

УДК 004.75

В. Муліш, Є. Крилов

ПРИШВИДШЕННЯ ПРОЦЕСУ УЗГОДЖЕННЯ ДАНИХ У ВИСОКОНАВАНТАЖЕНИХ РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ ШЛЯХОМ ВПРОВАДЖЕННЯ РОЗПОДІЛЕНОГО ТРАНЗАКЦІЙНОГО ГОДИННИКА

Анотація: у статті описано математичну модель та запропоновано концепцію впровадження розподіленого транзакційного годинника для пришвидшення процесу узгодження даних у високонавантажених розподілених інформаційних системах. Експерименти, проведені на прикладі системи підтримки фінансових операцій із 3–7 екземплярами бази даних, підтверджують висунуте твердження. Результати експериментів демонструють скорочення тривалості процесу узгодження даних у середньому на 50 % порівняно з нерозподіленим транзакційним годинником та на 66 % із MongoDB Replica Set, а для критичних операцій — час на узгодження даних є уп'ятеро менше. Запропонований підхід, водночас, підвищує пропускну здатність системи, масштабованість та її стійкість до відмов.

Ключові слова: інформаційна система, розподілена база даних, узгодженість даних, розподілений транзакційний годинник.

Вступ

У сучасних умовах глобальної цифрової трансформації, стрімкого зростання обсягів даних та інтенсивності їх обробки, високонавантажені інформаційні системи стають критично важливими для широкого спектру застосувань — від фінансових транзакцій та електронної комерції до систем реального часу та хмарних обчислень.

За даними Statista, глобальний інтернет-трафік протягом останнього десятиліття постійно зростає. Щорічно обсяг інтернет-трафіку збільшується приблизно на 20 % [1]. Цей факт свідчить про те, що майбутнє інформаційних систем пов'язане з обробкою все більшої кількості запитів, і традиційні методи оптимізації стають недостатніми для забезпечення високої якості обслуговування користувачів.

Основними викликами, що постають перед високонавантаженими інформаційними системами, є забезпечення необхідного рівня узгодженості даних, збереження високого рівня продуктивності, масштабованості та стійкості до відмов.

© В. Муліш, Є. Крилов

Аналіз досліджень і публікацій

Більшість підходів до удосконалення способів обробки запитів клієнтів, що полягають в удосконаленні методів узгодження даних у високонавантажених розподілених інформаційних системах, стикаються з певними обмеженнями. Так, наприклад, один з підходів полягає у використанні нерозподіленого транзакційного годинника. Такий підхід покращує продуктивність системи, пришвидшує процес узгодження даних [2], але, все ж, має певні недоліки. Перш за все, це створення єдиної точки відмови у високонавантажених розподілених інформаційних системі, яка, при цьому ж, є найбільш навантаженим місцем усієї системи та, відповідно, найбільш вразливою до перенавантаження та відмов. Таке централізоване рішення не лише обмежує загальну продуктивність інформаційної системи, але й негативно впливає на її масштабованість та стійкість до відмов, що особливо важливо в умовах високих навантажень.

Постановка задачі

Метою статті є опис та розробка підходу пришвидшення процесу узгодження даних у високонавантажених розподілених інформаційних системах шляхом впровадження розподіленого транзакційного годинника. У рамках дослідження передбачається подолання наявних обмежень, що притаманні існуючим рішенням, шляхом формулювання та реалізації нової концепції, яка забезпечує не лише швидше узгодження даних, а й кращі показники продуктивності, масштабованості та відмовостійкості системи при збереженні необхідного рівня узгодженості.

Математична модель розподіленого транзакційного годинника

Перш за все, необхідно провести формалізацію математичної моделі розподіленого транзакційного годинника. Модель описує механізм періодичної агрегації запитів у транзакції із забезпеченням рівня послідовної узгодженості.

Позначимо множину усіх вузлів, що генерують команди для розподіленого транзакційного годинника у вигляді:

$$I = \{ 1, 2, \dots, N \}, \quad (1)$$

де N – загальна кількість вузлів.

