DOI 10.34185/1562–9945–5–166–2026–06
УДК 004.65

О.В. Резніченко, О.А. Ляшенко, В.В. Архипова

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ЗАТРИМОК РЕПЛІКАЦІЇ У РОЗПОДІЛЕНІЙ БАЗІ ДАНИХ MONGODB

Анотація. Представлена робота присвячена актуальній задачі визначення впливу факторів та побудови моделі затримок реплікації у розподіленій базі даних MongoDB в умовах інтенсивного навантаження із використанням методу активного зондування. Його застосування дозволило подолати обмеження дискретності стандартних засобів спостереження, а саме секундного інтервалу збору метрик, та забезпечити отримання точних значень затримок реплікації за рахунок запису даних із точними часовими мітками на первинному вузлі та їх зчитування із вторинних вузлів.

На основі отриманих експериментальних даних щодо впливу рівня гарантії запису, типу операції, завантаження процесора, часу очікування у чергах, кількості паралельних клієнтських з'єднань, кількості документів у пакеті та часового інтервалу між операціями запису, використовуючи алгоритми градієнтного бустингу, проведено оцінку значущості факторів. Аналіз показав, що вирішальний вплив на виникнення та зростання затримок реплікації мають кількість паралельних клієнтських з'єднань, обсяг документів у пакеті та часовий інтервал між операціями запису. Також підтверджено, що час очікування у чергах та завантаження центрального процесора не несуть значного впливу на процес утворення затримок.

За результатами дослідження побудовано нелінійну математичну модель роботи кластера (коефіцієнт детермінації $R^2 = 0,9786$), що може бути використана під час проектування, оптимізації та конфігурування розподілених NoSQL–систем для прогнозування параметрів реплікації та вибору оптимальних параметрів роботи систем.

Ключові слова: градієнтний бустинг, прогнозування, оцінка значущості факторів, апроксимація, коефіцієнт детермінації, машинне навчання, екологічний моніторинг, математичне моделювання, метод активного зондування, синтетичний моніторинг.

Постановка проблеми. Системи екологічного моніторингу завдяки глобальному розповсюдженню на земній кулі та інтенсивному збору даних генерують велику кількість статистичної інформації [1–4].

Ці дані накопичуються у розподілених базах даних, а отже збереження їх актуальності є критичним питанням. Його розв'язання визначатиме ефективність роботи систем оповіщення та прийняття рішень [1] у разі різких змін концентрацій забруднюючих та токсичних речовин.

Серед сховищ даних, що часто використовуються для збереження інформації про довкілля, виступають нереляційні бази даних MongoDB завдяки простоті розгортання

© Резніченко О.В., Ляшенко О.А., Архипова В.В., 2026
Copyright for this paper by its authors. Use permitted under License CC BY 4.0.
ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

та роботи, легкості масштабування, а також можливості одночасного збереження інформації різних типів [5–7].

Основним способом забезпечення дублювання інформації на різних вузлах є реплікація між первинним і вторинними вузлами бази даних, а збереження узгодженості даних можливе при мінімізації затримок реплікації. Таким чином, знання закономірностей процесів реплікації даних виступає важливим фактором підтримання затримок на мінімальному рівні [8, 9].

Визначення затримок реплікації та їх характеристик також має свої проблеми, що пов'язані з обмеженнями обладнання. Так, вбудований засіб збирання метрик [10] MongoDB, визначаючи затримку [11] як різницю між часом операції на первинному вузлі і часом застосованої операції на вторинному, оперує дискретними показниками, які не дозволяють оцінити реальні значення затримок реплікації.

Аналіз останніх досліджень і публікацій. У дослідженнях широко використовуються альтернативні підходи, що не застосовують збирання системних характеристик вбудованими засобами [12–22].

Підходи, основані на активному моніторингу, часто використовують для діагностики у випадках, коли відсутня можливість вимірювання показників стандартними засобами або отримані дані не є достовірними (наприклад, при дослідженні електричних [14] та комп'ютерних мереж [15–18], у системах кібербезпеки [19] тощо). На відміну від пасивного моніторингу (passive monitoring), при якому метрики зчитуються за допомогою системних інструментів або датчиків, активне зондування (active probing) генерує синтетичний потік даних з його подальшою обробкою системою, отже здійснюється активний вплив на неї.

У дослідженні [12] авторами проведено порівняльний аналіз пасивного (passive monitoring) та активного (active probing) підходів до моніторингу комп'ютерних мереж і доведено переваги останнього для випадків, коли пасивного спостереження недостатньо. Застосування синтетичного моніторингу (synthetic monitoring) дозволяє виконувати безперервний контроль продуктивності та вчасно виявляти затримки та збої [13].

Авторами [18] запропоновано метод виявлення проблем із надійністю та швидкістю вебсервісів (загальних, часткових, періодичних затримок) на основі синтетичного моніторингу та математичного аналізу отриманих трасувань. Розроблено модель типових причин зниження продуктивності сервера і перевірено її як у експериментальних умовах, так і в роботі реального вебсайту у фінансовому секторі.

Дослідження [20] пропонує огляд стратегій проактивного моніторингу високонадійних баз даних, що поєднує відстеження на рівні сервісів, телеметрію, прогнозну аналітику та автоматизоване керування для запобігання інцидентам. Впровадження таких інструментів дозволило зменшити кількість незапланованих перемикань, стабілізувати продуктивність та мінімізувати кількість хибних сповіщень.

У праці [21] запропоновано розширення традиційних тригерів для баз даних, що дозволяє періодично перевіряти складні зміни, описані SQL–запитами, замість

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

примітивних подій. Впровадження цього підходу базується на ефективному інкрементному оновленні представлень для виявлення подій.

У дослідженні [22] розглядається синтетичний моніторинг як проактивний підхід до оцінки продуктивності, доступності та функціональності застосунків. Інтеграція синтетичного моніторингу в конвеєри CI/CD дозволяє підтримувати безперервний цикл розробки, підвищувати надійність, та виявляти дефекти до того, як вони вплинуть на користувачів.

Проаналізовані літературні джерела свідчать про універсальність підходів, основаних на активному впливі на досліджувану систему (активне зондування), та підтверджують можливість їх використання для аналізу характеристик реплікації MongoDB.

Мета дослідження. Метою дослідження є визначення фактичних затримок реплікації у розподіленому кластері MongoDB та закономірностей їх виникнення, а також побудова математичної моделі для оцінки й прогнозування впливу основних факторів на характеристики процесу реплікації.

Викладення основного матеріалу дослідження. Для організації експериментів використано локальний кластер бази даних MongoDB (Intel Core i5–8250U 1,6 ГГц; DDR3 16 ГБ, SSD 120 ГБ), що складається з трьох вузлів: первинного (Primary) та двох вторинних (Secondary) (порти 27020, 27021, 27022) [23]. Така конфігурація моделює зберігання даних моніторингу на окремих вузлах з обмеженими ресурсами; також в цьому дослідженні повністю виключено випадкові мережеві затримки та їх вплив на процес реплікації.

Для створення навантаження на кластер та запису результатів про затримки реплікації, обсяг брудного кешу WiredTiger, стан внутрішніх черг блокувань та системних операцій розроблено Python–скрипт.

Для подолання системних обмежень стандартного моніторингу MongoDB (а саме дискретного інтервалу збору метрик у 1 с) та оцінки реальної затримки реплікації використано метод активного зондування (active probing), або синтетичного моніторингу (synthetic monitoring). Вимірювання затримок реалізується таким чином:

– на Primary–вузлі формується спеціальний документ, час створення якого визначено і записано у вигляді часової мітки T_1;

– документ записується до журналу операцій oplog первинного вузла, потім копіюється на Secondary;

– відбувається зчитування цього документа з Secondary–вузла (режим переваги вторинного вузла Secondary Preffered);

– затримка реплікації ΔT обчислюється як різниця між часом зчитування T_2 та часовою міткою (1):

$$\Delta T = T_2 - T_1 \quad (1)$$

Для розподілених баз даних одним з параметрів, що характеризує їх роботу, є рівень гарантії запису (Write Concern) [24]:

– w:1, при якому запис відбувається на первинний (Primary) вузол, і, не чекаючи запису на вторинному (Secondary), робиться підтвердження щодо успішності операції;

– w: «majority», при якому запис робиться на більшість вузлів (для даного випадку – на 2 з 3), і тоді вважається успішним;

– w: all та інші комбіновані види, що передбачають підтвердження всіма вузлами, або використовують для одних операцій один тип рівня гарантії запису, а для інших – другий.

