УДК 004.65
DOI https://doi.org/10.32689/maup.it.2024.3.2

Володимир КОЗУБ
магістр за спеціальністю «Системи захисту від несанкціонованого доступу», Національний авіаційний університет, volodymyr.kozub85@gmail.com
***ORCID:** 0009-0007-6740-5300*

ТЕХНОЛОГІЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЗБЕРІГАННЯ У NO-SQL БАЗАХ ДАНИХ

Анотація. *У статті представлено результати використання методів дедуплікації і стиснення для оптимізації зберігання даних у хмарних No-SQL базах даних.*

Метою *роботи є зменшення обсягу даних, що зберігаються, за рахунок використання технології Hadoop MapReduce для обробки інформації та MongoDB для зберігання агрегованих пар ключ-значення.*

Методологія. *Дослідження базується на комбінації дедуплікації та стиснення даних, виконаних за допомогою Hadoop MapReduce. Цей підхід дозволяє обробляти великі обсяги інформації, оптимізуючи процеси зберігання в MongoDB.*

Результати. *Проведено серію експериментів для оцінки зменшення обсягів даних та перевірки швидкості обробки запитів. Запропонована архітектура системи демонструє легкість інтеграції з існуючими інструментами резервного копіювання, що робить цю технологію практичною для впровадження в реальних умовах. Результати експериментів свідчать про високу ефективність застосування даних технологій для великих файлів, що дозволяє зменшити вимоги до сховища на понад 90%.*

Наукова новизна. *Запропоноване рішення впроваджує інноваційний підхід до обробки та зберігання даних у хмарних середовищах. Вперше в контексті No-SQL баз даних об'єднуються методи дедуплікації та стиснення, що створює нові можливості для економії місця і підвищення продуктивності систем. Дослідження розширює застосування цих методів, включаючи потенціал для мультимедійних файлів та потокових даних у реальному часі.*

Висновки. *Отримані результати свідчать про високу ефективність використання технології дедуплікації та стиснення для зменшення обсягів даних у хмарних No-SQL базах. Впровадження даних методів дозволяє значно знизити витрати на зберігання, підвищити швидкість обробки даних та адаптуватися до зростаючих потреб сучасних індустрій. Наступні етапи дослідження включатимуть розробку прогностичних моделей для оптимізації застосування технологій у реальному часі, що відкриває нові горизонти в управлінні даними.*

Ключові слова: *No-SQL бази даних, дедуплікація даних, стиснення даних, оптимізація хмарного зберігання, ефективність зберігання.*

Volodymyr KOZUB. TECHNOLOGY FOR IMPROVING STORAGE EFFICIENCY IN NO-SQL DATABASES

Abstract. *The article presents the results of using deduplication and compression methods to optimize data storage in cloud No-SQL databases.*

The purpose of the article *is to reduce the volume of stored data by using Hadoop MapReduce technology for information processing and MongoDB for storing aggregated key-value pairs.*

Methodology. *The study is based on a combination of data deduplication and compression performed using Hadoop MapReduce. This approach allows you to process large amounts of information by optimizing storage processes in MongoDB.*

Results. *A series of experiments was conducted to evaluate the reduction of data volumes and check the speed of processing requests. The proposed system architecture demonstrates ease of integration with existing backup tools, making this technology practical for implementation in real-world environments. The results of the experiments indicate the high efficiency of the application of these technologies for large files, which allows to reduce storage requirements by more than 90%.*

Scientific novelty. *The proposed solution introduces an innovative approach to data processing and storage in cloud environments. For the first time in the context of No-SQL databases, deduplication and compression methods are combined, which creates new opportunities for saving space and increasing system performance. The research extends the applications of these techniques to include the potential for multimedia files and real-time streaming data.*

Conclusions. *The obtained results testify to the high efficiency of using deduplication and compression technology to reduce data volumes in cloud-based No-SQL databases. The implementation of these methods allows you to significantly reduce storage costs, increase the speed of data processing and adapt to the growing needs of modern industries. The next stages of research will include the development of predictive models to optimize the application of technologies in real time, which opens new horizons in data management.*

Key words: *No-SQL database, data deduplication, data compression, cloud storage optimization, storage efficiency.*

Вступ. Постановка проблеми. Сьогодні хмарні обчислення змінили способи управління, а також зберігання даних. Натепер пропонуються гнучки, масштабовані та ефективні рішення, які використовують Інтернет. Так як обсяги даних продовжують зростати в геометричній прогресії, аутсорсинг управління базами даних у хмарних середовищах став важливим компонентом сучасних бізнес-стратегій. Ті ж самі популярні соціальні медіаплатформи, такі як Facebook і Twitter, кожного дня генерують великий обсяг даних, який обраховується сотнями терабайтів на день. а фондові біржі обробляють потоки даних, які більше, ніж терабайт на годину.