Множина об'єктів даних, над якими можуть виконуватись операції, визначається як:

$$D = \{ d_1, d_2, \dots \}. \quad (2)$$

Тип операції представлено множиною:

$$\tau = \{ Create, Update, Delete \}. \quad (3)$$

Множина операцій у системі, що використовує розподілений транзакційний годинник, може бути представлена у вигляді:

$$O = \{ r = (i_r, t_r, \mu_r, d_r, \tau_r, p_r) \}, \quad (4)$$

де кожна операція r складається з: i_r – номер вузла, що її створив, $i_r \in I$, що описано в (1); t_r – час створення, $t_r \in R$; μ_r – пріоритет операції, $\mu_r \in N$; d_r – ідентифікатор об'єкту даних, $d_r \in D$, що описано в (2); τ_r – тип операції, $\tau_r \in \tau$, що описано в (3); p_r – корисне навантаження, що представляє об'єкт даних.

У вузлах розподіленого транзакційного годинника встановлюється фіксована тривалість вікна агрегації $T > 0$, яка визначає період об'єднання операцій у транзакції. Кожна операція r закріплюється за певним вікном агрегації k .

Множина усіх операцій, що належать одному вікну агрегації R_k задається як:

$$R_k = \{ r \in O \mid t_r \in [kT, (k+1)T] \}. \quad (5)$$

Властивості множин R_k гарантують їх диз'юнктивність для різних k та повноту покриття усіх операцій з множини O .

Множина усіх операцій, що належать одному вікну агрегації R_k над певним об'єктом даних задається як:

$$R_k(d) = \{ r \in R_k \mid d_r = d \}. \quad (6)$$

Функція агрегації, що перетворює множину операцій над певним об'єктом даних з певного вікна агрегації $R_k(d)$ у результуючу транзакцію $C_k(d)$ може бути представлена як:

$$f(R_k(d)) = C_k(d). \quad (7)$$

Роботу результуючої функції можна описати набором правил. Серед усіх операцій, першими на результуючу функцію подаються операції з найвищим пріоритетом. Результати роботи такої результуючої функції першими подаються на подальшу обробку. Тому, такі операції мають пріоритет над іншими. Серед операцій з однаковим пріоритетом, у якості результату результуючої функції, обирається операція з $\tau_r = Delete$. Вона автоматично стає результуючою транзакцією $C_k(d)$. Якщо ж такої операції у множині $R_k(d)$ не існує, то результуюча операція визначається як:

$$C_k(d) = \{ arg \max_{r \in R_k(d)} t_r \mid \tau_r \neq Delete \}. \quad (8)$$

Оскільки запити впорядковані за часом, і результуюча транзакція є найновішою з усіх операцій, що стосуються певного об'єкту даних, то загальний порядок операцій буде збережений – таким чином забезпечується послідовна узгодженість даних.

Запропонована модель розподіленого транзакційного годинника описує ефективний механізм пришвидшення процесу узгодження даних за рахунок агрегації

запитів, встановлення пріоритетів для запитів, мінімізації надлишкових змін та конфліктів між запитами, зберігаючи логічну послідовність подій у системі.

Реалізація розподіленого транзакційного годинника

Впровадження розподіленого транзакційного годинника у високонавантажену розподілену інформаційну систему, замість нерозподіленого транзакційного годинника, перш за все, дозволить зменшити час та ресурси, що витрачаються на узгодження даних у розподіленій системі.

Окрім пришвидшення процесу узгодження даних, розподілений підхід до реалізації транзакційного годинника надає додаткову можливість горизонтального масштабування високонавантаженої інформаційної системи. Даний факт сприятиме, перш за все, збільшенню пропускну здатності системи. Також, такий підхід забезпечить високий рівень стійкості системи до відмов. Виникнення збоїв у роботі одного з вузлів транзакційного годинника критично не вплине на загальну роботу системи, адже єдина точка відмови у високонавантаженій розподіленій інформаційній системі була усунута, порівняно із системою, що використовує нерозподілений транзакційний годинник.

На відміну від реалізації нерозподіленого транзакційного годинника, який не мав можливості працювати у кількох вузлах одночасно, розподілений транзакційний годинник набув такої можливості завдяки ряду удосконалень у його математичній моделі та безпосередній реалізації. Перш за все, варто відмітити зміну способу комунікації елементів інформаційної системи із розподіленим транзакційним годинником. Якщо у випадку нерозподіленого транзакційного годинника комунікація відбувалась з використанням HTTP запитів, то у реалізації розподіленого транзакційного годинника для комунікації використовується платформа для багатопотокової обробки подій Apache Kafka.