Експериментальний розподіл затримок реплікації для w:1 та w: «majority» представлено на рис. 1, параметри отриманих розподілів – у табл. 1.

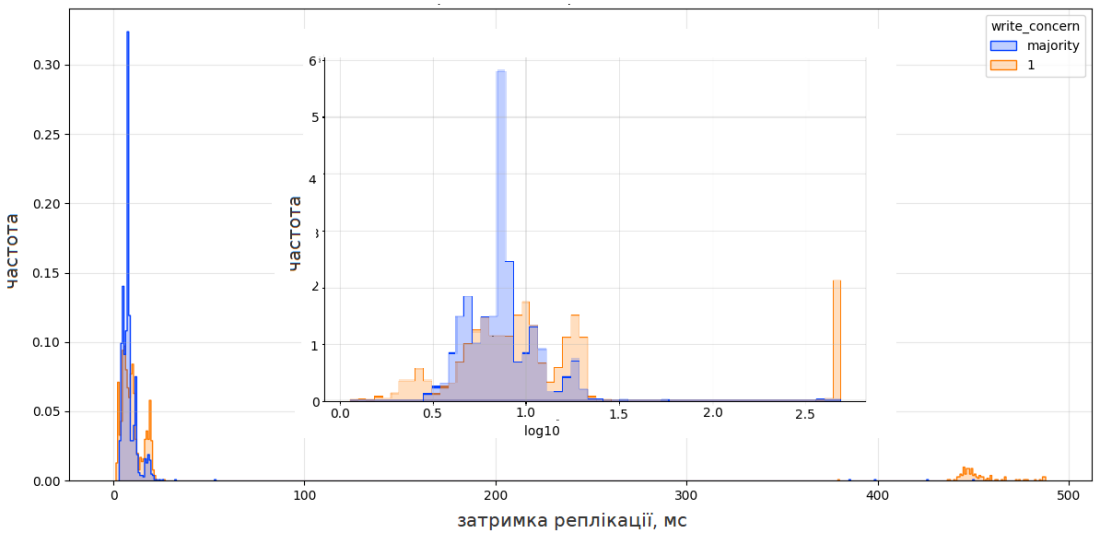


Рисунок 1 – Вплив рівня гарантії запису (Write Concern) на розподіл затримок реплікації (мс) для w:1 та w: «majority» (лінійний та логарифмічний вигляд)

Таблиця 1

Статистичні характеристики розподілу затримок реплікації (мс) в залежності від рівня гарантії запису, мс

Метрика	w:1	w: «majority»
Середнє значення (mean)	43,31	8,79
Стандартне відхилення (std)	122,37	16,62
Мінімальне значення (min)	1,05	3,22
Перцентілі P50	6,78	7,08
P95	450,89	17,26
P99	481,50	21,87
Максимальне значення (max)	491,46	385,80

Відповідно до отриманих результатів, для обох досліджених рівнів гарантії запису підтверджено бімодальний розподіл затримок реплікації, що більш виражений для режиму w: 1.

Основний масив значень затримок міститься в області низьких значень (0–50 мс). Але для w: 1 на діаграмі також присутня зона високих затримок у діапазоні 400–500 мс.

Згідно з даними таблиці 1, при використанні концепції гарантії запису w: 1 реплікація є статистично більш повільною: середнє значення затримки становить 43,31

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

проти 8,79 мс для w: «majority», ще більша різниця спостерігається на високих перцентилях – наприклад, при P_{99} затримка становить 481,5 мс для w:1 проти 21,87 мс для w: «majority».

Ці закономірності пов'язані із особливостями взаємодії вузлів, що розташовані на одному комп'ютері.

При w: 1 первинний **(**Primary**)** вузол одразу підтверджує запис і починає наступні операції. Це призводить до заповнення оперативної пам'яті та накопичення «брудного кешу» (Dirty Cache) двигуна WiredTiger в межах 8,86–11,01 Мб. І коли кеш переповнюється, ініціюється процес скидання даних на диск під час контрольної точки (Checkpointing), що призводить до тимчасового припинення операцій запису журналу (Journaling) та блокування роботи фізичних дисків [25, 26].

Наслідком цього є утворення 105 аномальних викидів (9% всіх вимірювань) для w: 1 з затримками 440–480 мс. Причому для цього діапазону визначено, що у самої MongoDB немає внутрішніх черг або блокування (*{'total': 0, 'readers': 0, 'writers': 0}*).

Для w: «majority» первинний вузол очікує підтвердження запису даних від вторинного. Цей часовий інтервал дає можливість скидати «брудний кеш» без значної кількості зупинок, хоча в окремі моменти часу значення можуть досягати 15,88 Мб. Кількість аномальних викидів в цьому випадку становить 3 (0,25 %).

Таким чином, використання режиму w: «majority» є оптимальним рішенням з точки зору забезпечення надійності збереження інформації у базі даних і зменшення критичних затримок реплікації.

Використання w: 1, незважаючи на швидке підтвердження запису інформації на первинний диск, може призводити до розсинхронізації даних на вторинних вузлах та виникнення блокувань.

Для режиму w: «majority» було проведено експеримент щодо визначення впливу типу операції (Operation Type) (insert, find, update, delete) на затримки реплікації (рис. 2, табл. 2).

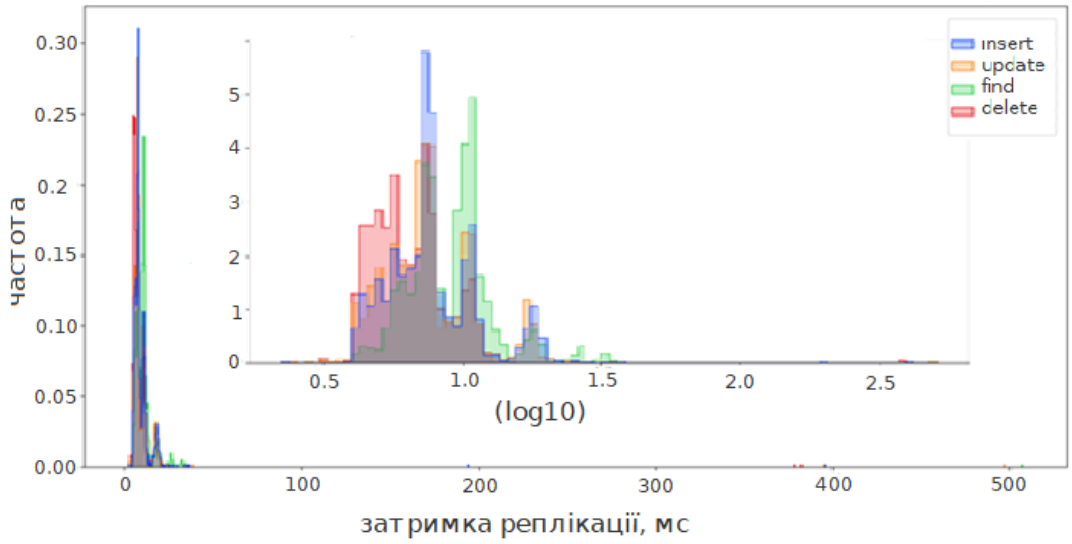


Рисунок 2 – Вплив типу виконуваних операцій (insert, find, update, delete) на розподіл затримок реплікації (лінійний та логарифмічний вид)

Порівнюючи вищезазначені типи операцій, слід відмітити, що отримані розподіли мають схожий характер.

