

Ці методи можна використовувати окремо чи в комбінації для досягнення оптимальної продуктивності під час роботи з великими обсягами даних [3, 4, 5].

Існує достатня кількість методів оптимізації великих даних, головне – це вміти правильно їх поєднати. Адже саме правильно підібрані методи оптимізації для потоку Big Data дають можливість передавати, зберігати та користуватись великими даними.

Список використаних джерел

1. Wikipedia. URL: https://en.wikipedia.org/wiki/Big_data. (дата звернення: 27.11.2023).

2. Gagroup. URL: <https://qagroup.com.ua/publications/shcho-take-big-data-i-iaak-tse-pratciuiie/> (дата звернення: 27.11.2023).

3. Aiconference. URL: <https://aiconference.com.ua/uk/news/tehnologii-big-data-klyuchevie-harakteristiki-osobennosti-i-preimushchestva-97883> (дата звернення: 27.11.2023).

4. Степанюк О. С., Січко Т. В. Особливості використання реляційних та нереляційних баз даних в Big Data. *Комп'ютерні технології обробки даних: матеріали всеукр. наук.-практ. конф., м. Вінниця, 2020. С. 103–106.*

5. Ткачук Н. О., Січко Т. В. Застосування Big Data у бізнесі. *Комп'ютерні технології обробки даних: матеріали всеукр. наук.-практ. конф., м. Вінниця, 2022. С. 224–226.*

УДК 004.658.2

*Проценко А. С., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,
Гончар В. М., асистент кафедри інформаційних технологій*

СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ REDIS: ПЕРЕВАГИ ТА НЕДОЛІКИ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі, де обробка великого обсягу даних є ключовим аспектом для багатьох галузей, системи управління базами даних стають невід'ємною частиною розробки програмного забезпечення. Однією з таких систем є Redis, яка привертає увагу своєю унікальною архітектурою та функціональністю.

Далі розглянемо основну характеристику, переваги й недоліки цієї СУБД. Також намагатимемося зрозуміти, у яких сценаріях використання Redis може бути найефективнішим і чи відповідає вона вимогам різноманітних завдань, які постають перед розробниками сучасного програмного забезпечення.

Redis (Remote Dictionary Server) – відкрите програмне забезпечення, основна особливість якого полягає в збереженні даних у формі пар ключ-значення, які зберігаються в оперативній пам'яті, з можливістю забезпечувати довговічність зберігання на бажання користувача. Ця СУБД з відкритим початковим кодом та написана на ANSI C.

Переваги:

➤ **Висока швидкість** – одна з ключових переваг Redis, що досягається завдяки використанню оперативної пам'яті для зберігання даних. Якщо в довговічності даних немає потреби, Redis може пропонувати значно кращі результати, порівняно з СУБД, які зберігають кожен зміну на диск перед завершенням транзакції. Також Redis може ефективно виконувати операції читання та запису без помітної різниці у швидкості. До прикладу, під час тестування на сервері з CPU Xeon X3320 2,5 ГГц вдалося забезпечити 110 000 операцій запису і 81 000 операцій читання за секунду. Це робить Redis відмінним вибором для завдань, де вимагається максимальна продуктивність і миттєвий доступ до даних, як-от системи кешування та інші сценарії реального часу.

➤ **Модель даних у Redis** – це асоціативний масив, у якому ключі відображаються в значення. Основна відмінність між Redis та іншими базами такого типу в тому, що значення словника не обмежені рядковими типами. На додачу до рядків підтримуються такі абстрактні типи даних: списки рядків, множини рядків (непорядкований набір неповторюваних елементів), впорядковані множини рядків (набори неповторюваних елементів впорядкованих за пов'язаним значенням з рухомою комою), хеші, в яких ключі і значення є рядками. Тип значення визначає, які операції можна виконувати з цим значенням. У Redis доступні високорівневі атомарні операції з боку, як-от перетин, об'єднання та різниця між множинами та списками.

➤ **Потужні механізми реплікації та розподіленості** визначають Redis як ефективний інструмент для масштабування й забезпечення надійності. Застосовуючи модель «master-slave» для реплікації, Redis дає змогу створювати дублікати даних з будь-якого сервера Redis довільну кількість разів. Цей підхід особливо корисний для масштабування читання (але не запису) чи надлишковості даних. Версія Redis 3.0 та пізніші включають у себе Redis Cluster, що розширює можливості системи для створення розподілених сховищ. Він дає змогу автоматично розподіляти дані між численними вузлами, сприяючи створенню відмовостійких конфігурацій, де дані дублюються на різних вузлах. Це гарантує неперервну доступність навіть у випадку відмови одного з вузлів, не призводячи до зупинки системи загалом [1].

➤ **Легкість конфігурації та швидкість інсталяції.** Процес встановлення Redis вражає простотою, даючи змогу швидко розпочати роботу з цією СУБД.