Використання асинхронного підходу до обміну даними за допомогою Kafka забезпечує зменшення часу необхідного для передачі даних та узгодження даних у системі, а також забезпечує високу пропускну здатність, можливість масштабування та покращує надійність системи за рахунок реплікації даних. Бінарний протокол, на основі TCP, що використовується в Kafka є ефективнішим за HTTP протокол, який має більші накладні витрати на запис та зчитування заголовків запитів та відповідей, а також на створення з'єднання для передачі запитів та отримання відповідей. Kafka ж, у свою чергу, оптимізована для високошвидкісного потокового передавання даних, тому час доставки повідомлення може складати менше 1 мс [3].

Також, розподілений транзакційний годинник отримав цілий ряд покращень реалізації, порівняно з нерозподіленим транзакційним годинником. По-перше, було

удосконалено та оптимізовано формат даних, що використовується для передачі та зберігання даних у транзакційному годиннику. По-друге, було удосконалено спосіб обробки вхідних повідомлень транзакційного годинника: замість періодичного пулінгу (pooling) вхідних повідомлень використовується реактивний підхід, який оптимальніше використовує серверні ресурси та зменшує затримки на доставку вхідних повідомлень у функцію обробки повідомлень розподіленого транзакційного годинника. Також, варто відмітити, що розподіл операцій над конкретними об'єктами даних між вузлами транзакційного годинника відбувається без накладних витрат. Задля цього використовуються вбудовані можливості платформи Kafka. Завдяки цьому факту у системі, що використовує розподілений транзакційний годинник існує гарантія послідовної узгодженості, адже усі операції над певним об'єктом даних в рамках вікна агрегації обов'язково будуть оброблені одним і лише одним екземпляром розподіленого транзакційного годинника.

Експериментальні дослідження використання розподіленого транзакційного годинника для пришвидшення процесу узгодження даних

Задля демонстрації реальної можливості застосування розподіленого транзакційного годинника як частини високонавантаженої розподіленої інформаційної системи було спроектовано та реалізовано три варіанти розподіленої системи підтримки фінансових операцій. Перша варіація системи використовує стандартний для MongoDB механізм Replica Set, другий варіант системи використовує нерозподілений транзакційний годинник у ролі механізму, відповідального за узгодження даних, а третя система – розподілений транзакційний годинник. Архітектури описаних систем, схематично зображені на рис. 1.

Окрім демонстрації можливості застосування розподіленого транзакційного годинника важливо, також, підтвердити той факт, що його застосування дійсно пришвидшує процес узгодження даних у системі. Задля цього було проведено ряд експериментів, що доводять ефективність рішення використати розподілений транзакційний годинник у ролі механізму реплікації даних, порівняно з нерозподіленим транзакційним годинником та стандартним для MongoDB механізмом Replica Set.

За основу для проведення експериментів було взято реалізацію систем, сценарії та результати експериментів з попередніх досліджень використання нерозподіленого транзакційного годинника [2].

Перший експеримент має на меті показати позитивний ефект використання розподіленого транзакційного годинника на швидкодію системи за рахунок

пришвидження процесу узгодження даних. Для цього було спроектовано та проведено наступний експеримент: один користувач за дуже короткий проміжок часу виконує серію операцій над одним банківським рахунком — наприклад, поповнює його, знімає кошти чи ініціює перекази. Через це у інформаційній системі виникає стан гонки, тобто конкуренція за зміну одного й того самого об'єкту даних з різних джерел. У такі моменти для інформаційної системи надзвичайно важливо максимально швидко обробити усі запити і не втратити жодного оновлення. Результатом експерименту є час обробки усіх операцій над банківським рахунком, що визначається як час від моменту відправки усіх операцій до моменту оновлення фінального стану рахунку, із урахуванням усіх операцій на кожному екземплярі розподіленої бази даних, що використовується у системі.