Таблиця 2

Статистичні характеристики розподілу затримок реплікації (мс)
в залежності від типу операції, мс

Тип операції	Delete	Find	Insert	Update
Середнє (Mean)	7,38	10,16	8,85	8,86
Перцентилі:				
P50	6,27	9,32	7,45	7,19
P95	10,93	17,95	17,7	17,1
Максимальне значення (max)	382,35	507,54	395,69	497,15
Dirty Cache, середнє значення, Мб	10,5	8	3,5	5,5

Більша частина затримок є незначною (медіана становить 6–9 мс), що свідчить про стабільність роботи системи. Оскільки операції відрізняються за складністю, можна було б очікувати, що дані по затримках також будуть різнитися, але відсутність значних розбіжностей є непрямим свідченням про відсутність впливу з боку центрального процесору (CPU).

Кількість аномальних викидів є незначною, величина значних затримок, як і в попередньому випадку, складає до 0,5 с, але вони виникають навіть для операції find, яка, на відміну від інших, не створює нових записів і не змінює дані. Отже, це підтверджує гіпотезу, що основною причиною є блокування самої операційної системи під час скидання сторінок пам’яті (Checkpointing), коли системний механізм контролю стану вузлів (Heartbeat) стає у загальну чергу з фоновими процесами MongoDB.

Аналізуючи накопичення «брудного кеша» (Dirty Cache), можна зробити висновок, що найбільше його створюється при виконанні операції delete (10–11 Мб). Але утворення сплесків затримок спостерігається серед усіх типів операцій (380–500 мс).

Для визначення впливу завантаження процесора та пам’яті (CPU and RAM load) було проведено тестування системи шляхом вимірювання цих параметрів і відповідних їм затримок реплікації кожні 0,5 с для режиму w=majority (табл. 3).

Для оцінки зв’язку між зазначеними параметрами було визначено коефіцієнт кореляції відповідно до рівняння (2) [27] :

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2 \sum_{i=1}^{n}(y_i-\bar{y})^2}, \quad (2)$$

де x_i – значення завантаження CPU, RAM в момент часу i;

y_i – затримка реплікації між первинним та вторинним вузлами у момент часу *i;*

$\bar{x}, \bar{y}$ – середні вибіркові значення відповідних параметрів.

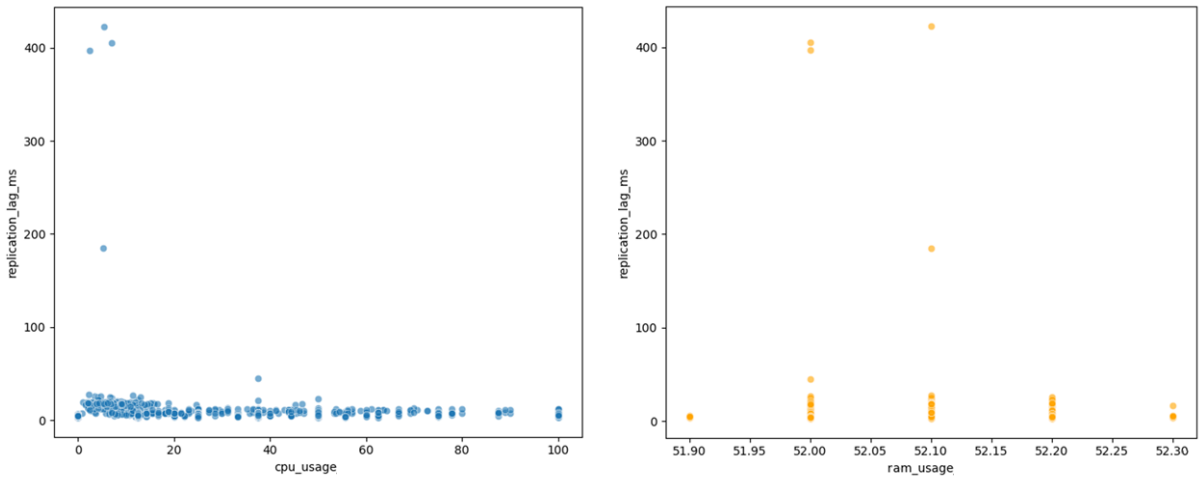
ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Таблиця 3

Коефіцієнти кореляції між затримкою кореляції та системними параметрами (при w=majority)

Параметр	Коефіцієнт кореляції r
Використання RAM, %	0,01727
Використання CPU, %	–0,003539

Також залежності між затримками реплікації та відповідними системними параметрами показані на рис.3.



а б

Рисунок 3 – Вплив CPU (cpu_usage) (а) та RAM (ram_usage) (б) на затримки реплікації (replication_lag_ms)

Отримані дані табл. 3, а також залежності, показані на рис. 3, свідчать про те, що ресурси комп'ютера не мають кореляції з затримками реплікації, а отже, не несуть на неї значного впливу.

Експеримент щодо часу очікування у чергах (Queue Wait) імітує ситуацію, коли дані потрапляють у систему безперервно, але вони повинні очікувати у внутрішньосистемній черзі (наприклад, через раптове коротке блокування ресурсів за рахунок скидання даних з пам'яті WiredTiger або збільшення кількості інших потоків операцій). Дослідження проводилося шляхом внесення додаткових затримок між циклами виконання операцій тривалістю 5, 100, 1000 та 2000 мс. Аналізуючи отриманий розподіл (рис. 4, табл. 4), можна відмітити, що зміна часу очікування не впливає на характер розподілу: пік розподілу приходиться на діапазон 5–10 мс, медіанні значення майже співпадають (6,23–6,86 мс), робота системи є стабільною за виняткам викидів 2157,84 мс при затримці 2000 мс.

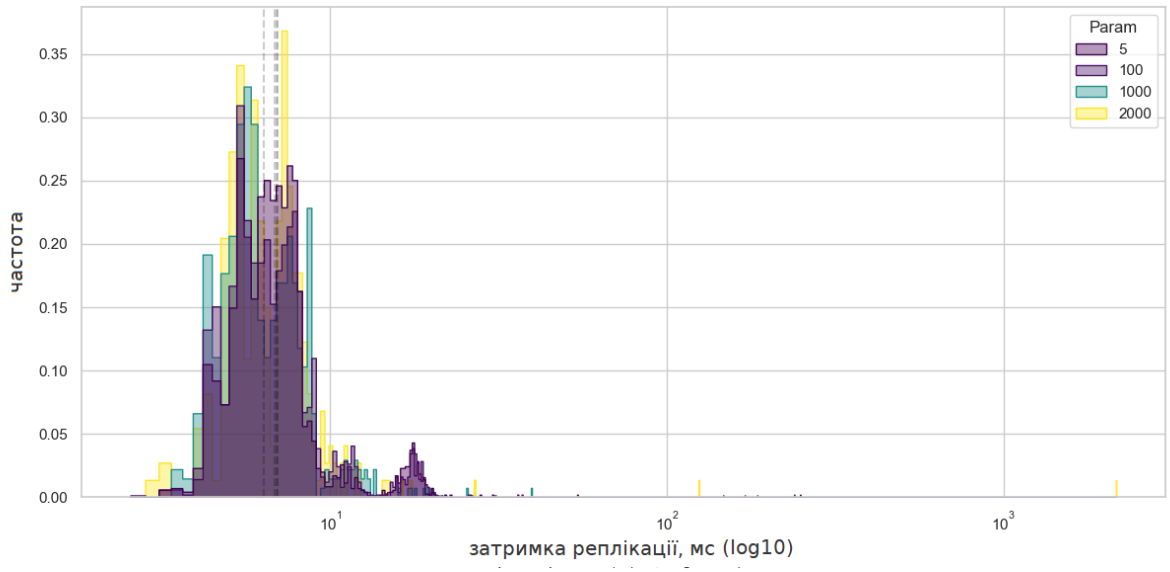


Рисунок 4 – Вплив часу очікування у чергах на розподіл затримок реплікації в залежності від часу очікування у чергах (5, 100, 1000, 2000 мс) (логарифмічна шкала)

Таблиця 4

Статистичні характеристики розподілу затримок реплікації (мс) в залежності від часу очікування у чергах, мс

