнення таких ситуацій та для вирівнювання шансів на фіксацію дійсної хронології формування запитів користувачами у різних кластерах, в межах інтервалів коригування $I_x, x \in [1 \dots BrMax]$ кожного брокеру Br_x визначається власне вікно доступності W_x . Тривалість такого вікна доступності має бути не менша ніж середній час доставки пакетів від користувачів кластеру до свого Br_x . Відтак, власний інтервал коригування кожного брокеру дорівнює:

$$I_x = W_x + \Delta I_x, \quad (3.1)$$

де ΔI_x – залишкове вікно, тривалість якого має бути не менша за середній час трансляції пакетів від активного брокеру до найвіддаленішого веденого брокеру МХС (рисунок 3.2).

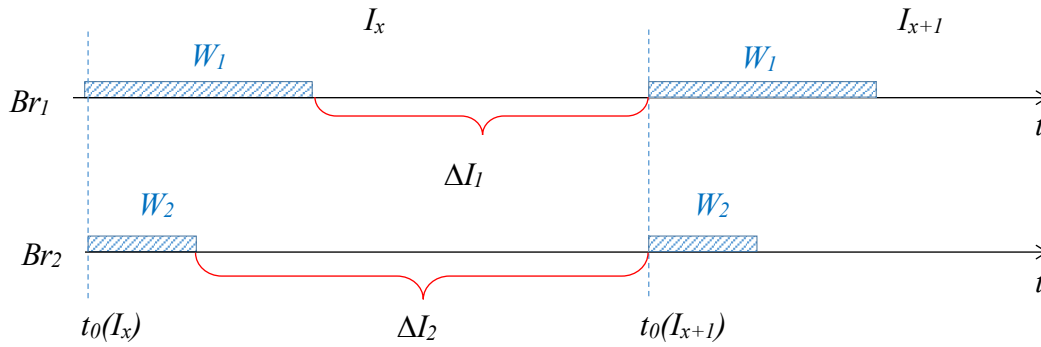


Рисунок 3.2 – Тривалість вікон доступності та залишкових інтервалів для різних брокерів

При узгодженні порядку запитів на запис у якості єдиного уніфікованого значення інтервалу коригування МХС використовується:

$$I = \max_{x=1..BrMax} I_x. \quad (3.2)$$

Для обробки вхідних пакетів апіорно визначаються пріоритети сортування $Prior_x$, які залежать від тривалості вікон доступності W_x : чим довше W_x у брокері Br_x , тим вищий пріоритет сортування мають пакети, що надійшли від нього іншим брокерам, тобто, якщо $W_1 > W_2$, то $Prior_1 > Prior_2$.

Визначимо затримку запису $Lat_{wr}(Br_x(to), j)$ пакету даних $PR(Br_x(to), j)$ з номером j при активній реплікації у георозподіленій МХС як різницю між часом $t_{wr}(Br_x(to))$ отримання дозволу на запис даних у веденому брокері $Br_x(to)$ та часом $t_u(Br_a(from))$ потрапляння пакету від користувача у найближчий брокер $Br_a(from)$ (рисунок 3.3):

$$Lat_{wr}(Br_x(to), j) = t_{wr}(Br_x(to)) - t_u(Br_a(from)). \quad (3.3)$$

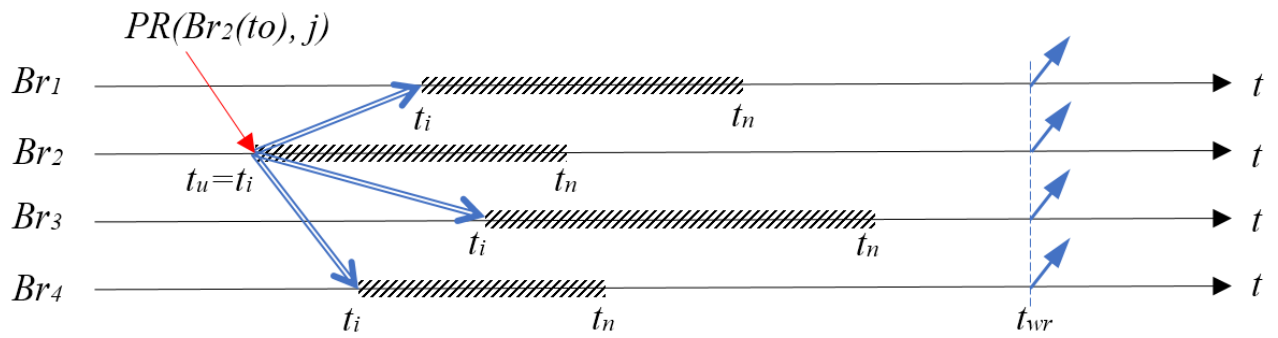


Рисунок 3.3 – Приклад розташування інтервалів, що визначають затримки запису пакетів на кожному брокері

При цьому до збігу часу $Lat_{wr}(Br_x(to), j)$ в усіх брокерах має завершитися процес впорядкування номерів вхідних пакетів. Тривалість процесу впорядкування номерів $T_{consist}(Br_x(to), j)$ визначається як різниця часу між часом $t_n(Br_x(to))$ останньої корекції номеру пакету $PR(Br_x(to), j)$ та часом доставки $t_i(Br_x(to))$ цього пакету до $Br_x(to)$:

$$T_{consist}(Br_x(to), j) = t_n(Br_x(to)) - t_i(Br_x(to)). \quad (3.4)$$

Для МХС, де для синхронізації запису даних мають використовуватися засоби комунікації через глобальну мережу, терміни доставки пакетів між брокерами, що визначені як:

$$t_{tr}(Br_a(from), Br_x(to)) = t_i(Br_x(to)) - t_u(Br_a(from)), \quad (3.5)$$

вносять суттєвий вклад у затримку формування відкоректованих номерів у загальній послідовності пакетів $Lat_{consist}(Br_x(to), j)$ та мають розглядатися як незалежні випадкові процеси. Отже, $Lat_{consist}(Br_x(to), j)$ для кожного брокера визначається як сума тривалості формування впорядкованих номерів пакетів та терміну доставки пакетів від ведучого брокера:

$$\begin{aligned}
Lat_{consist}(Br_x(to), j) &= \\
&= T_{consist}(Br_x(to), j) + t_{tr}(Br_a(from), Br_x(to)) = \\
&= t_n(Br_x(to)) - t_i(Br_x(to)) + t_i(Br_x(to)) - t_u(Br_a(from)) = \\
&= t_n(Br_x(to)) - t_u(Br_a(from)) .
\end{aligned} \tag{3.6}$$

і розглядається як випадковий процес, який окрім самої процедури інтервального узгодження номерів загальної послідовності, визначається ймовірнісними величинами t_u , t_i та t_n :

Визначимо максимально дозволений термін надходження пакетів між брокерами як інтервал максимально дозволеного запізнення $MaxLate$:

$$MaxLate = \max_{a,x=1..BrMax} t_{tr}(Br_a(from), Br_x(to)) = K_{Late} \cdot I, \tag{3.7}$$

який має бути не менший за максимальний середній час доставки пакетів між будь-якими двома брокерами. При цьому індекс запізнення K_{Late} розраховується як стеля від ділення:

$$K_{Late} = \left\lceil \frac{MaxLate}{I} \right\rceil. \tag{3.8}$$

Таким чином, сортуванню підлягатиме фактично обмежена вибірка вхідних пакетів з генеральної сукупності, що складається з фіксованої кількості монотонно зростаючих номерів пакетів, які надійшли до брокера протягом максимально дозволеного терміну надходження $MaxLate$. При цьому впорядкування пакетів має відбуватися послідовно у кожному інтервалі окремо, враховуючи номери пакетів починаючи з самого раннього.