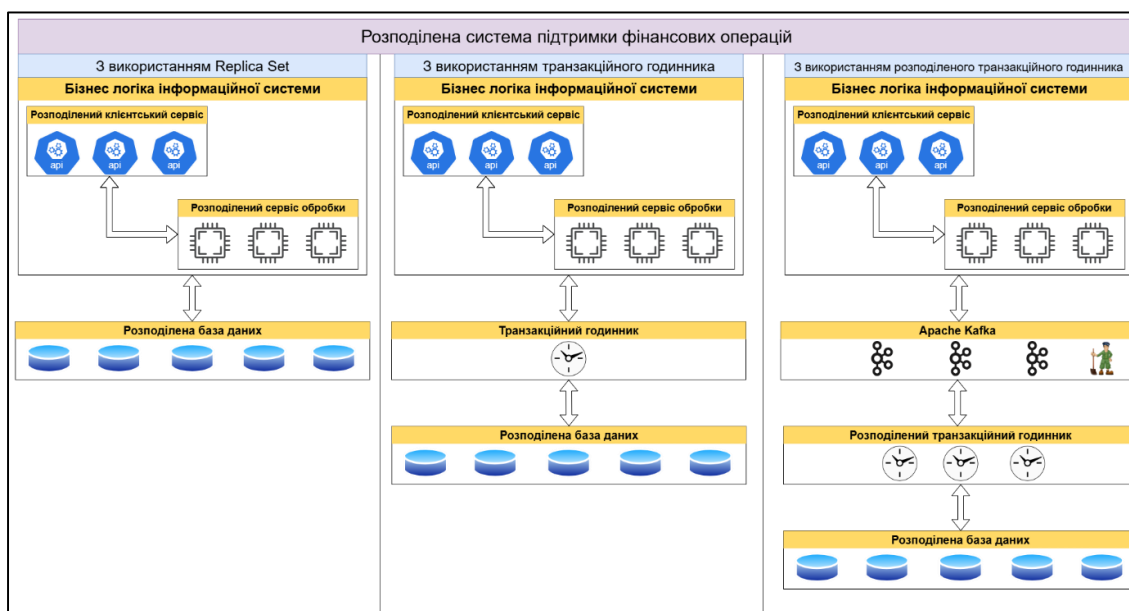


Рисунок 1. Архітектура варіацій системи підтримки фінансових операцій

Експеримент було повторено у системах, що використовують 3, 5 та 7 екземплярів бази даних. Також, було проведено ряд експериментів у системах з різною кількістю клієнтських сервісів та сервісів обробки, але через ідентичність отриманих результатів, надалі, вони будуть представлені разом. Кожен експеримент було повторено 100 разів, медіанне значення результатів вважається результатом експерименту. Задля кращої наочності результатів, у першому наборі експериментів розподілений транзакційний годинник працює у одному екземплярі. Результати першого експерименту у системі з використанням 3 екземплярів бази даних (БД) під керівництвом розподіленого, нерозподіленого транзакційних годинників та механізму Replica Set зображені на рис. 2.

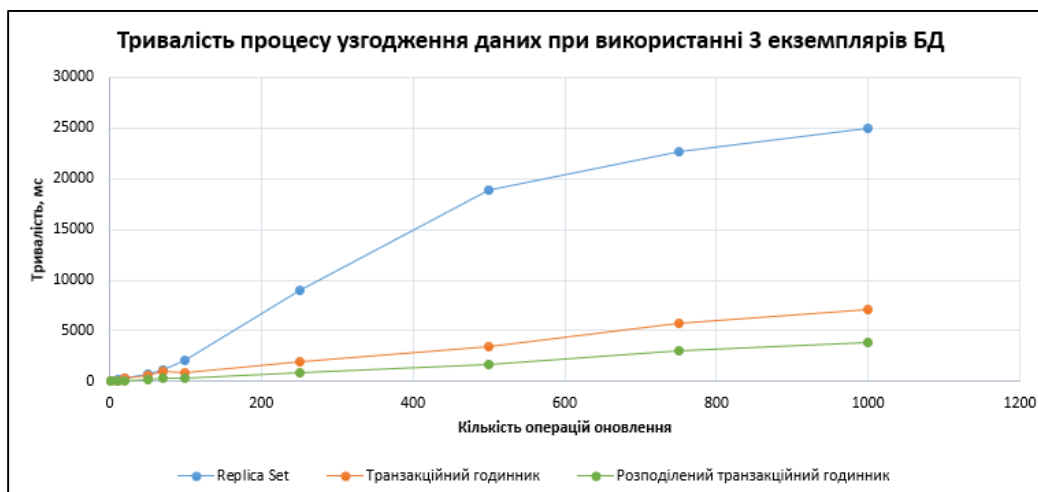


Рисунок 2. Тривалість процесу узгодження даних у системах з трьома екземплярами бази даних

Результати ідентичного експерименту, але з використанням 7 екземплярів бази даних зображено на рисунку 3.