Час очікування	**5**	**100**	**1000**	**2000**
Середнє (Mean)	7,87	8,12	6,93	15,32
Перцентилі:				
P50	6,81	6,86	6,23	6,75
P95	17,24	17,53	11,69	10,97
Максимальне значення (max)	185,52	248,76	39,39	2157,84

Визначення впливу кількості відкритих клієнтських з’єднань (Connections), що одночасно приєдналися до бази даних, досліджено для 10, 100, 500 та 1000 паралельних потоків (при w=majority). Отримані результати демонструють (рис. 5, табл. 5) сильну залежність затримок реплікації від цього параметра. При невеликій кількості з’єднань (10) робота системи стабільна, затримки мають куполоподібний вузький розподіл (середнє значення 6,29 мс, P_{50}=5,96 мс). При збільшенні кількості з’єднань до 100 утворюється довгий «хвіст» затримок (P_{95}=88,01 мс). При зростанні навантаження до 500–1000 з’єднань реплікація десинхронізується, середня затримка різко збільшується до 639,09 та 6333,57 мс, P_{95} досягає 10139,61 мс. Це пояснюється виникненням конкуренції за ресурси, і система значний час проводить в очікуваннях.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

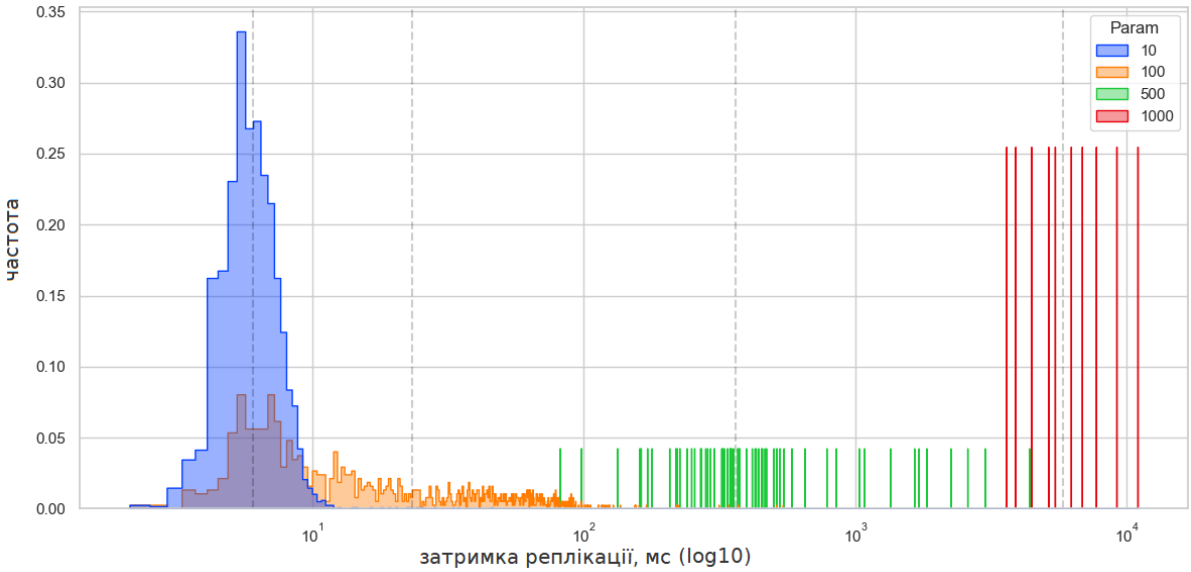


Рисунок 5 – Вплив кількості паралельних клієнтських з'єднань (10, 100, 500, 1000) на затримки реплікації (логарифмічна шкала)

Таблиця 5

Статистичні характеристики розподілу затримок реплікації (мс) в залежності від кількості паралельних клієнтських з'єднань

Кількість з'єднань	**10**	**100**	**500**	**1000**
Середнє (Mean)	6,29	35,5	639,09	6333,57
Перцентилі:				
P50	5,96	23,18	361,34	5825,04
P95	8,67	88,01	2257,94	10139,61
Максимальне значення (max)	135,72	539,94	4362,54	10947,02

Вплив кількості документів у пакеті (Batch Size) досліджено для конфігурацій 10, 50, 100, 500 документів (періодичність відправлення 1 с) (рис. 6, табл. 6).

Аналіз результатів вказує на наявність зворотної залежності між розміром пакету та затримкою реплікації. Так, при невеликій кількості документів у пакеті (10) основна частина затримок зміщується в область більших значень (середнє значення становить 102,15 мс, стандартне відхилення – 277,62 мс), а крім того, спостерігається значний сплеск в районі 1000 мс ($\log_{10}$=3).

При збільшенні пакету одночасно оброблюваних документів до 500 розподіл стає більш вузьким з піком щільності 5–10 мс (середнє значення затримки знижується до 5,32 мс, стандартне відхилення – 1,89 мс).

Це пов'язується із тим, що при відправленні пакета документів існують постійні накладні витрати (запити статусу, записи heartbeat, синхронізація журналів тощо), а у великих пакетах ці витрати зменшуються з розрахунку на один документ, оскільки розподіляються на весь пакет документів. Крім того, вторинні вузли більш ефективно і швидко використовують операції з oplog, коли документи надходять групами.

Слід відмітити, що при меншому навантаженні (10 документів у пакеті) «дзвін» розподілу зміщений правіше (середнє значення mean для пакета з 10 документів складає 102,15 мс), ніж при більших навантаженнях (500 – 5,32 мс).

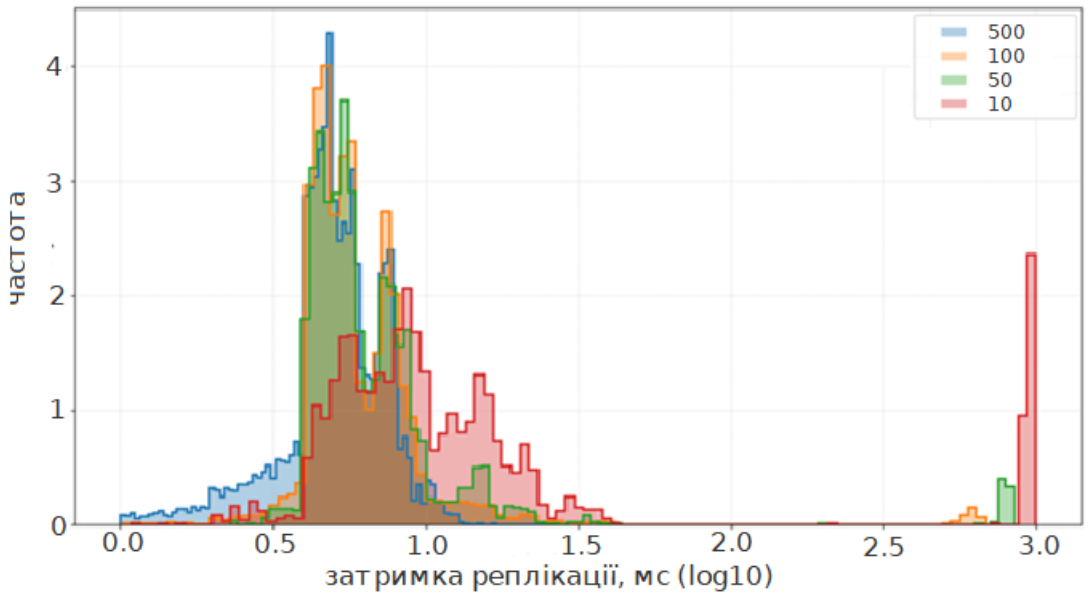


Рисунок 6 – Вплив розміру документів у пакеті (Batch Size)
на розподіл затримок реплікації (логарифмічна шкала)

Таблиця 6

Статистичні характеристики розподілу затримок реплікації
при зміні кількості документів у пакеті

Розмір пакету (Batch)	**Середнє (Mean), мс**	**Медіана (Median), мс**	**Стандартне відхилення (Std), мс**	**Мініма–льне зна-чення (min), мс**	**Максима–льне зна-чення (max), мс**	**Коефі–цієнт варіації (cv), %**
10	102,15	9,00	277,62	1,08	995,33	271,78
50	22,53	5,67	109,46	2,29	840,69	485,88
100	12,31	5,5	59,48	1,13	656,85	484,14
500	5,32	5,02	1,89	1,02	17,24	35,56

Це може пов’язуватися з декількома факторами:

– при одночасному передаванні великого пакету документів відправлення відбувається безперервним потоком, і для кожного циклу виконуються підготовчі операції (пов’язані з запитом статусу, записом «маркера» тощо), тому у випадку передавання невеликого пакету ці затримки будуть грати значну роль у загальному часі операції;

– при передаванні великої кількості даних кеш системи використовується весь час, для невеликої кількості – переходить в енергозберігаючий стан.