3.3 Принцип формування ключа-ідентифікатора пакетів

Особливістю запропонованого методу впорядкування вхідних даних є обов'язкова передача між ДЦ МХС пакетів з даними *data* та їхніми унікальними ідентифікатора.

Ключ-ідентифікатор *key* кожного вхідного пакета формується у ведучому брокері, який отримав пакет від користувача. До такого ключа-ідентифікатора додається відкоригований номер пакету із загальної послідовності, що формується у кожному брокері, та пара $(key, data)$ використовується для визначення моменту, коли дані можуть бути записані у репліки МХС. Таким чином, пакет від користувача породжує два кроки: передачі корисних даних від одного брокера до усіх інших та упорядкування пакетів у брокерах за номерами у послідовності загального порядку. Після виконання цих механізмів у ППЗ виконується процедура отримання дозволу на запис корисних даних у кожному брокері, відповідно до результатів виконання обох попередніх кроків.

У якості ключа $key = PR(Br_x, N_{gl})$ можна використовувати кортеж спеціального формату, у якому буде міститися відкоригований за апіорно визначений термін номер N_{gl} , ідентичний для цього пакету у всіх брокерах. При цьому запис даних у поточну репліку можливий лише тоді, коли до кожного брокера вже потрапили і корисні дані, і відповідний ключ $PR(Br_x, N_{gl})$ з відкоригованим номером N_{gl} , отже, вхідний пакет з ключем $PR(Br_x, N_{gl})$ буде записано у репліку поточного брокера Br_x за умови, що вхідний пакет з номером $(N_{gl} - 1)$ вже було записано у цю репліку.

Визначимо ключ $PR(Br_x, N_{gl})$ як кортеж елементів [29]

$$\begin{aligned}
PR(Br_x(to), N_{gl}(to)) &= \\
&= \left\langle Br_x(from), N_{temp}^W(from), N_{temp}(from), N_{gl}(from), I(from), \right. \\
&\quad \left. N_{temp}^W(to), N_{temp}(to), N_{gl}(to), I(to) \right\rangle
\end{aligned} \tag{3.9}$$

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

$N_{temp}^W(from)$ – 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;

$N_{temp}^W(to)$ – 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 $N_{temp}^W(from) = N_{temp}^W(to)$, $N_{gl}(from) = N_{gl}(to)$, $I(from) = I(to)$ та $Br_x(from) = Br_x(to)$.

Такий формат ключа, що містить групу параметрів про джерело пакету та групу параметрів про місце отримання пакету, не дозволяє повторне використання ключів-ідентифікаторів з однаковим складом групи параметрів про джерело пакету, а отже дозволяє підтримувати лише одноразове застосування даних від користувачів в інших брокерах.

3.4 Алгоритм Corr коректування номерів вхідних пакетів у послідовність загального порядку

При надходженні кожного вхідного пакету $PR(Br_x(to), j)$ до $Br_x(to)$ формується кортеж вікон M_{late} . Номера пакетів з кортежу цих вікон M_{late} можуть бути скориговані у зв'язку з появою нового пакету. Кортеж M_{late} містить усі інтервали та вікна, починаючи з вікна-джерела $Br_x(from)$ поточного вхідного пакету, і до $Br_x(to)$. Наприклад, якщо вхідний пакет $PR(Br_3(to), j)$ надійшов протягом інтервалу $I(to)$ із залишкового вікна брокера Br_1 з індексом запізнення $K_{Late}=2$, тобто, за два інтервали коригування до поточного вікна (рисунок 3.4), то кортеж M_{late} має включати усі вікна-джерела з проміжку від $\Delta I_1(from)=\Delta I_1(to-K_{Late})=\Delta I_1(to-2)$ та до $\Delta I_1(to)$, пакети з яких вже надходили до брокера Br_3 , тобто, $|M_{late}|=5$ та:

$$M_{late} = \langle \Delta I_1(to-2), W_1(to-1), \Delta I_1(to-1), W_1(to), \Delta I_1(to) \rangle. \quad (3.10)$$

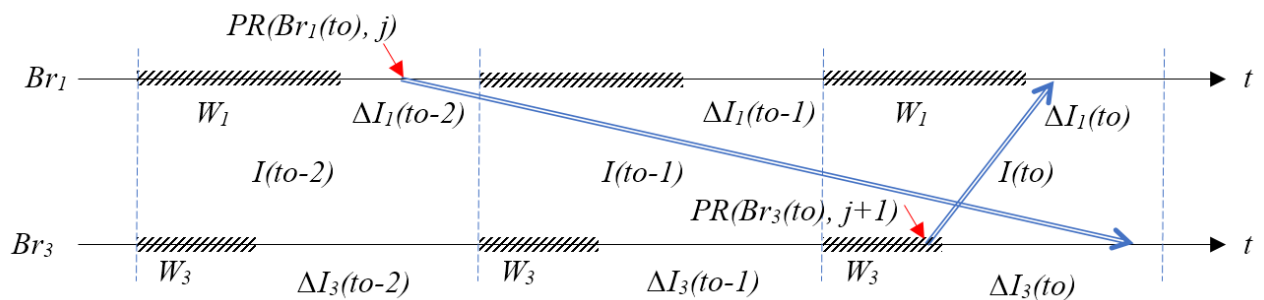


Рисунок 3.4 – Приклад формування кортежу вікон для коригування номерів

пакетів $PR(Br_1(to), j)$ та $PR(Br_3(to), j+1)$

Якщо до $Br_x(to)$ протягом $MaxLate$ не потрапляли інші пакети, тобто, $|Mcor|=0$, або коректне місце j_x має бути після усіх існуючих номерів послідовності, то виконується додавання наступного номеру у кінець вже існуючої послідовності номерів:

$$j_x = N_{free}(I(to)), \quad (3.11)$$

де $N_{free}(I(to))$ – перший вільний номер загальної послідовності поточного $I(to)$ у брокері $Br_x(to)$. Після цього $N_{free}(I(to))$ збільшується на 1. Якщо $|Mcor| \neq 0$ та коректне положення j_x знаходиться серед $\forall_{i=1 \dots |Mcor|} \exists_{x=1 \dots BrMax} PR(Br_x(to), j_x)$, то відбувається вставка номеру у існуючу загальну послідовність так, щоб не порушити попередні номери пакетів з $Mcor$, а номери усіх наступних пакетів збільшуються на 1.