Всі ці величезні обсяги даних вимагають постійного збереження, оновлення, синхронізації, що обумовлює використання хмарних платформ, які дозволяють компаніям позбутися власних серверів і перейти повністю на використання аутсорс систем збереження даних. Цей крок давно економічно обґрунтований і є більш вигідним для компанії, ніж підтримка власних технічних і програмних рішень для роботи з масивами даними. Простота збереження великих обсягів даних також несе за собою і негативний фактор, компанії виявилось простіше працювати з неструктурованими і напівструктурованими даними, що викликає великі труднощі для традиційних реляційних систем керування базами даних (СКБД). Платформи СКБД здебільш розроблені для структурованих даних і направлені на прискорені роботу з ними, в той час як нереляційні системи (No-SQL системи (Not only SQL)) намагаються відмовитись від прийнятих рішень в реляційних СКБД. No-SQL бази даних вирішують задачі масштабованості і гнучкість через інструменти підтримки кінцевої узгодженості, які увійшли в концепцію BASE («базова доступність» («basically available»), «м'який стан» («soft state»), «кінцева узгодженість» («eventual consistency»)), та роблять їх придатними для розподілених хмарних мереж і реалізації додатків для роботи в режимі реального часу.

Однак велика кількість дубльованих даних у таких системах викликає нові труднощі, зокрема зростання витрат на їх зберігання. Коли потрібно керувати великим обсягом скопійованих даних, вартість зберігання збільшується, а загальна ефективність роботи системи може знижуватися. Окрім цього, із ростом обсягів даних організації стикаються з викликами, пов'язаними з розширенням систем, забезпеченням їхньої безпеки та сумісності. Для вирішення цих проблем компанії комбінують локальні сховища даних із хмарними резервними копіями, а також застосовують передові методи управління даними. Одними з найбільш дієвих підходів є видалення дубльованих даних (дедуплікація) і стиснення, що зменшує розмір збереженої інформації. Це дозволяє суттєво економити місце у сховищах і мережеві ресурси.

Пропонується застосовувати для покращення зберігання даних у хмарних базах No-SQL методи дедуплікації (усунення дублювань) і стиснення, щоб зменшити зайві дані та оптимізувати зберігання. Це виконується за рахунок використання платформи MapReduce, яка видаляє дублікати з даних розподіленої файлової системи Hadoop (HDFS), що формує унікальні записи у форматі «ключ-значення». А вже перетворені записи зберігаються в No-SQL базі даних MongoDB. Після цього дані стискаються за допомогою алгоритму Gzip, що ще більше зменшує обсяг необхідного для них місця та підвищує швидкість передачі даних між різними вузлами в хмарі.

Інтеграція можливостей великомасштабної обробки даних Hadoop із гнучкою структурою MongoDB дозволяє в новому підході пропонувати ефективні рішення для роботи з великими обсягами інформації в розподілених хмарних середовищах та знижувати витрати на зберігання даних без втрат в загальній продуктивності системи.

Аналіз останніх досліджень і публікацій. Недавні дослідження широко досліджували застосування методів дедуплікації в хмарних системах зберігання, використовуючи такі концепції, як хешування та MapReduce для оптимізації керування даними. Рой-Хубара та Штурм підкреслили, що традиційні методи проектування є неадекватними для сучасних середовищ баз даних, особливо з точки зору адаптивності та масштабованості [1]. Їх систематичний огляд виявив прогалини у вирішенні нефункціональних вимог, які є критичними для продуктивності розподілених систем.

Рамзан та ін. досліджували основні проблеми, з якими стикається розподілене зберігання в No-SQL, включаючи узгодженість, затримку, пропускну здатність і безпеку [2]. Їх систематичний огляд підкреслив важливість таких методів, як індексування, кешування та хешування, для підвищення продуктивності високонавантажених систем. Ці проблеми актуальні в додатках із великомасштабною реплікацією даних і обробкою запитів, де навіть невелика неефективність може призвести до значного зниження продуктивності системи. Вони запропонували кілька підходів до вирішення цих проблем, зосередившись на балансуванні навантаження системи та оптимізації запитів.

Кім і Лі досліджували стратегії дедуплікації даних для підвищення ефективності зберігання, забезпечуючи при цьому цілісність і конфіденційність даних [3]. Їх робота підкреслила проблеми сумісності між методами шифрування та дедуплікації, підкресливши необхідність безпечних методів, які не впливають на продуктивність. Це важливо в хмарних середовищах із високим навантаженням, де безпека даних так само важлива, як і оптимізація зберігання.

Кумар та ін. представив покращений підхід для дедуплікації з використанням диференціальної еволюції в системах зберігання великих даних [4]. Їхній метод значно покращує як пропускну здатність, так і коефіцієнт дедуплікації, що робить його життєздатним рішенням для керування великомасштабними системами розподілених даних. Їх результати продемонстрували значне покращення використання сховища, особливо в хмарних і розподілених середовищах.

Ванг та ін. надав ширший огляд методів дедуплікації, наголошуючи на зменшенні простору для зберігання та покращеному управлінні даними [5]. Вони проаналізували продуктивність алгоритмів дедуплікації в різних хмарних середовищах зберігання, наголошуючи на масштабованості як критичному факторі успіху у великих системах. У документі також обговорюється компроміс між точністю дедуплікації та продуктивністю системи, що є критичним фактором у середовищах із високим трафіком.