Часовий інтервал між операціями запису (Interval) виникає у зв’язку з тим, що інформація не може потрапляти у систему безперервно, а, як у випадку з моніторингом

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

довкілля потрапляє пакетами з інтервалами. Для визначення впливу даного параметру до бази даних на затримки реплікації було проведено серію досліджень з періодичністю збирання даних 0, 1, 5 та 10 с з пакетом документів 20 шт (рис. 7).

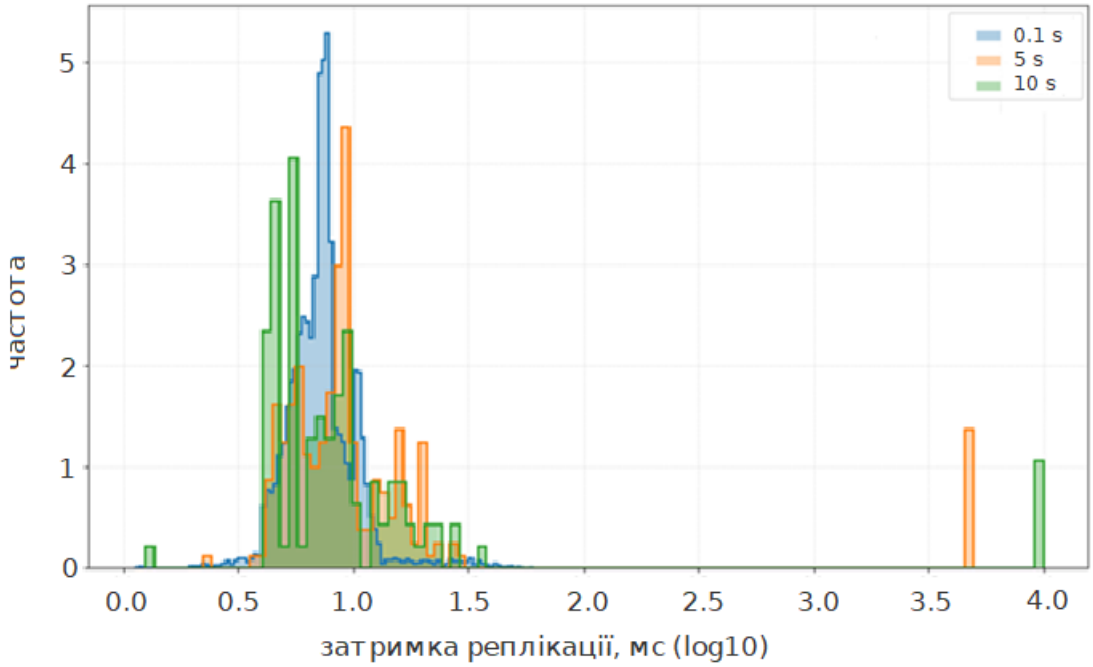


Рисунок 7 – Вплив часового інтервалу між операціями запису (Interval) на розподіл затримок реплікації (логарифмічна шкала)

При визначеннях затримок реплікації через різні періоди часу (через 0,1, 5 та 10 с з пакетом документів 20) слід відмітити наступне (табл. 7, рис. 7).

При вимірюваннях через 0,1 с майже всі точки потрапляють у досить вузький діапазон затримок 5–10 мс з середнім значенням 8,92 мс і медіаною 7,16 мс. Максимальна затримка складає 54,07 мс, що свідчить про стабільну роботу системи.

Під час експериментів, при яких збір даних відбувається через 5–10 с, крім затримок у діапазоні близько 7–9 мс (медіана 7,36 та 8,55 мс), присутні також екстремальні лаги (4902,05 мс та 9911,69 мс).

Таблиця 7

Статистичні характеристики розподілу затримок реплікації в залежності від інтервалу між операціями запису

Періодичність, с	**Середнє (Mean), мс**	**Медіана (Median) мс**	**Стандартне відхилення (Std), мс**	**Мінімальне значення (min), мс**	**Максимальне значення (max), мс**	**Коефіцієнт варіації (cv), %**
0,1	8,92	7,16	7,37	3,99	54,07	82,63
5	131,38	8,55	764,87	4,11	4902,05	582,18
10	420,84	7,36	1984,19	1,24	9911,69	471,48

Це може пояснюватися особливостями процесу вимірювання при тривалих паузах – вторинний вузол переходить у режим очікування, і затримка визначається відносно попереднього маркера вимірювання, що призводить до утворення аномальних значень.

Для оцінки впливу отриманих експериментальних параметрів на затримки реплікації було використано алгоритм градієнтного бустингу (Gradient Boosting Regressor), що врахував наступні параметри: кількість клієнтських з'єднань (Connections), кількість документів у пакеті (Batch), часовий інтервал між операціями запису (Interval), завантаження процесора (cpu), час очікування у чергах (queue wait), і на його основі створено графік значущості ознак (Feature Importance) (рис. 8) та табл. 8.

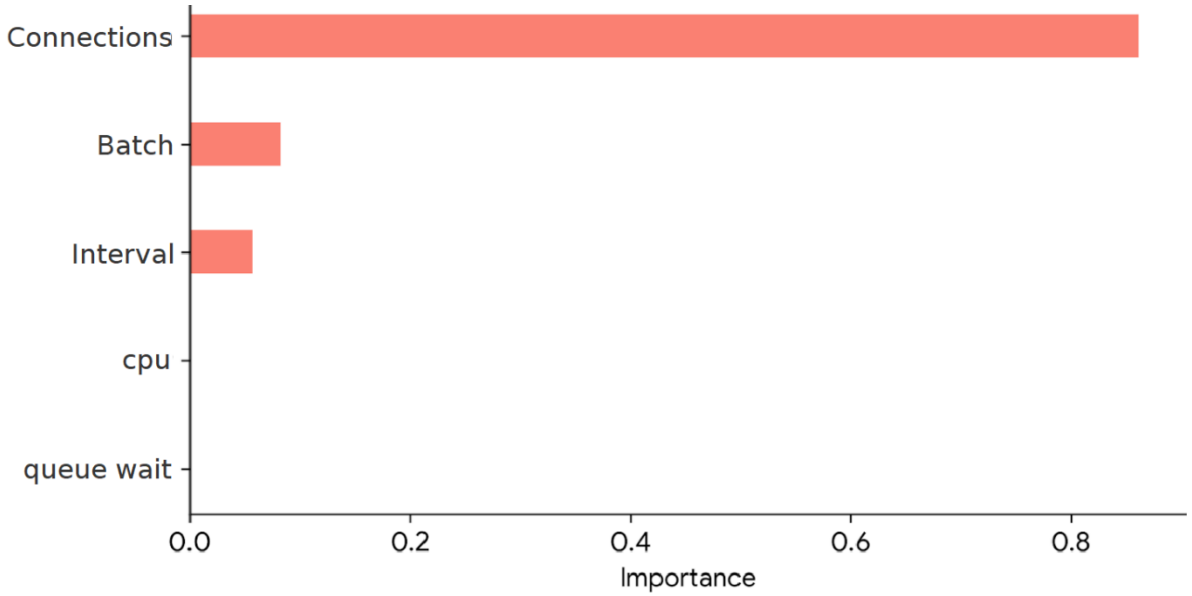


Рисунок 8 – Оцінка значущості впливу параметрів на затримки реплікації

Аналіз рис. 8 і табл. 8 показав, що серед значущих параметрів найбільший вплив на затримку реплікації мають: кількість з'єднань, інтервал звертання до бази даних та розмір документів у пакеті.