Для кожного брокеру у кожному інтервалі $I(to)$ визначається величина вільного номеру загальної послідовності $N_{free}(I(to))$, а також усі брокери мають один початковий номер інтервалу коригування.

При визначенні коректного номеру вхідного пакету у існуючої впорядкованої послідовності пакетів з масиву $Mcor$ треба зберігати коректний порядок вікон, з яких надійшли пакети. Це означає, що при сортуванні номерів пакетів з $Mcor$ пакети, що надійшли з попереднього вікна, наприклад, з $\Delta I_x(from) = \Delta I_x(to - K_{Late})$ мають передувати пакетам, що надійшли з вікон $W_x(from) = W_x(to - (K_{Late} - 1))$ та $\Delta I_x(from) = \Delta I_x(to - (K_{Late} - 1))$. Крім того, пакети, що надійшли протягом одного вікна від брокерів з вищим пріоритетом повинні мати менші номери, ніж пакети, що надійшли від брокерів з меншим пріоритетом. Цей підхід надає змогу надати шанс пакетам від віддалених користувачів бути обробленими згідно дійсної хронології створення запитів від користувачів, які теоретично могли би бути розташовані далі від закріпленого за їхнім кластером брокеру у момент надсилання запиту. Окрім цього, наявність пріоритетів обслуговування

запитів від різних кластерів, що надійшли одночасно та мають на меті змінити один й той же запис, унеможливує появу конфліктів порядків для конкурентних записів. Схема описаного алгоритму Corr визначення коректних номерів вхідних пакетів серед пакетів кортежу $Mcor$ показано на рисунку 3.5.

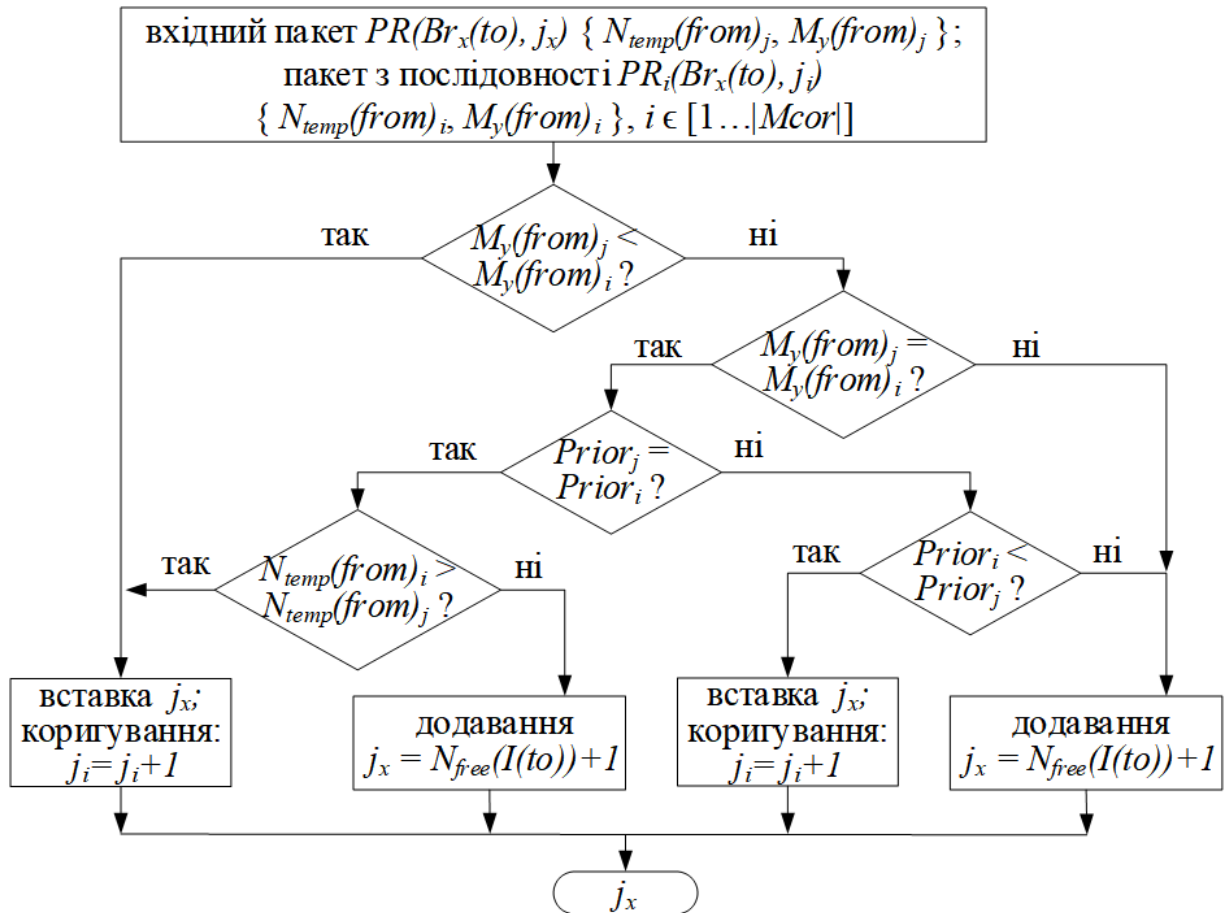


Рисунок 3.5 – Алгоритм Corr визначення коректного значення номеру j_x вхідного пакету $PR(Br_x, j_x)$ у загальній послідовності

Для пакетів, що вийшли з одного вікна, наприклад, $W_1(from)$, впорядкування у брокері $Br_x(to)$ повинно виконуватися згідно порядку їх виходу з вікна-джерела, тобто, пакет, який вийшов першим з $W_1(from)$ матиме номер менший, ніж пакет, що вийшов другим, незалежно від порядку їх надходження до $Br_x(to)$.

Розглянемо приклад використання алгоритму Corr у МХС 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 брокера співвідносяться між собою так: $W_1 > W_2 > W_3 > W_4$, що визначає відповідні пріоритети у сортуванні пакетів у межах одного брокера як: $Pr_1 > Pr_2 > Pr_3 > Pr_4$. Нехай у цьому прикладі найменший вільний номер у загальній послідовності вхідних пакетів до кожного брокера є 7 та протягом попередніх інтервалів коригування вхідних пакетів до МХС не надходило.

Коли до брокеру Br_2 протягом залишкового вікна ΔI_2 у I_{21} у момент $t_7(Br_2)$ надходить пакет від Br_3 з ключем-ідентифікатором $PR(Br_3, 7) = \langle Br_3, 1, \emptyset, 7, 21, 1, \emptyset, 7, 21 \rangle$, то його ключ визначається як $PR(Br_2, 7) = \langle Br_3, 1, \emptyset, 7, 21, \emptyset, 1, 7, 21 \rangle$, а $N_{free}(I(to)) = 8$ (рисунок 3.6).