Чжан та ін. заглибились у методи дедуплікації, що використовуються в хмарних системах, порівнюючи сильні та слабкі сторони різних методів [6]. Вони підкреслили застосовність і ефективність цих методів у сценаріях реального світу, де обсяг даних і моделі доступу динамічно змінюються. Їх огляд показав, як різні підходи до дедуплікації впливають на час відгуку системи та економію місця для зберігання.

Огляд рішень в роботі [7-9] окреслив ключові проблеми масштабованості, безпеки та ефективності, пропонуючи потенційні рішення для подолання складності, притаманної хмарній дедуплікації та запропонував компромісні рішення між глибиною дедуплікації та споживанням системних ресурсів, пропонуючи масштабовані методи, які оптимізують обидва показники.

Коушик та ін. розглянули технічні вузькі місця, що виникають через дедуплікацію даних, такі як обмеження дискового введення/виведення та керування метаданими [10]. У своїй роботі вони представили інноваційні рішення для подолання цих перешкод, включаючи оптимізовані алгоритми та масштабовані методи індексування, які мають вирішальне значення для підтримки високої продуктивності та ефективності системи у великомасштабних хмарних середовищах. Вони підкреслили важливість дедуплікації в режимі реального часу в системах реального часу, особливо там, де навантаження системи та пропускна здатність даних постійно змінюються.

Поєднання цих методів вирішує проблеми оптимізації зберігання, покращуючи продуктивність системи, використання пропускної здатності та безпеку даних у хмарних і розподілених середовищах.

Постановка завдання. Метою статті є аналіз технології підвищення ефективності зберігання даних у хмарних No-SQL за рахунок оптимізації процесу зберігання і зменшення надмірності даних. Передбачається, що використання паралельної обробку через Hadoop MapReduce та MongoDB для зберігання унікальних пар ключ-значення дозволить суттєво зменшити обсяг даних, що передаються, і збільшити пропускну здатність мережі.

Виклад основного матеріалу.

Загальний підхід до використання No-SQL баз даних в хмарних обчисленнях. Практичне використання No-SQL баз даних у хмарних обчисленнях дозволяє досягати високої швидкості обробки та масштабування даних для додатків із великим навантаженням. Це обумовлено основними можливостями No-SQL баз даних: горизонтальне масштабування на вимогу; резервування та реплікація даних; гнучке управління ресурсами; оптимізація запитів і кешування; автоматизація процесів резервного копіювання; моніторинг та налаштування тривог.

Горизонтальне масштабування передбачає автоматичне додавання нових вузлів, коли обсяг запитів зростає, що забезпечує стабільну роботу без втрати швидкості обробки. Дані автоматично дублюються на кількох вузлах, щоб забезпечити доступність навіть при відмовах (цей процес налаштовується таким чином, щоб зменшити ризик втрати даних і забезпечити швидке відновлення після збою).

Для економії ресурсів No-SQL бази налаштовуються з урахуванням поточного навантаження, щоб автоматично масштабуватися вгору або вниз, забезпечуючи мінімальну кількість активних вузлів при зменшенні обсягу запитів. Налаштування кешування та оптимізація запитів забезпечують мінімальні затримки доступу до даних, особливо для найбільш запитуваної інформації. Це дозволяє зменшити навантаження на основну базу та підвищити швидкість відповіді.

У хмарі легко налаштувати регулярне резервне копіювання, яке автоматично зберігає дані на окремих серверах, що забезпечує додатковий рівень захисту даних і швидке відновлення в разі непередбачуваних ситуацій. За допомогою хмарних інструментів проводиться моніторинг продуктивності No-SQL бази даних. А налаштовані тривоги дозволяють оперативно реагувати на потенційні проблеми (перевантаження або збої у доступності) ще до того, як їх вплив на систему досягне критичного.

Однак одна з проблем баз даних No-SQL полягає в їх сильно денормалізованій структурі, що призводить до значних обсягів надлишкових даних. Зі збільшенням обсягу надлишкових даних організаціям стає все складніше ефективно управляти ними. Це робить вкрай необхідним використання інструментів і стратегій для підвищення ефективності зберігання даних. Досягнення цієї мети може значно знизити операційні витрати, що актуально з огляду на стрімке зростання як мережевих вимог, так і ресурсів, необхідних для зберігання і передачі даних.

Для вирішення цих проблем вагому роль відіграють такі технології, як дедуплікація та стиснення. Усуваючи дублікати даних і зменшуючи розмір файлів, що зберігаються, ці методи можуть дати значну

економію в обсязі необхідного сховища. Оскільки обсяги даних продовжують зростати в геометричній прогресії, особливо в хмарних системах, які обробляють терабайти інформації, застосування цих методів стає необхідним для підтримки економічно ефективних і масштабованих рішень для управління даними.