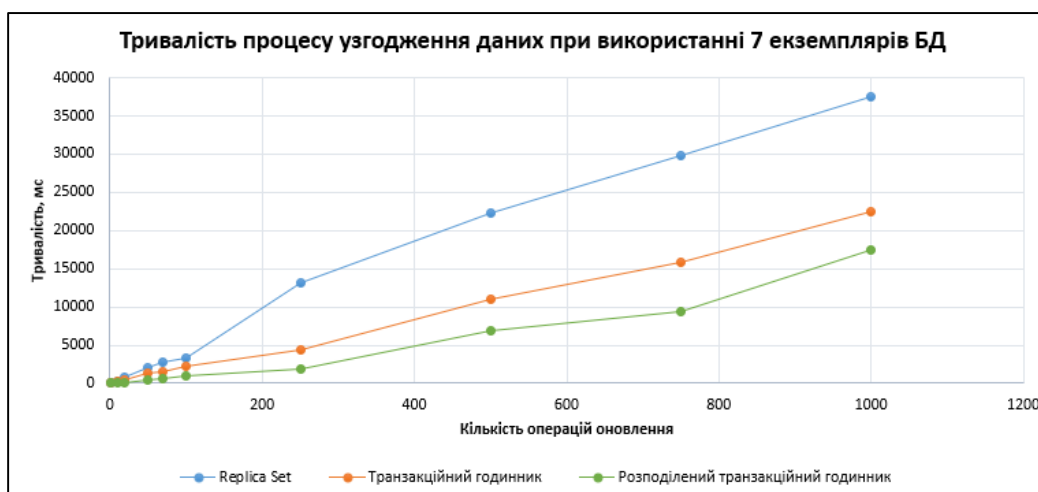


Рисунок 3. Тривалість процесу узгодження даних у системах із сімома екземплярами бази даних

Результати експериментів з використанням п'яти екземплярів бази даних у системі показали ідентичний результат, тому вони не будуть наведені окремо, але вони підпадають під загальний аналіз результатів.

Відповідно до результатів, використання розподіленого транзакційного годинника у високонавантажених інформаційних системах, наприкладі системи підтримки фінансових

операцій, гарантує пришвидшення процесу узгодження даних. Тобто, час необхідний на узгодження даних у системі, що використовує розподілений транзакційний годинник у ролі механізму реплікації даних, на, в середньому, 50% менше за час необхідний для узгодження даних у аналогічній системі, що використовує нерозподілений транзакційний годинник та на, в середньому 66% менше за систему, що використовує MongoDB Replica Set.

Суть наступного експерименту полягає у тому, щоб довести, що використання розподіленого транзакційного годинника підвищує швидкодію процесу узгодження критичних даних у високонавантажених інформаційних системах. Для демонстрації цього було проведено наступний дослід: у розробленій системі підтримки фінансових операцій деяка кількість користувачів одночасно відправляють запити на оновлення власних даних, реєстрації нового користувача та інших операцій, що не є критичними (пріоритетними) у системі. У цей же час, інший користувач проводить операцію депозиту на свій рахунок. Ця операція в рамках розробленої системи є пріоритетною (має вищий пріоритет відносно інших), тому результати обробки цієї операції мають бути поширені на усі екземпляри розподіленої бази даних з максимальною швидкістю. Експеримент проводився з різною кількістю некритичних запитів, але операція депозиту на рахунок завжди одна. У систему вона відправляється після того як було відправлено приблизно 60-70% запитів з операціями над некритичними даними. Операції над некритичними даними генеруються послідовно. Цей процес займає близько 3 секунд. Результатом експерименту вважається час обробки критичної операції над рахунком, що визначається як час від моменту відправки операції над рахунком у систему до моменту, коли кожен екземпляр бази даних, що існує у системі, отримає оновлену версію стану банківського рахунку користувача з урахуванням проведеної операції.