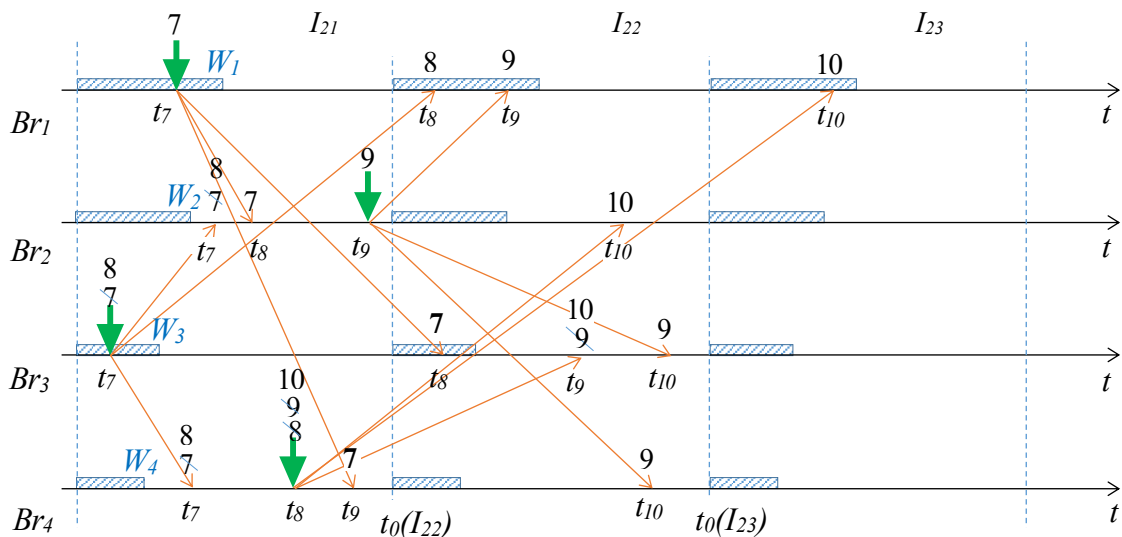


Рисунок 3.6. – Приклад впорядкування номерів пакетів за алгоритмом Sort

Коли ж до цього ж брокеру протягом того ж залишкового вікна у момент $t_8(Br_2)$ надходить пакет від Br_1 , то відбувається зміна номеру попереднього пакету на $PR(Br_2, 8) = \langle Br_3, 1, \emptyset, 7, 21, 1, \emptyset, 8, 21 \rangle$, пакет від Br_1 отримує у поточному брокері ключ $PR(Br_2, 7) = \langle Br_1, 1, \emptyset, 7, 21, \emptyset, 2, 7, 21 \rangle$ та

$N_{free}(I(to))$ на момент $t_8(Br_2)$ дорівнюватиме 9. Заміна номерів відбувається у зв'язку з тим, що пріоритет $Pr_1 > Pr_3$ отже, серед пакетів, що надійшли протягом одного залишкового вікна, найменший номер має бути у того пакета, що надійшов з брокеру з більшим пріоритетом.

Аналогічно за алгоритмом Согг у брокері Br_4 за інтервали I_{21} та I_{22} впорядкована за номерами послідовність пакетів буде мати вигляд $\langle t_9(Br_4), t_7(Br_4), t_{10}(Br_4), t_8(Br_4) \rangle$. При сортуванні цих пакетів протягом $MaxLate=2 \cdot I$ спочатку мають отримати номери пакети, які було надіслано з попередніх вікон доступності. Саме тому, пакети від Br_1 та Br_3 , які було надіслано до Br_4 з вікон доступності W_1 та W_4 у I_{21} , мають отримати менші номери, ніж пакет, що надійшов до Br_4 протягом залишкового вікна у I_{21} навіть якщо за хронологією вони потрапили до брокеру пізніше.

Отже, хронологія надходження пакетів до ведених брокерів визначається не за фізичним часом, а за черговістю $N_{temp}(to)$ надходження пакетів у межах кожного вікна $\forall_{y=1 \dots |M_{late}|} M_y(to)$, а до ведучих брокерів $Br_x(from)$ – за черговістю $N_{temp}(from)$ надходження пакетів від користувачів протягом вікна-джерела $M_y(from), y=1 \dots |M_{late}|$. Таким чином, реплікація запитів на запис на базі запропонованого алгоритму визначення коректного номеру вхідного пакету 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.

4 МЕТОД УЗГОДЖЕННЯ ВХІДНИХ ДАНИХ У ГЕОРОЗПОДІЛЕНИХ МХС

4.1 Загальний опис методу узгодження вхідних пакетів до ДЦ МХС

Метод узгодження вхідних пакетів до брокерів МХС, елементи якої розглянуто у розділі 3, складається з наступних кроків:

- за апіорно визначеними середніми значеннями часу доставки пакетів від користувачів до найближчого брокеру та часу доставки пакетів між брокерами визначаються вікна W_x та ΔI_x , інтервали коригування I_x та пріоритети $Prior_x$ для кожного брокеру, а також уніфіковане значення інтервалу коригування I та індекс запізнення K_{Late} ;

- при надходженні пакету до брокеру на базі визначених на попередньому кроці величин формується масив пакетів $Mcor$, що надійшли з кортежу вікон M_{late} , та ключ-ідентифікатор пакету з корекцією номеру у послідовності загального порядку відповідно до алгоритму Corr.

Функціональна схема узгодження номерів серед пакетів масиву $Mcor$ наведено на рисунку 4.1.