Системна архітектура запропонованої технології. Архітектура запропонованої технології складається з декількох рівнів, які показано на рисунку 1. Процес починається з отримання даних із розподіленої файлової системи Hadoop (HDFS). HDFS спеціально створена для того, щоб зберігати дуже великі обсяги інформації, які розподіляються між багатьма комп'ютерами, об'єднаними в кластер.

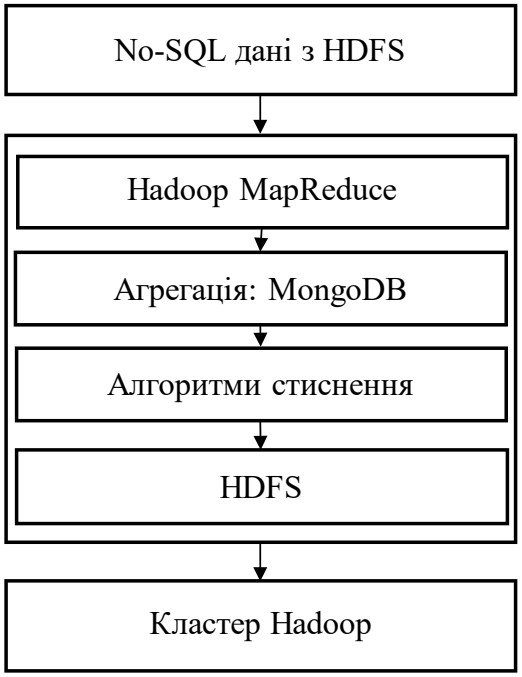

Рис. 1. Системна архітектура запропонованої технології

У кожному з цих комп'ютерів або «вузлів» зберігається частина даних, що дозволяє обробляти величезні файли, які можуть займати від сотень мегабайт до кількох терабайт. Завдяки тому, що дані розподілені між багатьма вузлами, система може легко збільшувати обсяги зберігання, додаючи нові комп'ютери в кластер. Це робить HDFS ідеальним вибором для середовищ, де потрібно обробляти та зберігати великі масиви інформації, як-от в аналітичних системах чи платформах для обробки великих даних.

У хмарних середовищах дані зберігаються у вигляді пласких текстових файлів, що може призвести до значних накладних витрат на здатність операційної системи отримувати дані і на управління ними, а це робить даний формат непрактичним для роботи з великими наборами даних. Традиційні реляційні бази даних не мають необхідних інструментів для ефективної обробки таких обсягів, але цю проблему вирішує фреймворк MapReduce за рахунок можливості паралельної обробки.

Після завершення процесу видалення дублікатів дані зводяться до набору пар «ключ-значення», які зберігаються у вигляді таблиці. Для цього використовується MongoDB – No-SQL база даних, що працює з документами. Завдяки документній структурі MongoDB можна легко працювати зі складними даними, адже документи можуть включати вкладені ключі та значення, а також масиви, що дуже зручно для зберігання багаторівневих чи багатовимірних даних.

Далі відбувається стиснення даних, щоб оптимізувати їх зберігання. Стиснуті файли знову зберігаються у HDFS, що суттєво зменшує загальний обсяг пам'яті, яку займають дані. Це не лише економить простір для зберігання, але й робить передачу між вузлами більш ефективною, оскільки зменшені файли потребують менше мережевих ресурсів. Завдяки поєднанню методів видалення дублікатів, структурованого зберігання в MongoDB та стиснення забезпечується ефективне управління великими обсягами даних, оптимізуючи як зберігання, так і мережеву передачу.

Експеримент. Відкритість платформ Hadoop та MongoDB дозволяє самостійно не тільки налаштовувати модулі, а й переписувати коди окремих компонентів. У дослідженні використовувалися версії

Hadoop 3.4.0 і MongoDB 7.0, розгорнуті на 64-розрядній системі Ubuntu, яка підтримує завантаження двох операційних систем. Комп'ютер має 16 ГБ оперативної пам'яті та 500 ГБ жорсткого диска, налаштованого для роботи з розподіленою файловою системою.

Після налаштування Hadoop і MongoDB додано бібліотеки для стиснення даних за допомогою алгоритму GZip, щоб зробити зберігання більш ефективним. Усі процеси запускалися в середовищі розробки Eclipse Juno, де були інтегровані необхідні бібліотеки для роботи з Hadoop і MongoDB. Для оцінки використання пропускної здатності було також підключено клієнтську систему з такою ж конфігурацією. На рисунку 2 показано весь процес роботи в Hadoop HDFS.