До того ж Redis відома своєю легкістю налаштування, що робить її привабливою для розробників, які цінують ефективність та мінімальний час витрачений на налагодження.

Недоліки:

➤ **Обмеженість об'ємом пам'яті** є суттєвим недоліком Redis. Оскільки вона зберігає всі дані в оперативній пам'яті, об'єм доступної пам'яті може бути обмеженим. Це обмеження призводить до ситуації, коли Redis стає менш придатним для роботи, де зберігання всієї інформації в оперативній пам'яті стає витратним та неефективним. Такий аспект може потребувати ретельного планування та оптимізації використання ресурсів для досягнення найкращої продуктивності системи.

➤ **Обмежена здатність виконувати складні запити.** Орієнтований на швидкодію та простоту операцій із даними, Redis може виявитися менш ефективним під час обробки запитів, які вимагають складних операцій чи аналізу даних за складними критеріями. Це обмеження може становити проблему для проєктів, де важлива гнучкість і розширена функціональність обробки даних. Під час вибору інструментів розробники повинні враховувати цю особливість Redis і вибирати його для завдань, де простота та швидкодія важливіші, або доповнювати його іншими рішеннями у випадках, коли потрібно виконати складні операції.

➤ **Втрата даних** у разі вимкнення системи. Навіть з урахуванням наявності механізму реплікації, який дає змогу створювати копії даних на різних серверах, існує ризик втрати інформації у випадку, якщо не всі репліки успішно синхронізовані перед вимкненням системи. Це може виникнути у випадку непередбачуваних помилок або затримок у процесі реплікації. Цю потенційну вразливість важливо враховувати під час проєктування системи та розробки стратегій резервного копіювання для запобігання втрати даних у випадку аварійних ситуацій або вимушеного вимкнення системи.

➤ **Обмежені можливості безпеки** можуть призвести до потенційних проблем під час встановлення відкритого доступу до сервера. Redis не завжди надає розширені засоби для безпеки даних, тому важливо усвідомлювати це і вживати відповідні заходи, як-от використання автентифікації для захисту інформації. У великих або критичних проєктах треба уважно розглядати додаткові заходи безпеки та налаштування системи для мінімізації можливих ризиків [2, 3].

Отже, Redis є потужною системою управління базами даних з унікальною архітектурою та низкою переваг, які визначають її популярність серед розробників. Висока швидкість завдяки використанню оперативної пам'яті, гнучка модель даних, легкість конфігурації, а також механізми реплікації та розподіленості роблять Redis ідеальним вибором для сценаріїв, де важлива продуктивність і надійність. Однак існують певні недоліки, як-от обмеженість об'ємом

пам'яті, виконання складних запитів, ризик втрати даних під час вимкнення системи. Ці обмеження важливо враховувати під час вибору Redis для конкретного проєкту та вживати відповідних заходів для оптимізації та захисту від потенційних проблем.

Список використаних джерел

1. Redis. URL: <https://en.wikipedia.org/wiki/Redis> (дата звернення: 20.11.2023).
2. Redis на практичних прикладах. URL: <https://devzone.org.ua/post/redis-na-praktichnikh-prikladakh> (дата звернення: 22.11.2023).
3. Introduction to Redis. URL: <https://redis.io/docs/about/> (дата звернення: 22.11.2023).

УДК 004.65:004.056.5

*Шевцов М. В., здобувач 2 курсу спеціальності 122 Комп'ютерні науки,
Гончар В. М., асистент кафедри інформаційних технологій*

ВИКОРИСТАННЯ КРИПТОГРАФІЇ ДЛЯ ЗАХИСТУ ДАНИХ У БАЗІ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі, коли інформація є найціннішим ресурсом, забезпечення безпеки даних стає критично важливим завданням для багатьох організацій. Паролі користувача, цифрові підписи та інша конфіденційна інформація може бути злита в мережу за будь-яких обставин – протиправних дій зловмисників чи через просту несправність сервера. Тому зберігати цю інформацію в базах даних треба у зашифрованому форматі. До того ж необхідно, щоб із шифру не можна було отримати ключ у тому вигляді, в якому його вводив користувач. Саме інтеграція криптографічних методів та хеш-функцій стає ключовим аспектом забезпечення конфіденційності та цілісності даних.

Найвідомішими криптографічними алгоритмами для захисту інформації, які використовуються зокрема і в базах даних, є:

➤ AES (Advanced Encryption Standard) – це симетричний блоковий метод, який використовує ключ для шифрування та розшифрування даних в блоках фіксованого розміру. Він забезпечує доволі високий рівень безпеки і широко використовується для шифрування даних у реляційних базах даних [1].

➤ RSA (Rivest–Shamir–Adleman) – це асиметричний алгоритм, який використовує пару ключів для шифрування та розшифрування даних. Публічний ключ