Результати експериментів у системах, що використовують 3 екземпляри розподіленої бази даних зображені на рис. 4.

З рис. 4 стає очевидно, що система, яка використовує Replica Set для узгодження критично важливих даних потребує часу у десятки, а то і тисячі разів більше. До того ж, необхідний для цього час постійно збільшується. Така тенденція прослідковується для усіх проведених експериментів, що стосуються узгодження критично важливих даних. Такі результати можна пояснити тим фактом, що у механізмі транзакційного годинника присутня можливість встановлення вищих пріоритетів для певних операцій. У той час як механізм Replica Set оброблює усі вхідні операції в порядку їх надходження. Тому, лінійна залежність між часом, необхідним на узгодження даних та кількістю операцій, що була описана у попередніх експериментах залишається актуальною і в цьому типі експериментів. Тому, у

подальших порівняннях та аналізі результатів, варто зосередитись на різниці у системах, що використовують розподілений та нерозподілений транзакційні годинники.

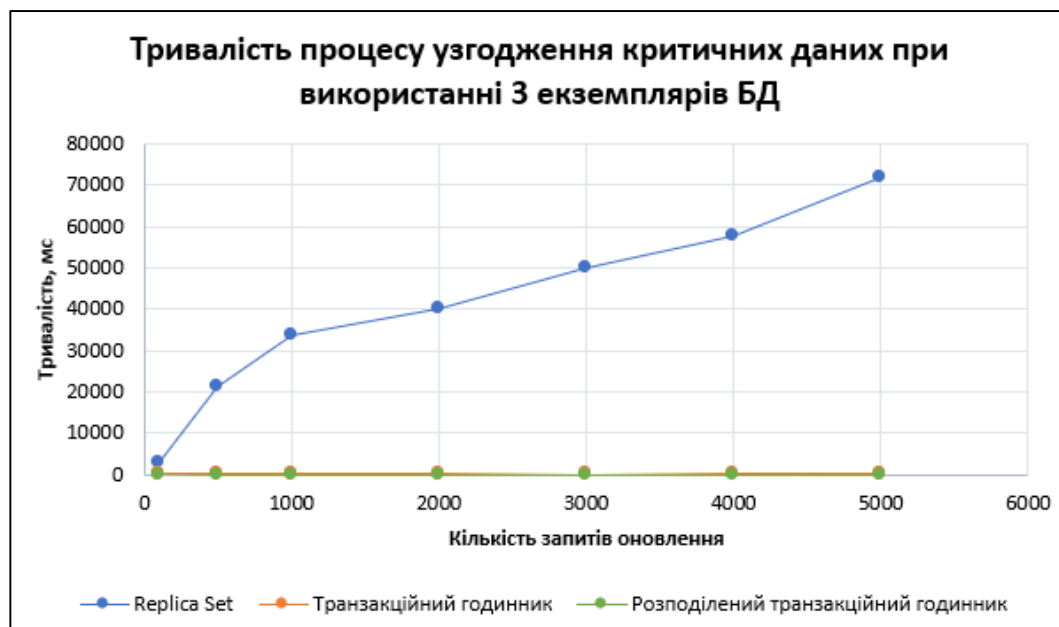


Рисунок 4. Тривалість процесу узгодження критичних даних у системі з використанням 3 екземплярів бази даних

Якщо ж з результатів експериментів прибрати результати систем з використанням Replica Set, то більш наочною стане різниця між системами, що використовують різні типи транзакційних годинників. Результати аналогічних експериментів у системах, що використовують різні типи транзакційних годинників та різну кількість екземплярів розподіленої бази даних зображені на рис. 5.

Проаналізувавши результати, можна стверджувати те, що використання розподіленого транзакційного годинника дійсно пришвидшує процес узгодження критично важливих даних у високонавантажених інформаційних системах порівняно з аналогічними системами, що використовують нерозподілений транзакційний годинник та механізм MongoDB Replica Set.

Так, перш за все, варто відзначити, що різниця у швидкодії процесу узгодження даних у інформаційних системах, що використовують розподілений транзакційний годинник та інформаційних системах, що використовують нерозподілений транзакційний годинник, становить, в середньому, 5 разів. По-друге, варто зауважити, що системи з розподіленим транзакційним годинником показують більш стабільні результати, що

робить роботу усієї системи більш стабільною та передбачуваною. Також, необхідно відмітити той факт, що приріст часу, необхідного для узгодження критично важливих даних у системі з використанням розподіленого транзакційного годинника, залежно від кількості вузлів розподіленої бази даних, що існують у системі, становить лише 10-20 мс, порівняно з 50 мс у системах з нерозподіленим транзакційним годинником.