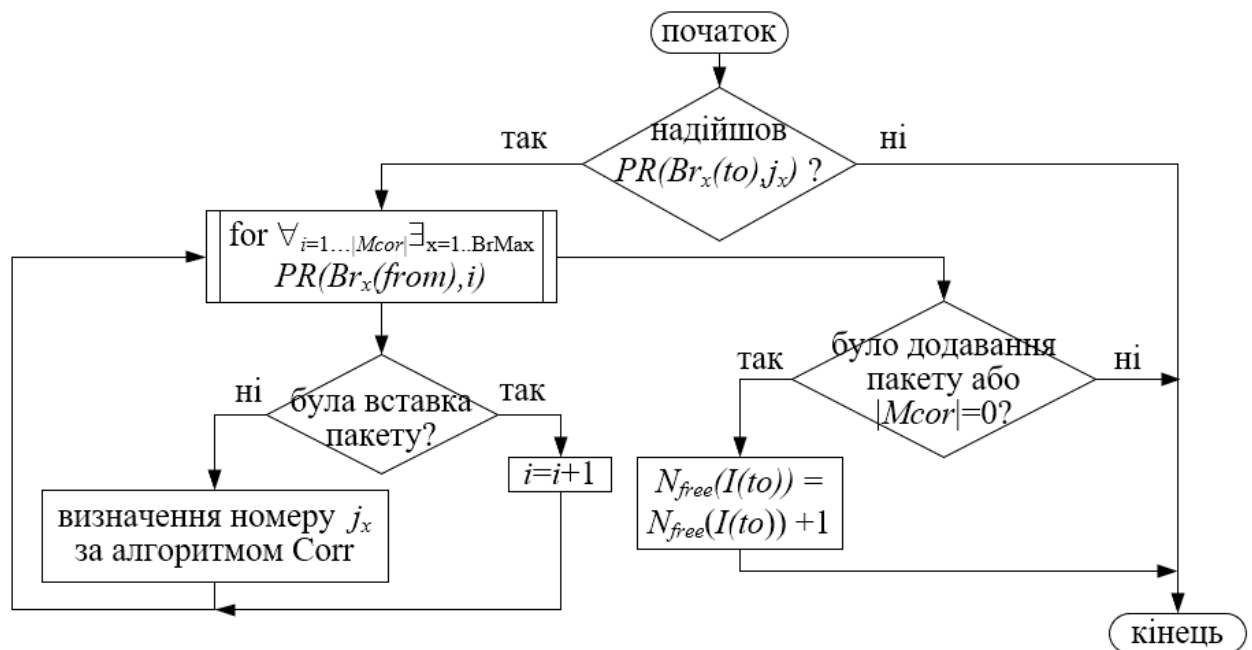


Рисунок 4.1 – Функціональна схема узгодження номерів серед пакетів $Mcor$

4.2 Програмна верифікація методу узгодження вхідних пакетів у георозподілених репліках МХС

Для програмної верифікації запропонованого методу розроблено імітаційну модель, у якій розглядається МХС з чотирьох провайдерів ХП, отже $BrMax=4$. Імітаційна модель складається з наступних блоків:

- блок вхідних констант містить: кількість брокерів у системі, тривалість I , вектор значень W_x , середні значення інтервалів між пакетами від користувачів до кожного брокеру, та інтервалів доставки пакетів між брокерами, загальна кількість вхідних повідомлень до кожного брокеру;

- блок генерації потоків пакетів від користувачів та потоків пакетів від активного брокеру згідно апріорно обраним розподілам ймовірності;

- блок формування та впорядкування номерів загальної послідовності пакетів, де також розраховані за формулою (6) затримки $Lat_{consist}(Br_x(to), j)$ групуються у окремі вибірки випадкових значень по кожному брокеру;

- блок презентації результатів для визначення, збору та візуалізації кумулятивних функцій розподілу ймовірності затримки $CDF(Lat_{consist}(Br_x(to), j)) = CDF(Lat)$ формування коректних номерів пакетів у загальній послідовності по кожному брокеру окремо.

Визначені функціональні частини імітаційної моделі для оцінки затримки виконуються послідовно (рисунок 4.2) так, що вихідні змінні кожного блоку є вхідними для наступного.

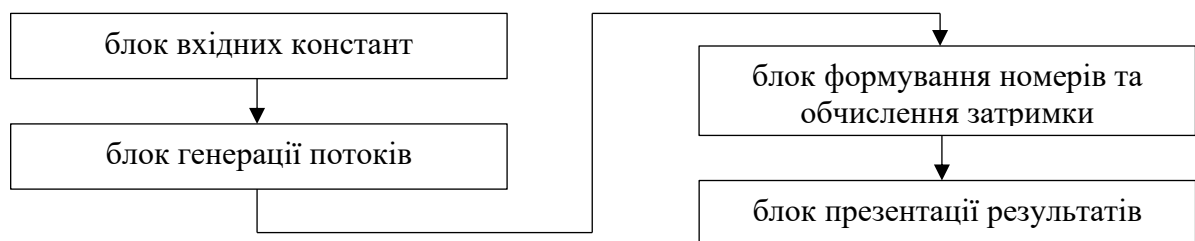


Рисунок 4.2 – Функціональна схема імітаційної моделі

Потоки вхідних запитів на запис від кінцевих користувачів та відповідно час до їх надходження до кожного брокера у моделі вважаються незалежним випадковим величинами з однаковим типом функції щільності ймовірності $PDF_{Br_a(from)}(t_u)$ та різними значеннями математичного очікування та СКВ. Це означає, що при створенні імітаційної моделі інтервального впорядкування вхідних пакетів у мультимарній системі необхідно генерувати окремі, відповідно до максимальної кількості брокерів $BrMax$ у МХС, набори випадкових чисел для відтворення потоків запитів від користувачів, щільність ймовірності яких $\forall a=1..BrMax PDF_{Br_a(from)}(t_u)$ можна визначити на основі теоретичних припущень, історичних даних для аналогічних систем або на основі даних з блоку ППЗ для збору та аналізу вхідного трафіка МХС.

В імітаційній моделі використовується один тип та однакові параметри розподілу щільності ймовірності $PDF_{Br_a(from)}(t_u)$ інтервалів між моментами $t_u(Br_a(from))$ потрапляння пакетів від кінцевих користувачів у найближчий брокер $Br_a(from)$ для всіх брокерів, а також однакова функція розподілу щільності ймовірності $PDF_{Br_a(from),Br_x(to)}(t_i)$ інтервалів доставки пакетів як у прямому напрямку від брокера-джерела $Br_a(from)$ до брокера $Br_x(to)$, так і у зворотному, але з різними математичними очікуваннями та СКВ, що має відображати різну дистанцію між брокерами. Це потребує генерації окремо $C_{BrMax}^2 = C_4^2 = 6$ різних потоків випадкових чисел з визначеними параметрами розподілу $PDF_{Br_a(from),Br_x(to)}(t_i)$.

Для відтворення нестабільності мережевого трафіку, що можуть бути пов'язані з різними незалежними причинами, як наприклад, робочі години, сезонність чи коливання фондового ринку, у імітаційних експериментах використано різні закони розподілу щільності ймовірності. До кожного брокера у кожному експерименті подається не менш ніж 50000 вхідних пакетів.

Імітаційні експерименти виконані у програмному середовищі чисельних обчислень Matlab. Лістинг імітаційної моделі наведено у додатку Б.