Таблиця 8

Оцінка значущості ознак (Feature Importance)

Фактор / Позначення	**Значущість (Importance)**
Кількість з'єднань / Connections	0,54 (54%)
Інтервал звернення / Interval	0,28 (28%)
Розмір пакета / Batch	0,18 (18%)
Час очікування у чергах (queue wait)	0,001 (0,1%)
Завантаження процесора / cpu	0,001 (0,1%)

Інші параметри за впливом складають менше 1 %, тому у подальшому аналізі вони не враховуються.

Подальшим кроком було побудування математичної моделі, яка б описувала сумарний вплив значущих параметрів на затримки реплікації. Для цього було взято всі

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

файли *.csv зі статистичними даними щодо кількості паралельних клієнтських з'єднань (C), часового інтервалу між операціями запису (I) та кількості документів у пакеті (B).

Для апроксимації було обрано декілька видів функцій та описано ними експериментальні дані (табл. 9).

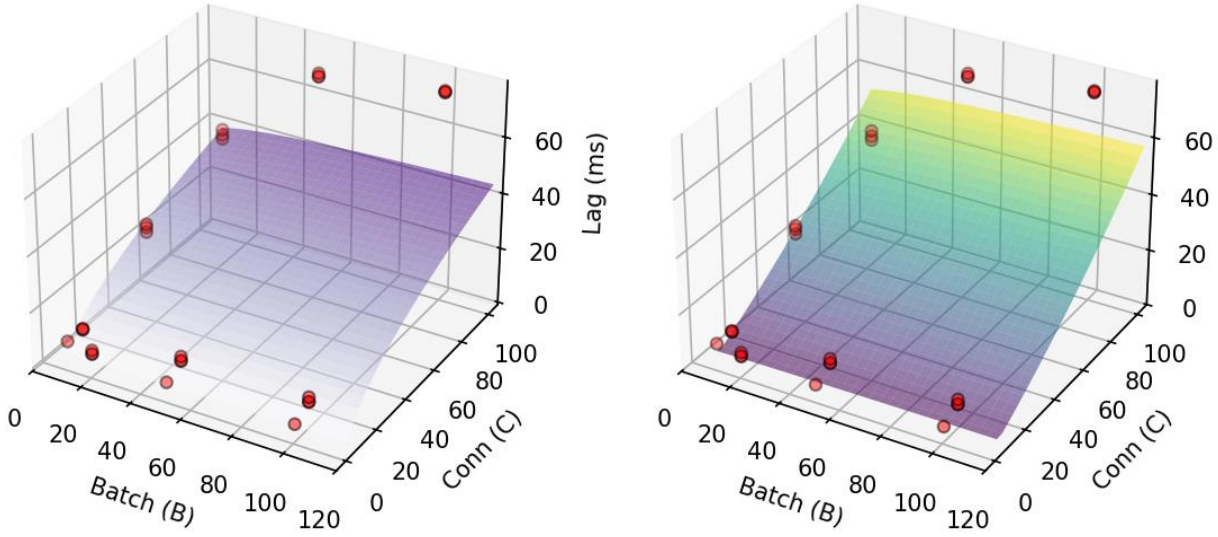
Таблиця 9

Функції для визначення затримки реплікації (lag)

та відповідні їм коефіцієнти детермінації

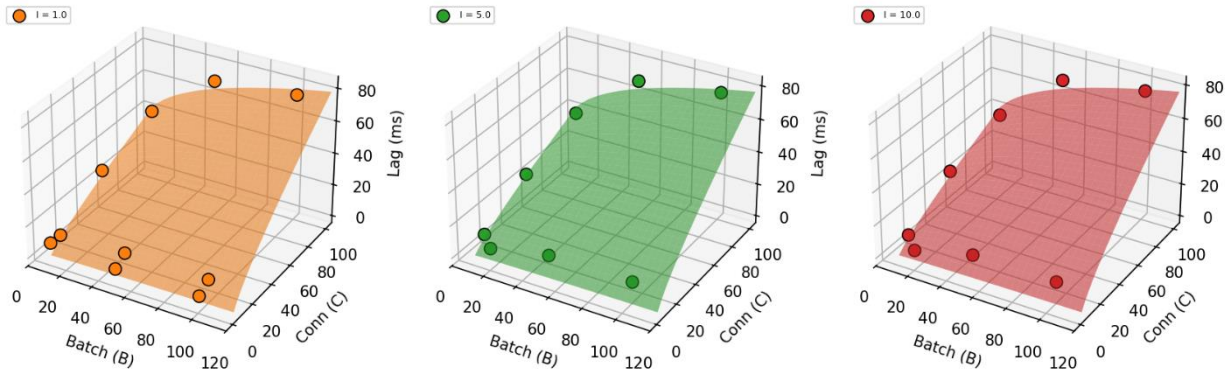
Вид функції	**Рівняння**	**R^2**
Степенева	$lag = 4{,}0668 \cdot B^{0{,}0592} \cdot C^{0{,}4937} \cdot I^{-0{,}0749}$	0,8176
Поліноміально–логаріфмічна	$lag = 5{,}8283 \cdot B^{0{,}0397} \cdot C^{-0{,}2115+0{,}1474\ lnC} \cdot I^{-0{,}0052}$	0,9533
Нелінійна експоненційна	$lag = 0{,}6691 \cdot B^{0{,}2649} \cdot C^{0{,}7602} \cdot \exp(0{,}000031 \cdot C^{1{,}5}) \cdot I^{-0{,}0110}$	0,9786

Для порівняння теоретичних значень з експериментальними побудовано 3D–діаграми (рис. 9, 10). Модель (рис. 10) показала найвищий коефіцієнт детермінації (більше 0,97), що свідчить про те, що вона найбільш точно описує експериментальні дані.



а б

Рисунок 9 – Степенева (а) та поліноміально–логаріфмічна (б) поверхня залежності затримки реплікації (lag(ms)) від розміру пакету документів (Batch(B)) і кількості паралельних з'єднань (Conn(C))



а б в

Рисунок 10 – Нелінійна експоненційна поверхня залежності затримки реплікації (lag(ms)) від розміру пакету документів (Batch(B)) і кількості паралельних з'єднань (Conn(C)) для інтервалів між операціями запису (I): а – I=1 с, б – I=5 с, в – I= 10 с

Висновки. В результаті проведеного дослідження дістали подальшого розвитку закономірності процесу реплікації у розподілених нереляційних базах даних MongoDB за умов застосування методу активного зондування.

За допомогою алгоритму градієнтного бустингу проведено оцінку значущості факторів та визначено основні параметри, що впливають на затримки реплікації, а саме: кількість паралельних клієнтських з'єднань, обсяг документів у пакеті, часовий інтервал між операціями запису. Доведено відсутність істотного впливу часу очікування у чергах та показників завантаження процесора.

На основі отриманих емпіричних даних побудовано нелінійну математичну модель, що описує зв'язок між зазначеними параметрами та затримками реплікації (R^2 = 0,9786), яку може бути використано під час проєктування та оптимізації роботи нереляційних розподілених NoSQL–систем.

ЛІТЕРАТУРА

1. Ncube M. M., Ngulube P. Enhancing environmental decision–making: a systematic review of data analytics applications in monitoring and management. *Discover sustainability*. 2024. Vol. 5, no. 1. DOI: 10.1007/s43621–024–00510–0 (date of access: 23.08.2026).

2. Advances and challenges in iot sensors data handling and processing in environmental monitoring networks / S. Sathyamoorthy et al. *HAFED POLY journal of science, management and technology*. 2024. Vol. 5, no. 2. P. 40–60. DOI: 10.4314/hpjsmt.v5i2.3 (date of access: 22.08.2026).

3. Hasan S., Mohammed Y. Environmental monitoring systems using wireless sensor networks: an overview. *Journal of engineering and sustainable development*. 2020. Vol. 24, Special. P. 298–303. DOI: 10.31272/jeasd.conf.1.33 (date of access: 22.08.2026).