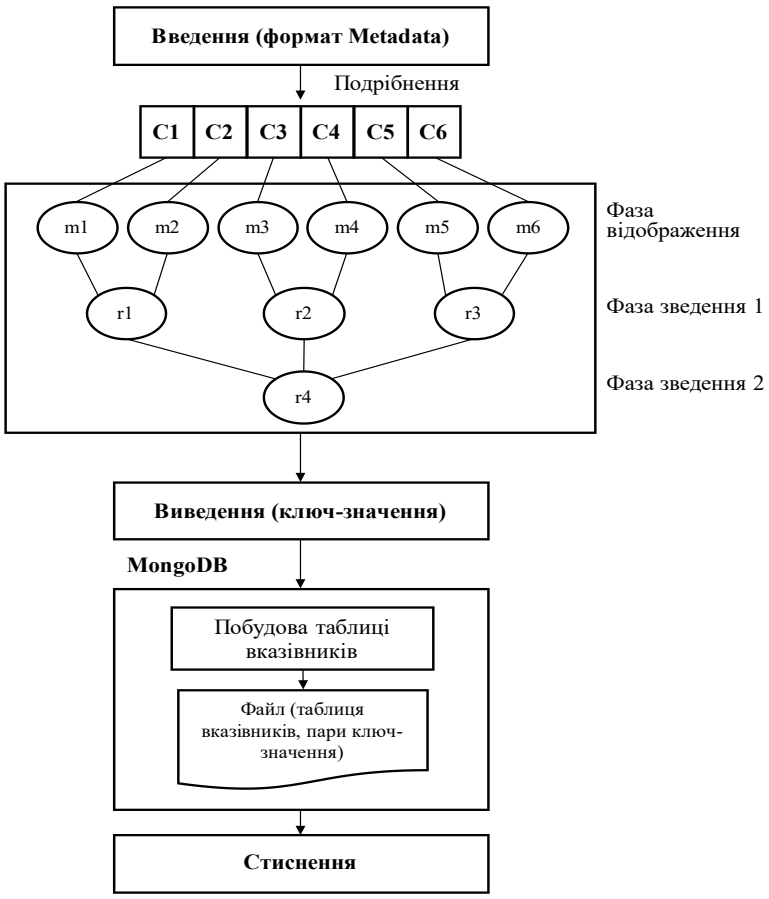

Рис. 2. Фаза виконання процесу Hadoop HDFS

Спершу вхідні дані завантажуються з HDFS, представлені у вигляді метаданих, пов'язаних із текстовими файлами. Ці метадані допомагають розбивати записи на фрагменти для дедуплікації, що дозволяє порівнювати лише однорідні типи даних, підвищуючи ефективність процесу. Отримані фрагменти змінного розміру потім обробляються на етапі MapReduce для видалення дублікатів даних.

Процес дедуплікації передбачає розбиття потоку даних на файли змінної довжини і запис їх на диск. Кожному сегменту присвоюється унікальний цифровий підпис за допомогою хешування md5, що допомагає ідентифікувати дублікати даних. Процес дедуплікації керується за допомогою фреймворку MapReduce, який працює за архітектурою master-slave. Головний вузол призначає робочим вузлам завдання зі створення мапи та редукції, гарантуючи, що всі релевантні сегменти даних будуть оброблені ефективно. На етапі відображення Hadoop хешує ключі і гарантує, що всі значення, пов'язані з певним ключем, будуть спрямовані до відповідного редуктора. Це забезпечує дедуплікацію як на рівні файлів, так і на рівні підфайлів.

В момент, коли головний вузол ініціює завдання, робочі вузли паралельно виконують відповідну карту або зменшують завдання. MapReduce інтерпретує дані під час обробки, причому завдання починається з розділення вхідних даних і призначення кожної частини окремим завданням карти. Після того, як робочий вузол завершує завдання відображення, проміжні пари ключ-значення, згенеровані завданням, зберігаються в пам'яті.

Фаза редукції починається зі збору всіх проміжних результатів з вихідних даних карти, застосовуючи функцію редукції для консолідації даних. Щоб підвищити ефективність зберігання та скоротити час обробки, проміжні пари ключ-значення з кожної задачі картографування не зберігаються окремо. Замість цього дані залишаються в пам'яті і передаються безпосередньо до редукторів, усуваючи непотрібні перетворення і прискорюючи процес виконання. Цей же принцип застосовується між редукторами, де дані передаються без перетворення в пари ключ-значення до останнього кроку редукції.

Після етапу MapReduce результати зберігаються в MongoDB завдяки її високій продуктивності та масштабованості. MongoDB особливо добре підходить для No-SQL додатків завдяки своїй структурі на основі документів, яка групує записи в колекції документів. Кожен документ функціонує як пара ключ-значення, ідентифікована унікальним ідентифікатором, згенерованим за допомогою хешування. У цьому випадку таблиця покажчиків створюється шляхом об'єднання всіх пар ключ-значення в документ MongoDB. Документно-орієнтована структура зберігання забезпечує легкий пошук та інтеграцію, дозволяючи відтворити оригінальний файл, зібравши дані з таблиці покажчиків.

Hadoop забезпечує відмінну продуктивність читання і запису, з вбудованою підтримкою ряду алгоритмів стиснення, які можуть бути обрані на основі специфіки даних в No-SQL додатках. Хоча стиснення допомагає економити місце на диску, воно може призвести до дещо більшого використання процесора на етапах стиснення і розпакування даних під час читання і запису. В запропонованому рішенні використовується алгоритм стиснення GZIP, щоб максимально заощадити місце на диску. GZIP, частина власної бібліотеки Hadoop, покладається на алгоритм DEFLATE, який поєднує в собі кодування Хаффмана та методи стиснення LZ77. Цей алгоритм забезпечує швидку декомпресію, гарантуючи, що файли можуть бути легко відновлені з мінімальним часом обробки.