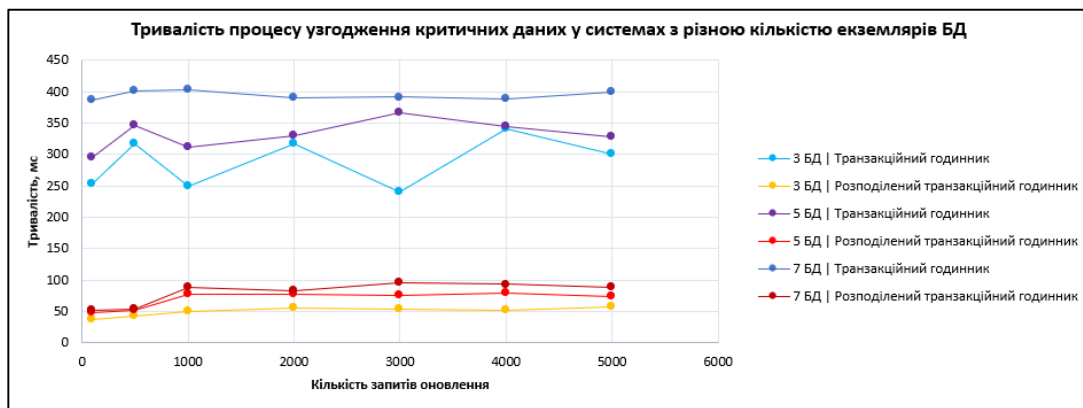


Рисунок 5. Порівняння тривалості процесу узгодження критичних даних у системах, що використовують різні типи транзакційного годинника та різну кількість екземплярів бази даних

Висновки

У даній статті було описано математичну модель та запропоновано концепцію впровадження розподіленого транзакційного годинника у розподілені високонавантажені інформаційні системи з метою пришвидшення процесу узгодження даних.

Запропонований підхід, окрім пришвидшення процесу узгодження даних, усуває єдину точку відмови у системі, підвищує її стійкість до відмов, рівень масштабованості та збільшує загальну пропускну здатність.

Результати експериментальних досліджень підтвердили ефективність розподіленого підходу до реалізації транзакційного годинника: використання розподіленого транзакційного годинника дозволяє зменшити час необхідний для узгодження даних, в середньому, на 50%, порівняно з часом, що необхідний для узгодження даних у аналогічній системі, що використовує нерозподілений транзакційний годинник та, в середньому, на 66% за систему, що використовує MongoDB Replica Set.

Помітна, також, і різниця у швидкодії процесу узгодження критичних даних у інформаційних системах, що використовують розподілений транзакційний годинник та інформаційних системах, що використовують нерозподілений транзакційний годинник,

яка становить, в середньому, 5 разів. Інформаційні системи, що використовують механізм MongoDB Replica Set, вимагають набагато більше часу для узгодження критичних даних. Швидкодія процесу узгодження критичних даних у таких системах може поступатись швидкодії аналогічних процесів у системах з розподіленим транзакційним годинником у сотні та, навіть, тисячі разів.

Список використаних джерел

1. Internet usage worldwide – statistics & facts. *Statista*. URL: <https://www.statista.com/topics/1145/internet-usage-worldwide> (дата звернення: 18.02.2025).
2. Муліш В., Крилов Є., Нікітін В., Анікін В. Використання транзакційного годинника у розподіленій системі підтримки фінансових операцій задля пришвидшення процесу узгодження даних. *Адаптивні системи автоматичного управління*. 2024. № 45. С. 26–33. URL: <https://doi.org/10.20535/1560-8956.45.2024.313064> (дата звернення: 20.02.2025).
3. Benchmarking Apache Kafka: 2 Million Writes Per Second (On Three Cheap Machines). *LinkedIn Engineering*. URL: <https://engineering.linkedin.com/kafka/benchmarking-apache-kafka-2-million-writes-second-three-cheap-machines> (дата звернення: 15.04.2025).