4. Резніченко О. В., Ляшенко О. А., Архипова В. В. Дослідження моделей баз даних для зберігання big data у сфері моніторингу довкілля. Системні технології. *2024*. № 6 (155). С. 185–195. DOI: https://doi.org/10.34185/1562–9945–6–155–2024–18

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

5. Framework for heterogeneous data management: An application case in a NoSQL environment from a Climatological Center / A. M. Jiménez–Galina et al. *Computación y sistemas*. 2024. Vol. 28, no. 1. DOI: 10.13053/cys–28–1–4474 (date of access: 22.08.2026).
6. Han M. The application of NoSQL database in Air Quality Monitoring. *2015 international conference on intelligent systems research and mechatronics engineering*, Zhengzhou, China, 11–13 April 2015. Paris, France, 2015. DOI: 10.2991/isrme–15.2015.25 (date of access: 22.08.2026).
7. Andreoli R., Cucinotta T. Inducing huge tail latency on a mongodb deployment. *2023 IEEE international conference on cloud engineering (IC2E)*, Boston, MA, USA, 25–29 September 2023. 2023. DOI: 10.1109/ic2e59103.2023.00020 (date of access: 22.08.2026).
8. Consistency issue and related trade–offs in distributed replicated systems and databases: a review / J. Ahmed et al. *Radioelectronic and computer systems*. 2023. No. 2. P. 171–179. DOI: 10.32620/reks.2023.2.14 (date of access: 22.08.2026).
9. Gorbenko A., Karpenko A., Tarasyuk O. Analysis of trade–offs in fault–tolerant distributed computing and replicated databases. *IEEE 11th international conference on dependable systems, services and technologies (DESSERT)*, Kyiv, Ukraine, 14–18 May 2020. 2020. DOI: 10.1109/dessert50317.2020.9125078 (date of access: 23.08.2026).
10. Alert conditions: Replication lag. *MongoDB Ops Manager Documentation*. URL:https://www.mongodb.com/docs/ops–manager/current/reference/alerts/replication–lag/ (date of access: 27.07.2026).
11. Hows D., Membrey P., Plugge E. The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB. 3rd ed. Berkeley : Apress, 2015. 428 p
12. Barlett G., Heidemann J., Papadopoulos C. Understanding Passive and Active Service Discovery. *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (San Diego, California, USA, October 24–26, 2007). New York : ACM, 2007. P. 361–374. DOI: https://doi.org/10.1145/1298306.1298356
13. Yang Y. Research on site reliability optimization technology based on synthetic monitoring in cloud environments. *5th international conference on artificial intelligence, automation and high performance computing (AIAHPC)*, Hefei, China, 19–21 September 2025. 2025. P. 729–732. DOI: 10.1109/aiahpc66801.2025.11290592 (date of access: 23.08.2026).
14. Approaches of probing signal type determination of means local probing for distributed power grids technical condition diagnostic / Y. Honcharenko et al. *Herald of khmelnytskyi national university*. 2022. Vol. 305, no. 1. P. 145–151. DOI: 10.31891/2307–5732–2022–305–1–145–151 (date of access: 23.08.2026).
15. Swargo R. M., Arifuzzaman M. Deploy–Efficient and fast network probing with time–series foundation models. *IEEE international conference on big data (bigdata)*, Washington, DC, USA, 15–18 December 2024. 2024. P. 8846–8848.
DOI: 10.1109/bigdata62323.2024.10824943 (date of access: 23.08.2026).
16. Adaptive diagnosis in distributed systems / I. Rish et al. *IEEE transactions on neural networks*. 2005. Vol. 16, no. 5. P. 1088–1109. DOI: 10.1109/tnn.2005.853423 (date of access: 23.08.2026).
17. Multipath probing and grouping in multihomed networks / J. Liao et al. *IEICE transactions on information and systems*. 2011. E94–D, no. 3. P. 710–713.

DOI: 10.1587/transinf.e94.d.710 (date of access: 23.08.2026).
18. Cito J., Gotowka D., Leitner P. Identifying web performance degradations through synthetic and real–user monitoring / J. Cito, D. Gotowka, P. Leitner, R. Pelette, D. Suljoti, S. Dustdar. *Journal of Web Engineering*. 2015. Vol. 14, № 5–6. P. 414–442.
URL: https://journals.riverpublishers.com/index.php/JWE/article/view/3845
19. Empirical and theoretical evaluation of active probing attacks and their countermeasures / X. Fu et al. *Information hiding*. Berlin, Heidelberg, 2004. P. 266–281. DOI: 10.1007/978–3–540–30114–1_19 (date of access: 23.08.2026).
20. Johnson M. Proactive Monitoring Strategies for High–Availability Database Systems / M. Johnson et al. 2020.
21. Badia A. Active database systems for monitoring and surveillance. *Intelligence and security informatics*. Berlin, Heidelberg, 2003. P. 296–307. DOI: 10.1007/3–540–44853–5_22 (date of access: 23.08.2026).
22. Dhir K. Synthetic Monitoring as a Quality Gate in Continuous Integration and Deployment (CI/CD) and AI Augmentation. *International Research Journal of Modernization in Engineering Technology and Science*. 2026. Vol. 8, № 4. P. 14.
DOI: https://www.doi.org/10.56726/IRJMETS95831
23. MongoDB. (n.d.). *Deploy a replica set for testing and development*. MongoDB Manual. Retrieved July 27, 2026, from https://www.mongodb.com/docs/manual/tutorial/deploy–replica–set–for–testing/
24. Definitive guide to mongodb: a complete guide to dealing with big data using mongodb / T. Hawkins et al. Apress L. P., 2013.
25. MongoDB. (n.d.). *Journaling.* MongoDB Manual. Retrieved July 27, 2026, from https://www.mongodb.com/docs/manual/core/journaling/
26. MongoDB. (n.d.). *WiredTiger Storage Engine.* MongoDB Manual. Retrieved July 27, 2026, from https://www.mongodb.com/docs/manual/core/wiredtiger/#std–label–storage–wiredtiger/
27. Gotelli N. J. A Primer of Ecological Statistics / N. J. Gotelli, A. M. Ellison. – 2nd ed. – Sunderland, Massachusetts: Sinauer Associates, Inc., 2013. – 614 p.

REFERENCES

1. Ncube, M. M., & Ngulube, P. (2024). Enhancing environmental decision–making: a systematic review of data analytics applications in monitoring and management. *Discover sustainability*, *5*(1). https://doi.org/10.1007/s43621–024–00510–0
2. Sathyamoorthy, S., Matthew, U. O., Adekunle, T. S., & Okafor, N. U. (2024). Advances and challenges in iot sensors data handling and processing in environmental monitoring networks. *HAFED POLY Journal of Science, Management and Technology*, *5*(2), 40–60. https://doi.org/10.4314/hpjsmt.v5i2.3
3. Hasan, S., & Mohammed, Y. (2020). Environmental monitoring systems using wireless sensor networks: An overview. *Journal of Engineering and Sustainable Development*, *24*(Special), 298–303. https://doi.org/10.31272/jeasd.conf.1.33

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

4. Reznichenko, O., Liashenko, O., & Arkhypova, V. (2025). Database models for storing big data. *System Technologies*, *6*(155), 155–195. https://doi.org/10.34185/1562–9945–6–155–2024–18

5. Jiménez–Galina, A. M., Maldonado–Macías, A. A., Olmos–Sanchez, K. M., Hernandez, I., Estrada–Saldana, F., & Vázquez–Gálvez, F. A. (2024). Framework for heterogeneous data management: An application case in a NoSQL environment from a Climatological Center. *Computación y sistemas*, *28*(1). https://doi.org/10.13053/cys–28–1–4474

6. Han, M. (2015). The application of NoSQL database in Air Quality Monitoring. *2015 international conference on intelligent systems research and mechatronics engineering*. Atlantis Press. https://doi.org/10.2991/isrme–15.2015.25