Результати експерименту. Для експерименту створені тестові дані з урахуванням їх розміру, надмірності та складності кортежів, які складалися з 12 текстових файлів із різними характеристиками. Щоб оцінити ефективність запропонованої технології підвищення ефективності зберігання у No-SQL базах даних, виконувалися послідовні резервні копії баз даних до і після дедуплікації та стиснення, що дозволило точно порівняти розмір файлів. Розмір кожного файлу також фіксувався після тестів із записами різного обсягу, як показано в Таблиці 1.

Таблиця 1

Порівняння за розмірами файлів

Розмір файлу (Кб)	Кількість рядків	Розмір файлу після дедуплікації	Розмір файлу після дедуплікації та стиснення
26 520,2	1000	20580,5	13520,8
55 290,7	2000	41990,1	24580,4
111 610,7	5000	75790,2	29710,0
231 420,4	10000	145 450,1	36880,6

Результати показали, що запропонована технологія особливо ефективна для обробки великих файлів, на відміну від файлів меншого розміру з аналогічним вмістом. Чим більше надлишкової інформації в наборі даних, тим краще запропоновані методи справляються з оптимізацією зберігання, особливо коли йдеться про великі обсяги даних.

Експерименти показують, що одна лише дедуплікація може звільнити понад 50% місця в сховищі, з потенціалом ще більшої економії при застосуванні стиснення поверх дедуплікованих даних. Фактично, понад 91% місця в сховищі можна заощадити, якщо повністю застосувати запропоновану технологію до No-SQL сховища даних. Це візуально представлено на рисунку 3, який підкреслює різке зменшення розмірів файлів після застосування дедуплікації з подальшим стисненням.

З діаграм видно, що стиснення дає суттєву перевагу в подальшій мінімізації сховища після виконання дедуплікації в No-SQL середовищах. Ефективність економії пам'яті залежить від типу даних, що обробляються. Відповідно до словника SNIA, ефективність зберігання визначається як відношення ефективної ємності системи зберігання до її початкової ємності. Формула виглядає наступним чином:

$$\frac{fs - ls}{fs} \times 100 ,$$

fs – оригінальний розмір файлу;

ls – розмір файлу після дедуплікації та стиснення.

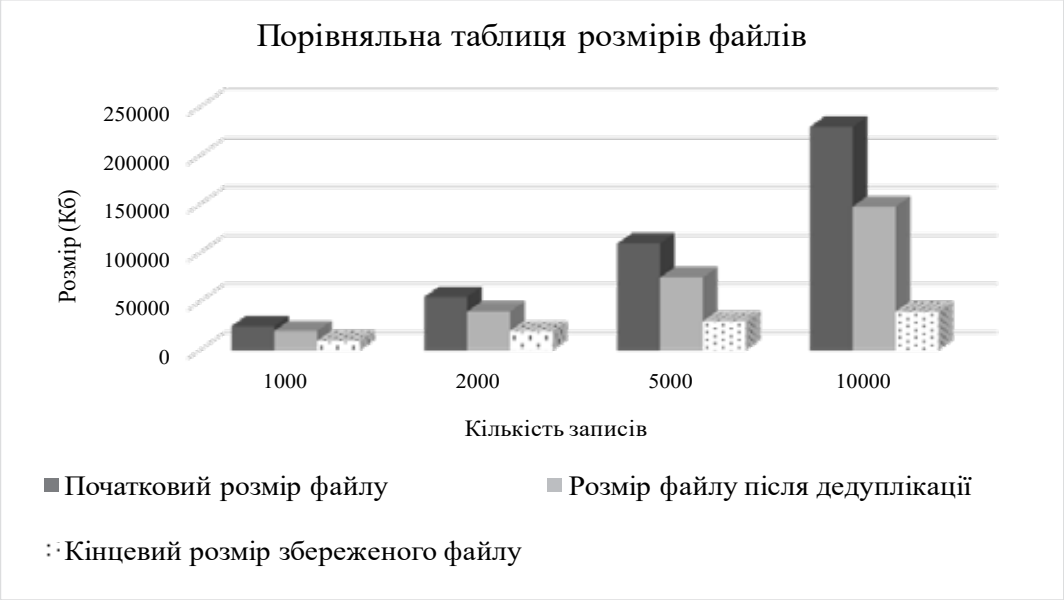

Рис. 3. Порівняння за розміром файлу

Рисунок 4 ілюструє підвищення ефективності зберігання даних після застосування дедуплікації та стиснення і показують, що запропоновані методи не тільки звільняють значний простір для зберігання, але й подвоюють економію, досягнуту лише завдяки дедуплікації. Загальний час обробки залежить як від розміру файлу, так і від кількості записів, що обробляються.