4.3 Аналіз експериментальних досліджень

В експериментах 1 – 3 час доставки пакетів між брокерами згенеровано за рівномірним законом з середніми значеннями $MeanBrToBr$ (мс) термінів доставки між брокерами $Br_1 - Br_4$:

$$MeanBrToBr = \begin{matrix} & Br_1 & Br_2 & Br_3 & Br_4 \\ Br_1 & \left| \begin{array}{cccc} 0 & 156 & 82 & 59 \\ 156 & 0 & 130 & 107 \\ 82 & 130 & 0 & 118 \\ 59 & 107 & 118 & 0 \end{array} \right. \\ Br_2 & \\ Br_3 & \\ Br_4 & \end{matrix} \quad (4.1)$$

та відповідними СКВ:

$$СКВ = \begin{matrix} & Br_1 & Br_2 & Br_3 & Br_4 \\ Br_1 & \left| \begin{array}{cccc} 0 & 8.0623 & 8.1048 & 8.1070 \\ 8.0425 & 0 & 8.1028 & 8.1103 \\ 8.1041 & 8.0480 & 0 & 8.0603 \\ 8.0988 & 8.1221 & 8.0839 & 0 \end{array} \right. \\ Br_2 & \\ Br_3 & \\ Br_4 & \end{matrix}$$

Також у експериментах 1 – 3 генерувалися окремі потоки інтервалів між запитами від користувачів за усіченим експоненціальним, рівномірним та Парето законами розподілу щільності ймовірності $PDF_{Br_a(from)}(t_u)$, середні значення яких наведено у таблиці 1.

Таблиця 4.1 – Середні значення $PDF_{Br_a(from)}(t_u)$ по кожному брокеру

№ експерименту	до Br_1 , мс	до Br_2 , мс	до Br_3 , мс	до Br_4 , мс
1	148	97	163	112
2	37	24	41	28
3	2	2	2	2

На рисунку 4.3 показано результати імітаційних експериментів з розрахунку кумулятивної функції розподілу ймовірності $CDF(Lat)$ затримки формування впорядкованого номеру загальної послідовності по кожному брокеру окремо. Для усіх імітаційних експериментів тривалість інтервалу коригування дорівнює $I = 295$ мс, а вектор вікон доступності для брокерів $Br_1 - Br_4$ відповідно $W_x = |90 \ 76 \ 30 \ 19|$ (мс), $x = [1...4]$.

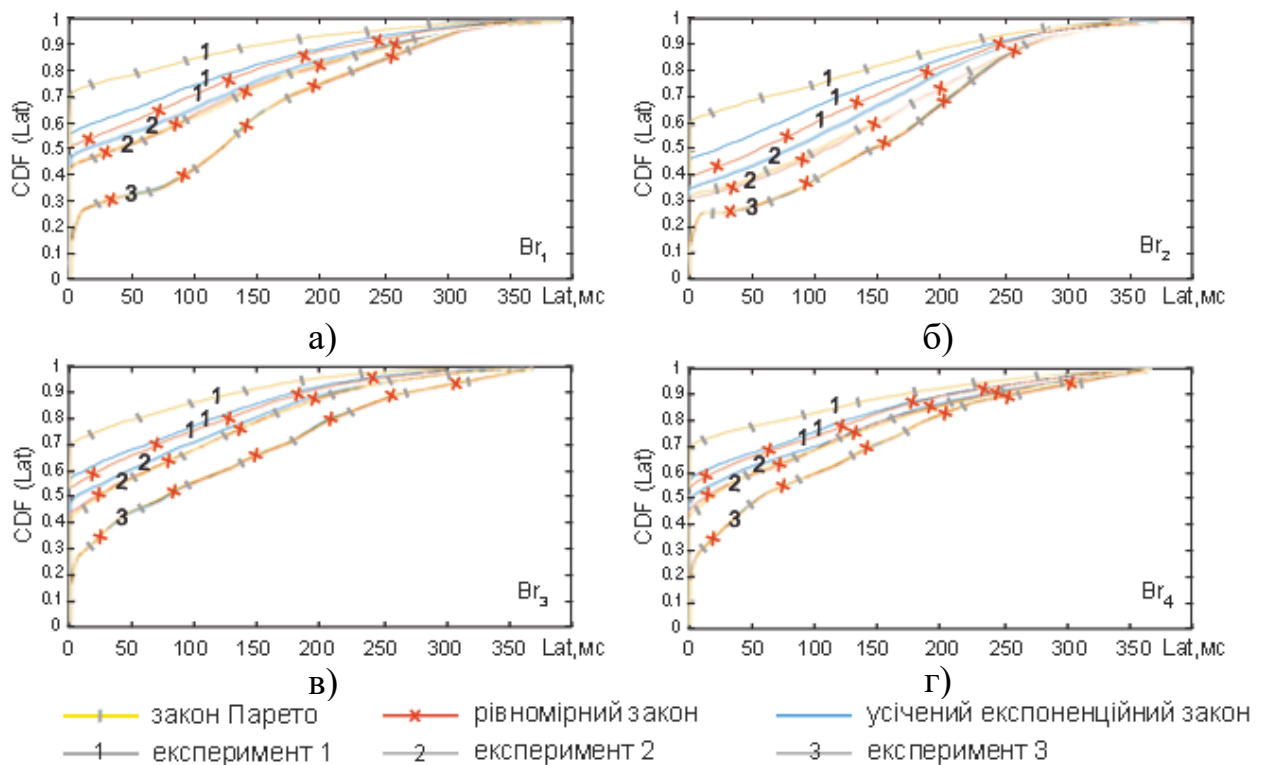


Рисунок 4.3 – Кумулятивні функції розподілу ймовірності $CDF(Lat)$ затримки формування коректних номерів загальної послідовності: а) у брокері Br_1 , б) у брокері Br_2 , в) у брокері Br_3 , г) у брокері Br_4

Криві $CDF(Lat)$ мають різні значення для рівномірного, усіченого експоненціального та Парето розподілу щільності ймовірності термінів доставки пакетів між брокерами, але мають тенденцію до повного зливання, що видно у кривій експерименту 3.

Порівняння результатів експериментів 1–3 на рисунку 4.3 показує, що

максимальна затримка узгодження вхідних пакетів не змінюється при збільшенні інтенсивності вхідних запитів навіть у 70 разів (таблиця 4.1). Це демонструє стійкість запропонованого методу узгодження вхідних пакетів у георозподілених репліках МХС до пікових вхідних навантажень.

У експериментах 4 – 7 середні значення $MeanBrToBr$ часу доставки пакетів між брокерами дорівнюють значенням матриці (4.1). У кожному з експериментів 4 – 7 міститься по три вхідних потоки $PDF_{Br_a(from)}(t_u)$ до кожного брокеру за законом Парето, усіченим експоненціальним та рівномірним (таблиця 4.2).

