

Оптимізація запитів до баз даних у веб-додатках

Стан проблеми. Сучасні веб-додатки потребують високої продуктивності для обробки великих обсягів даних у реальному часі, що стає ще складнішим у міру збільшення кількості користувачів та оброблюваних запитів.

При роботі з різними типами баз даних застосовуються різні стратегії оптимізації. Для реляційних БД оптимізація може включати індексацію даних, нормалізацію таблиць та оптимізацію складних SQL-запитів. Натомість для нереляційних БД часто використовують шардинг для горизонтального масштабування, кешування для зниження навантаження на БД, а також розподіл даних між кількома серверами. Правильне застосування цих технік дозволяє суттєво покращити продуктивність веб-додатків, залежно від типу використовуваної бази даних та специфіки завдань.

Постановка задачі. Необхідно провести аналіз продуктивності веб-додатків, які працюють на різних типах БД, і визначити оптимальні підходи для підвищення їх продуктивності. Особливо важливо розуміти, як реляційні та нереляційні БД впливають на швидкість обробки даних і загальну ефективність додатків.

Розв'язання задачі. Для вирішення було розроблено та протестовано два додатки з однаковим функціоналом, але з використанням різних типів баз даних: реляційної (MySQL) та нереляційної (MongoDB). Було застосовано різні підходи до оптимізації, щоб визначити найефективніші стратегії для кожного типу бази даних. Зокрема, використано такі методи:

- **Індексування:** Індексування є одним з ключових методів підвищення продуктивності запитів до бази даних. За допомогою індексів база даних може швидше знаходити потрібні записи, зменшуючи час пошуку та обробки інформації.

До оптимізації: Запит SELECT на таблиці, що містить 1 млн записів без індексу, займав у середньому 1200 мілісекунд для виконання.

Після оптимізації: Після додавання індексу на поле пошуку час виконання скоротився до 150 мілісекунд. Отже, час виконання запиту скоротився у **8 разів**.

- **Кешування:** це стратегія збереження результатів запитів до бази даних в пам'яті для повторного використання. Це дозволяє зменшити кількість запитів до бази та прискорити доступ до результатів.

До оптимізації: При 1000 одночасних запитах до бази даних час відповіді становив у середньому 800 мілісекунд.

Після оптимізації: Після впровадження кешування найчастіших запитів час відповіді скоротився до 100 мілісекунд, оскільки повторні запити оброблялись з кешу, при чому прискорення становить 87.5%

- **Комбінована стратегія (індексування та кешування):** Поєднання індексування та кешування дає змогу досягти максимальних показників продуктивності у веб-додатках, зокрема у випадках великого обсягу запитів та даних.

До оптимізації середній час обробки складного запиту на базі даних, що містить 5 млн записів, без індексів та кешування складав 2,5 секунди.

Після додавання індексів та кешування частих запитів середній час виконання скоротився до 300 мілісекунд, при чому спостерігається прискорення в 8 разів.

- **Денормалізація даних:** це ще один метод підвищення продуктивності запитів до бази даних, який полягає у зміні структури бази даних таким чином, щоб зменшити кількість JOIN-запитів між таблицями. Цей підхід може зменшити кількість запитів і збільшити швидкість виконання, особливо в системах, де виконуються часті складні запити з великою кількістю зв'язків між таблицями.

До оптимізації запит, який виконувався на п'яти таблицях за допомогою операції JOIN, займав 2500 мілісекунд. Після денормалізації даних і скорочення кількості JOIN-запитів час виконання скоротився до 700 мілісекунд. При цьому затримка відгуку скорочена на 72%.

- **Аналіз частоти запитів і створення кешу на основі популярності:**

Для досягнення максимальної ефективності важливо аналізувати, які запити до бази даних є найчастішими, і кешувати саме їх результати. Цей підхід дозволяє зменшити навантаження на базу даних за рахунок збереження найбільш популярних запитів у кеші.

До оптимізації запити до бази даних виконувались без урахування їх частоти, середній час відповіді складав 1200 мілісекунд. Після оптимізації після впровадження кешування популярних запитів середній час відповіді скоротився до 300 мілісекунд. При кешуванні запитів з урахуванням частоти їх використання час відповіді скоротився на 75%

Висновки. Оптимізація веб-додатків вимагає правильного вибору бази даних, оскільки це безпосередньо впливає на продуктивність. Реляційні бази даних краще підходять для транзакційних систем завдяки своїй структурованості та надійності. Нереляційні бази даних, в свою чергу, є оптимальними для роботи з великими обсягами неструктурованих даних, завдяки гнучкості та масштабованості. Застосування стратегій оптимізації, таких як індексація, кешування та денормалізація, дозволяє суттєво підвищити

швидкість обробки запитів і загальну ефективність веб-додатків. Правильне налаштування бази даних відповідно до специфіки завдань є ключем до досягнення високих результатів..

Література

1. Date, C. J. (2012). *Database Design and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media.
2. Strauch, C. (2011). *NoSQL Databases: An Overview*. *Journal of Database Management*, 22(4), 15-34.
3. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems*. Pearson.
4. Sadalage, P. J., & Fowler, M. (2013). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley.
5. Chen, L. (2017). *Database Systems: The Complete Book*. Prentice Hall.