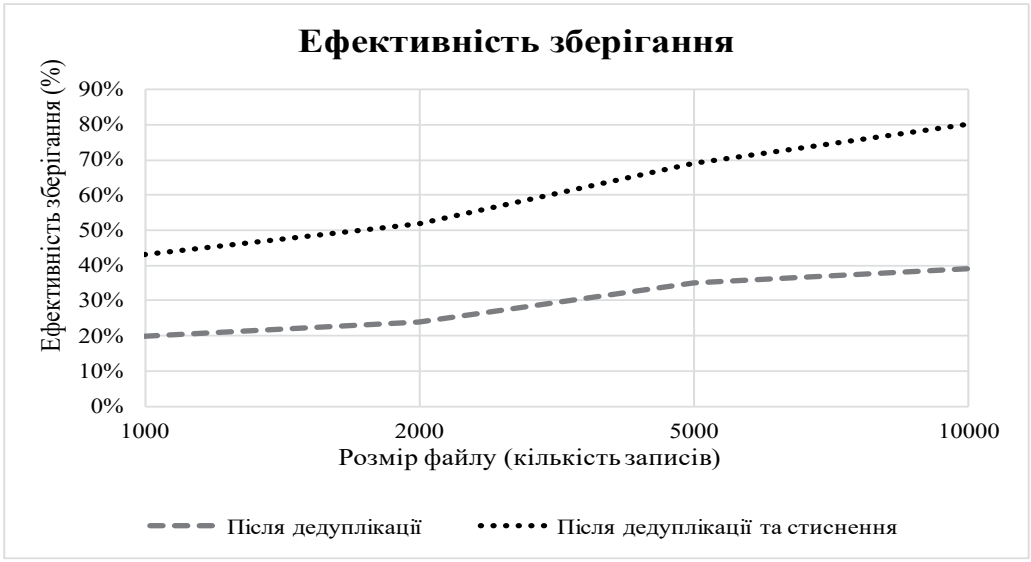

Рис. 4. Порівняння ефективності зберігання

На рисунку 5 показано відсоток ефективності зберігання для різних файлів баз даних з різною складністю кортежів (кількістю полів).

Експерименти, що проведені на файлах з різним структурним складом, показують кращу продуктивність зі збільшенням кількості структурної інформації. Для наборів даних зі складнішою структурою ефективність зберігання може досягати 90%, як показано на рисунку 5.

При реплікації даних між кількома вузлами запропонована технологія допомагає зменшити споживання пропускної здатності за рахунок стиснення файлів та усунення надлишкових даних. Це призводить до більш ефективного використання пропускної здатності, оскільки тільки необхідні дані

передаються на вимогу. На рисунку 6 графічно зображено економію пропускної здатності завдяки використанню запропонованої технології.

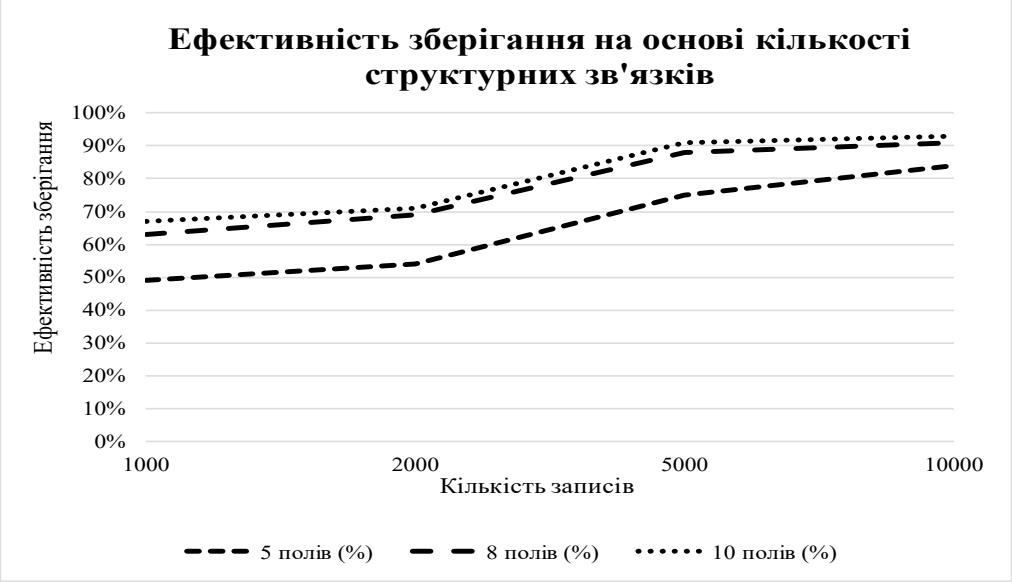

Рис. 5. Ефективність зберігання файлів з різною структурною інформацією

Рис. 6. Пропускна здатність під час міжвузлової передачі

Час, необхідний для передачі файлу від одного вузла до іншого, вимірюється в секундах, і експерименти показують, що час, який витрачається на передачу файлу після застосування запропонованої технології, значно менший порівняно з передачею файлу такого ж початкового розміру.