Таблиця 4.2 – Параметри розподілів щільності ймовірності для експериментів 4 – 7

№ експерименту	середнє значення $PDF_{Br_a(from)}(t_u)$	$PDF_{Br_a(from),Br_x(to)}(t_i)$			
		тип закону	СКВ		
4	$\begin{matrix} Br_1 & Br_2 & Br_3 & Br_4 \\ 10 & 10 & 10 & 10 \end{matrix}$	рівномірний	$\begin{matrix} & Br_1 & Br_2 & Br_3 & Br_4 \\ Br_1 & & 0 & 3.4552 & 1.7367 & 1.7372 \\ Br_2 & 3.4468 & & 0 & 2.3151 & 3.1861 \\ Br_3 & 2.3154 & 2.2994 & & 0 & 1.7272 \\ Br_4 & 1.7354 & 2.9007 & 1.7322 & & 0 \end{matrix}$		
5	$\begin{matrix} Br_1 & Br_2 & Br_3 & Br_4 \\ 10 & 10 & 10 & 10 \end{matrix}$	усічений експоненціальний	$\begin{matrix} & Br_1 & Br_2 & Br_3 & Br_4 \\ Br_1 & & 0 & 57.0506 & 19.3489 & 10.0041 \\ Br_2 & 57.1512 & & 0 & 50.1610 & 41.3886 \\ Br_3 & 19.3230 & 49.9747 & & 0 & 47.1434 \\ Br_4 & 10.0319 & 41.4167 & 47.3009 & & 0 \end{matrix}$		
6	$\begin{matrix} Br_1 & Br_2 & Br_3 & Br_4 \\ 2 & 2 & 2 & 2 \end{matrix}$	рівномірний	$\begin{matrix} & Br_1 & Br_2 & Br_3 & Br_4 \\ Br_1 & & 0 & 3.4552 & 1.7367 & 1.7372 \\ Br_2 & 3.4468 & & 0 & 2.3151 & 3.1861 \\ Br_3 & 2.3154 & 2.2994 & & 0 & 1.7272 \\ Br_4 & 1.7354 & 2.9007 & 1.7322 & & 0 \end{matrix}$		
7	$\begin{matrix} Br_1 & Br_2 & Br_3 & Br_4 \\ 2 & 2 & 2 & 2 \end{matrix}$	усічений експоненціальний	$\begin{matrix} & Br_1 & Br_2 & Br_3 & Br_4 \\ Br_1 & & 0 & 57.0506 & 19.3489 & 10.0041 \\ Br_2 & 57.1512 & & 0 & 50.1610 & 41.3886 \\ Br_3 & 19.3230 & 49.9747 & & 0 & 47.1434 \\ Br_4 & 10.0319 & 41.4167 & 47.3009 & & 0 \end{matrix}$		

Результати імітаційних експериментів 4 – 7 з розрахунку кумулятивної

функції розподілу ймовірності $CDF(Lat)$ затримки формування впорядкованого номеру загальної послідовності наведено на рисунку 4.4.

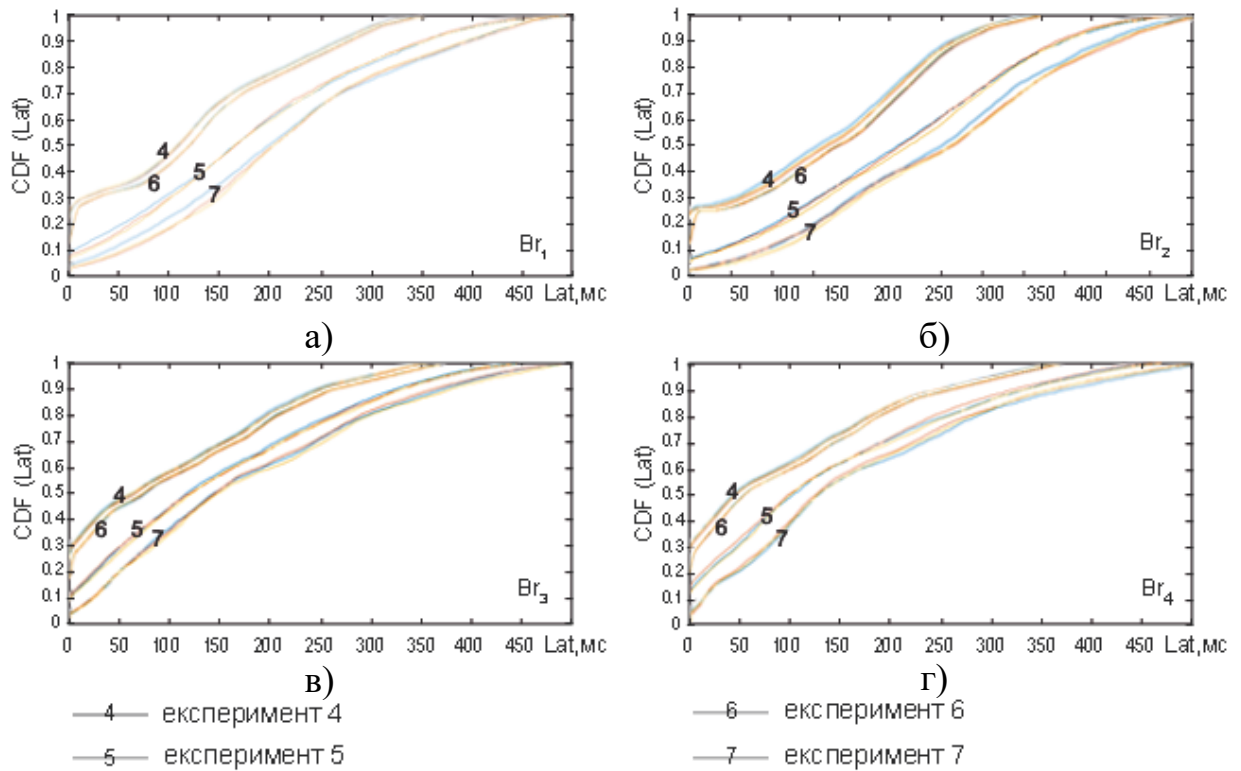


Рисунок 4.4 – Кумулятивні функції розподілу ймовірності $CDF(Lat)$ затримки формування коректних номерів загальної послідовності: а) у брокері Br_1 , б) у брокері Br_2 , в) у брокері Br_3 , г) у брокері Br_4

Аналіз результатів проведених експериментів 4 – 7 показали, що збільшення інтенсивності вхідних пакетів призводить до такого збільшення кількість пакетів у кожному інтервалі коригування I , що різниця між формами законів $PDF_{Br_a(from),Br_x(to)}(t_i)$ зникає. Отже, є сенс звертати увагу не на форму розподілу щільності ймовірності потоку вхідних пакетів особливо на високих інтенсивностях, а на СКВ $PDF_{Br_a(from),Br_x(to)}(t_i)$, що показано результатами імітаційних експериментів 4 – 7 на рисунку 4.4: чим більша дисперсія інтервалів доставки пакетів між брокерами при однакових СКВ вхідних потоків пакетів (таблиця 4.2), тим довша затримка формування коректних номерів загальної послідовності пакетів, що показує порівняння

пари кривих з експериментів 4 та 5, а також кривих з експериментів 6 та 7.

ВИСНОВКИ

На результатами огляду літературних джерел визначено поточні складнощі у ефективному застосуванні існуючих методів реплікації даних для потреб МХС. Показано, що затримка запису вхідних даних є показником ефективності методів реплікації у МХС.