7. Andreoli, R., & Cucinotta, T. (2023). Inducing huge tail latency on a mongodb deployment. *2023 IEEE international conference on cloud engineering (IC2E)*. IEEE. https://doi.org/10.1109/ic2e59103.2023.00020

8. Ahmed, J., Karpenko, A., Tarasyuk, O., Gorbenko, A., & Sheikh–Akbari, A. (2023). Consistency issue and related trade–offs in distributed replicated systems and databases: A review. *Radioelectronic and Computer Systems*, (2), 171–179. https://doi.org/10.32620/reks.2023.2.14

9. Gorbenko, A., Karpenko, A., & Tarasyuk, O. (2020). Analysis of trade–offs in fault–tolerant distributed computing and replicated databases. *IEEE 11th international conference on dependable systems, services and technologies (DESSERT)*. IEEE. https://doi.org/10.1109/dessert50317.2020.9125078

10. MongoDB. (n.d.). *Alert conditions: Replication lag*. MongoDB Ops Manager Documentation. Retrieved July 27, 2026, from https://www.mongodb.com/docs/ops–manager/current/reference/alerts/replication–lag/

11. Hows, D., Membrey, P., Plugge, E., & Hawkins, T. (2015). *The definitive guide to mongodb*. Apress. https://doi.org/10.1007/978–1–4842–1182–3

12. Bartlett, G., Heidemann, J., & Papadopoulos, C. (2007). Understanding passive and active service discovery. *The 7th ACM SIGCOMM conference*. ACM Press. https://doi.org/10.1145/1298306.1298314

13. Yang, Y. (2025). Research on site reliability optimization technology based on synthetic monitoring in cloud environments. *5th international conference on artificial intelligence, automation and high performance computing (AIAHPC)* (p. 729–732). IEEE. https://doi.org/10.1109/aiahpc66801.2025.11290592

14. Honcharenko, Y., Priadko, V., Poleshchuk, I., Synkivskyi, V., & Ostra, N. (2022). Approaches of probing signal type determination of means local probing for distributed power grids technical condition diagnostic. *Herald of Khmelnytskyi National University*, *305*(1), 145–151. https://doi.org/10.31891/2307–5732–2022–305–1–145–151

15. Swargo, R. M., & Arifuzzaman, M. (2024). Deploy–Efficient and fast network probing with time–series foundation models. *IEEE international conference on big data (bigdata)* (p. 8846–8848). IEEE. https://doi.org/10.1109/bigdata62323.2024.10824943

16. Rish, I., Brodie, M., Ma, S., Odintsova, N., Beygelzimer, A., Grabarnik, G., & Hernandez, K. (2005). Adaptive diagnosis in distributed systems. *IEEE Transactions on Neural Networks*, *16*(5), 1088–1109. https://doi.org/10.1109/tnn.2005.853423

17. Liao, J., Wang, J., Li, T., & Zhu, X. (2011). Multipath probing and grouping in multi-homed networks. *IEICE Transactions on Information and Systems*, *E94–D*(3), 710–713. https://doi.org/10.1587/transinf.e94.d.710
18. Cito, J., Gotowka, D., Leitner, P., Pelette, R., Suljoti, D., & Dustdar, S. (2015). Identifying web performance degradations through synthetic and real–user monitoring. *Journal of Web Engineering*, *14*(5–6), 414–442.
Retrieved from https://journals.riverpublishers.com/index.php/JWE/article/view/3845
19. Fu, X., Graham, B., Xuan, D., Bettati, R., & Zhao, W. (2004). Empirical and theoretical evaluation of active probing attacks and their countermeasures. *Information hiding* (p. 266–281). Springer Berlin Heidelberg. https://doi.org/10.1007/978–3–540–30114–1_19
20. Johnson, M., Carter, E., Brooks, S., Bennett, A., & Yusuff, M. (2020). *Proactive monitoring strategies for high–availability database systems.*
21. Badia, A. (2003). Active database systems for monitoring and surveillance. *Intelligence and security informatics* (p. 296–307). Springer Berlin Heidelberg. https://doi.org/10.1007/3–540–44853–5_22
22. Dhir, K. (2024). Synthetic monitoring as a quality gate in continuous integration and deployment (CI/CD) and AI augmentation. *International Research Journal of Modernization in Engineering Technology and Science*, *6*(4), 14. https://doi.org/10.56726/IRJMETS95831
23. MongoDB. (n.d.). *Deploy a replica set for testing and development*. MongoDB Manual. Retrieved July 27, 2026, from https://www.mongodb.com/docs/manual/tutorial/deploy–replica–set–for–testing/
24. Hawkins, T., Membrey, P., Plugge, E., & Hows, D. (2013). *Definitive guide to mongodb: A complete guide to dealing with big data using mongodb*. Apress L. P.
25. MongoDB. (n.d.). *Journaling.* MongoDB Manual. Retrieved July 27, 2026, from https://www.mongodb.com/docs/manual/core/journaling/
26. MongoDB. (n.d.). *WiredTiger Storage Engine.* MongoDB Manual. Retrieved July 27, 2026, from https://www.mongodb.com/docs/manual/core/wiredtiger/#std–label–storage–wiredtiger/
27. Gotelli, N. J., & Ellison, A. M. (2013). *A primer of ecological statistics* (2nd ed.). Sinauer Associates, Inc.

Received 08.06.2026.
Accepted 10.08.2026.
Published 30.10.2026

Mathematical modeling of replication lags in a distributed MongoDB database

Abstract. This paper addresses the topical problem of identifying the factors influencing replication lags and constructing a model of these delays in a distributed MongoDB database under heavy load conditions, using the active probing method. Its application made it possible to overcome the limitations of the discretization inherent in standard monitoring tools—namely, the one–second interval for collecting metrics—and to ensure the acquisition of accurate replication lags values by recording data with precise timestamps on the primary node and reading them from secondary nodes.

Based on the experimental data obtained regarding the impact of the write guarantee level, operation type, processor load, queue wait time, number of concurrent client

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

connections, number of documents in a batch, and the time interval between write operations, an assessment of feature importance was conducted using gradient boosting algorithms. The analysis showed that the number of concurrent client connections, the volume of documents in a batch, and the time interval between write operations have a decisive impact on the occurrence and increase of replication delays. It was also confirmed that queue wait times and processor load do not have a significant impact on the formation of delays.

Based on the study's results, a nonlinear mathematical model of the cluster's operation was developed (coefficient of determination $R^2 = 0.9786$), which can be used during the design, optimization, and configuration of distributed NoSQL systems to predict replication parameters and select optimal system operating parameters.

Keywords: gradient boosting, prediction, feature importance, approximation, coefficient of determination, machine learning, environmental monitoring, mathematical modeling, active probing method, synthetic monitoring.

Резніченко Олег Вікторович – аспірант кафедри інформаційних систем, Український державний університет науки та технологій, Дніпро, Україна.
ORCID: https://orcid.org/0009–0001–0194–5079
Ляшенко Оксана Анатоліївна – кандидат технічних наук, доцент, завідувач кафедри інформаційних систем, Український державний університет науки та технологій, Дніпро, Україна.
ORCID: https://orcid.org/0000–0002–9983–5504
Архипова Вікторія Вікторівна – кандидат технічних наук, доцент, старший науковий співробітник Наукового центру інноваційних матеріалів і технологій, Український державний університет науки та технологій, Дніпро, Україна.
ORCID: https://orcid.org/0000–0002–9722–5622

Reznichenko Oleh – Postgraduate student of Department of Information Systems, Ukrainian State University of Science and Technologies, Dnipro, Ukraine.
ORCID: https://orcid.org/0009–0001–0194–5079
Liashenko Oksana – Candidate of Technical Sciences, Associate Professor, Head of the Department of Information Systems, Ukrainian State University of Science and Technologies, Dnipro, Ukraine.
ORCID: https://orcid.org/0000–0002–9983–5504
Arkhypova Viktoriia – Candidate of Technical Sciences, Associate Professor, Senior Research Officer at the Scientific Centre for Innovative Materials and Technologies, Ukrainian State University of Science and Technologies, Dnipro, Ukraine.
ORCID: https://orcid.org/0000–0002–9722–5622