Висновки. При роботі No-SQL баз даних постійно виникає дублювання, і якщо не вирішувати цю проблему, то це може призвести до неефективного використання дискового простору та мережевих ресурсів. Запропонована технологія забезпечує ефективне вирішення цієї проблеми, використовуючи структурну інформацію для зменшення обсягу цих даних за рахунок дедуплікації і стиснення. Використовуючи комбінацію дедуплікації та стиснення пропонується дворівневий підхід до оптимізації зберігання даних, який легко інтегрується з існуючими інструментами резервного копіювання, надаючи організаціям простий спосіб підвищити ефективність зберігання даних без необхідності суттєвої перебудови інфраструктури управління даними.

Хоча поточна реалізація зосереджена на текстових даних, існує значний потенціал для використання цих методів на ширший спектр типів даних. Наприклад, застосування дедуплікації та стиснення до мультимедійних файлів, таких як зображення, відео та аудіо, або до потоків даних у реальному часі може ще більше підвищити економію місця на сховищах. Такі розширення можуть бути цінними в галузях, які обробляють величезні обсяги таких даних, зокрема в медіа, телекомунікаціях та додатках Інтернету речей. Крім того, застосування цих методів в середовищах реального часу, де дані повинні оброблятися і передаватися безперервно, може запропонувати значні переваги з точки зору ефективності зберігання і обробки.

У майбутньому планується дослідити аналітичні інструменти, такі як pig, hive та інші, для прогнозування розміру, структури та закономірностей даних. Ці прогнози можуть дозволити більш цілеспрямовано застосовувати методи дедуплікації та стиснення. Це підвищить ефективність зберігання за рахунок оптимізації того, коли і де ці методи використовуються. За наявності прогностичних моделей запропонована технологія може перетворитися на повноцінну динамічну систему, здатну коригувати свої стратегії дедуплікації та стиснення на основі аналізу вхідних даних у реальному часі.

Запропонована технологія є кроком вперед у підвищенні ефективності зберігання баз даних NoSQL. Поєднання дедуплікації та стиснення пропонує гнучке, масштабоване рішення, яке може адаптуватися до зростаючих потреб сучасних індустрій, керованих даними. Так як обсяги даних продовжують зростати, а вимоги до їх зберігання стають все більш вираженими, то фреймворки, які використовують нові технології оптимізації збереження даних, будуть відігравати важливу роль в управлінні ландшафтом даних, гарантуючи, що організації зможуть зберігати і передавати свої дані в економічно ефективний, результативний спосіб.

Список використаних джерел:

1. Roy-Hubara N., Sturm A. Design methods for the new database era: A systematic literature review. *Software and Systems Modeling,* 2019. № 19, pp. 297–312. doi:10.1007/s10270-019-00739-8.

2. Ramzan S., Bajwa I. S., Kazmi R., Amna. Challenges in NoSQL-based distributed data storage: A systematic literature review. *Electronics,* 2019. № 8, pp. 1–29. doi:10.3390/electronics8050488.

3. Kim W., Lee I. Survey on data deduplication in cloud storage environments. *Journal of Information Processing Systems*, 2021. № 17(3), pp. 658–673. doi:10.3745/JIPS.03.0160.

4. Kumar N., Shobha, Jain S. C. efficient data deduplication for big data storage systems. In *Progress in Advanced Computing and Intelligent Engineering*. 2019. № 714, pp. 351–371. 10.1007/978-981-13-0224-4_32

5. Wang C., Fu Y., Yan J., Wu X., Zhang Y., Xia H., Yuan Y. A cost-efficient resemblance detection scheme for post-deduplication delta compression in backup systems. *Concurrency and Computation: Practice and Experience.* 2022. № 34(3), pp. e6558. doi:10.1002/cpe.6558.

6. Zhang D., Le J., Mu N., Wu J., Liao X. Secure and Efficient data deduplication in JointCloud storage. *IEEE Transactions on Cloud Computing*. 2023. № 11(1), pp. 156–167. doi: 10.1109/TCC.2021.3081702.

7. Tan H., Zou X., Wan B., Gu Z., Xia W. SuperDelta: Multiple referenced base chunks scheme for fine-grained deduplication backup storage system. *Data Compression Conference Proceedings*. 2024. pp. 362–371. doi:10.1109/DCC58796.2024.00044.

8. Ge X., Zhou C. A data allocation strategy for deduplication backup systems in disk arrays. *Proceedings of SPIE – The International Society for Optical Engineering*. 2024. pp. 1325004. doi:10.1117/12.3038451

9. Zhang D., Deng Y., Zhou Y., Li J., Zhu W., Min G. MGRM: A multi-segment greedy rewriting method to alleviate data fragmentation in deduplication-based cloud backup systems. *IEEE Transactions on Cloud Computing*. 2023. № 11(3), pp. 2503–2516. doi:10.1109/TCC.2022.3214816

10. Koushik C. S. N., Choubey S. B., Choubey A., Sinha G. R. Data deduplication for cloud storage. In *Data Deduplication Approaches*. 2021. pp. 307–317. doi:10.1016/b978-0-12-823395-5.00010-0.