Аналіз особливостей функціонування МХС дозволив розробити алгоритм Corr корекції номерів вхідних пакетів у послідовності загального порядку. Особливістю розробленого алгоритму вирішення конфліктів при формуванні загальної послідовності вхідних пакетів є групування користувачів за географічними розміщенням навкруги найближчого брокера ППЗ та використання пріоритетів обслуговування пакетів від різних брокерів.

У роботі розроблено метод впорядкування вхідних пакетів протягом апріорно визначених рівно тривалих непересічних інтервалів за алгоритмом Corr.

Для оцінки затримок при формуванні загальної послідовності номерів вхідних пакетів розроблено імітаційну модель, особливістю якої є можливість відтворення різної кількості та географічного розміщення ДЦ різних провайдерів ХП у МХС. Результати програмної верифікації розробленого методу узгодження вхідних пакетів до георозподілених реплік МХС показали стійкість затримки узгодження вхідних запитів на запис при підвищенні інтенсивності вхідного потоку у 70 разів. Отримані результати дозволяють не тільки знизити затримку запису даних у репліки вже існуючих МХС, але й обрати найкраще географічне розташування ресурсів постачальників хмарних послуг під час проектування нових.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Maliye S.K. Multi-Cloud Automation: A Strategic Approach to Cloud Infrastructure Management. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*. 2024. Vol. 10, No. 6. P. 183–190.
2. Wu Z., Madhyastha H.V. Understanding the Latency Benefits of Multi-Cloud Webservice Deployments. *ACM SIGCOMM Computer Communication Review*. 2013. Vol. 43, No 2, P. 14–19.
3. Rafique A., Landuyt D.V., Reniers V., Joosen W. Towards an Adaptive Middleware for Efficient Multi-Cloud Data Storage [Електронний ресурс] *CrossCloud'17*, April 23, 2017, Belgrade, Serbia, 2017. – Режим доступа : <https://dl.acm.org/doi/10.1145/3069383.3069387> – 17.05.2025 p.
4. Rafique A. *Middleware for Data Management in Multi-Cloud: disser. PhD: Computer Science / Rafique Ansar.* – Leuven (Belgium), 2019. – 222 p.
5. Henning S. A configurable method for benchmarking scalability of cloud-native applications. *Empirical Software Engineering*. 2022. Vol. 27:143, P. 1–43.
6. Pandey A., Calyam P., Lyu Z., Joshi T. Fuzzy-Engineered Multi-Cloud Resource Brokering for Data-intensive Applications. *IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. 2021. P. 257–266.
7. Campêlo R.A., Casanova M.A., Guedes D.O., Laender A.H. A brief survey on replica consistency in cloud environments. *Journal of Internet Services and Applications*. 2020. Vol.11, Article No.: 1. P. 1–13.
8. Tricomi G., Merlino G., Panarello A., Puliafito A. Optimal Selection Techniques for Cloud Service Providers. *IEEE Access*. 2020. Vol, 8. P. 203591–203618. <http://doi.org/10.1109/ACCESS.2020.3035816>
9. Gilbert S., Lynch N. Brewer's Conjecture and the Feasibility of

Consistent, Available, Partition-Tolerant Web Services. *ACM SIGACT News*. 2002. Vol. 33. No. 2. P. 51–59. <http://doi.org/10.1145/564585.564601>

10. Vogels W. Eventually Consistent. *Communications of the ACM*. 2009. Vol. 52. No. 1. P. 40–44.

11. Ongaro D., Ousterhout J. K. In search of an understandable consensus algorithm. *Proceedings of USENIX ATC '14: 2014 USENIX Annual Technical Conference (USENIX ATC)*. 2014. P. 305–319.

12. Junqueira F. P., Reed B. C., Serafini M. Zab: High-performance broadcast for primary-backup systems *IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*. 2011. P. 245–256.

13. Van Renesse R., Altinbuken D. Paxos Made Moderately Complex. *ACM Comput. Surv.* 2015. Vol. 47(3). No. 42. P.1–36.

14. Li J., Michael E., Sharma N. K., Szekeres A., Ports D. R. K. Just Say No to Paxos Overhead: Replacing Consensus with Network Ordering. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16)*. 2016. P. 467–483.

15. Mao Y., Junqueira F. P., Marzullo K.. Mencius: building efficient replicated state machines for WANs. In *Proc. OSDI '08*. 2008. P. 369–384.

16. Lamport L. Fast Paxos. *Distrib. Comput.* 2006. Vol.19. No.2. P.79–103.

17. Yan X., Yang L., Wong B. Domino: using network measurements to reduce state machine replication latency in WANs. In *Proc. CoNEXT '20*. 2020. P. 351–363.

18. Moraru I., Andersen D. G., Kaminsky M. There is more consensus in Egalitarian parliaments. In *Proc. SOSP'13*. 2013. P. 358–372.

19. Charapko, A., Ailijiang, A., Demirbas, M. (2021). PigPaxos: Devouring the Communication Bottlenecks in Distributed Consensus. *SIGMOD'21: Proceedings of the 2021 International Conference on Management of Data*. 2021. P. 235–247. <https://doi.org/10.1145/3448016.3452834>

20. Shiozaki, K., Nakamura, J. Selection Guidelines for Geographical SMR

Protocols: A Communication Pattern-based Latency Modeling Approach. Stabilization, Safety, and Security of Distributed Systems: 26th International Symposium. 2024. P. 344–359. https://doi.org/10.1007/978-3-031-74498-3_25

21. Coelho, P., Pedone, F. GeoPaxos+: Practical Geographical State Machine Replication. 40th International Symposium on Reliable Distributed Systems. 2021. P. 233–243. <https://doi.org/10.1109/SRDS53918.2021.00031>

22. Eischer, M., Straßner, B., Distler, T. 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. 2020. Article No.: 13. P. 1–9. <https://doi.org/10.1145/3380787.3393686>

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

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

25. Petrescu, M. Replication in Raft vs Apache Zookeeper. Proceedings of the 9th International Workshop Soft Computing Applications (SOFA 2020). Advances in Intelligent Systems and Computing. 2020. P. 426–435.

26. Xiang Z., Vaidya N. H. Global Stabilization for Causally Consistent Partial Replication. ICDCN'20: Proceedings of the 21st International Conference on Distributed Computing and Networking. 2020. Article No.: 30. P. 1–10.

27. Kakwani D., Nasre R. Orion: Time Estimated Causally Consistent Key-Value Store. PaPoC'20: Proceedings of the 7th Workshop on Principles and Practice of Consistency for Distributed Data. 2020. Article No.: 3. P. 1–6.

28. Song H., Wang Y., Chen X., Feng H., Feng Y., Fang X., Cui H., Kong L. K2: On Optimizing Distributed Transactions in a Multi-region Data Store with TrueTime Clocks. [Електронний ресурс] – Режим доступа : <https://doi.org/10.48550/arXiv.2504.01460> – 11.05.2025 p.

29. 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.