

Міністерство освіти і науки України

**Одеський національний технологічний університет
Вінницький національний технічний університет
Інститут комп'ютерної інженерії, автоматизації,
робототехніки та програмування ім.П.Н.Платонова**



МАТЕРІАЛИ

**IV ВСЕУКРАЇНСЬКОЇ
НАУКОВО – ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ
МОЛОДИХ ВЧЕНИХ, АСПІРАНТІВ
ТА СТУДЕНТІВ**

**«КОМП'ЮТЕРНІ ІГРИ І МУЛЬТИМЕДІА
ЯК ІННОВАЦІЙНИЙ ПІДХІД
ДО КОМУНІКАЦІЇ - 2024»**

**26-27 вересня 2024 р.
ОДЕСА**

ПРЕЗИДІЯ ТА ОРГКОМІТЕТ КОНФЕРЕНЦІЇ

ГОЛОВА ПРЕЗИДІЇ

Богдан Єгоров, Президент ОНТУ, академік НААН України, д.т.н., професор

ЧЛЕНИ ПРЕЗИДІЇ

Надія Дец, к.т.н., доцент, в.о.ректора Одеського національного технологічного університету

Ольга Ольшевська, к.т.н., доцент, проректор з наукової роботи і міжнародних зв'язків Одеського національного технологічного університету.

ГОЛОВА ОРГКОМІТЕТУ

Сергій Котлик, к.т.н., доц. каф. Інформаційних технологій і кібербезпеки, ОНТУ

ЗАСТУПНИК ГОЛОВИ ОРГКОМІТЕТУ

Сергій Шестопалов, к.т.н., доц., каф. Комп'ютерної інженерії, ОНТУ

ЧЛЕНИ ОРГКОМІТЕТУ

Олексій Ізвалов, регіональний координатор Global Game Jam в Східній Європі, ETI ім.Ельворті,

Сергій Артеменко, зав.каф. Комп'ютерної інженерії, ОНТУ,

Михайло Кисленко, Unity Developer, DAL'S Games,

Олександр Романюк, зав.каф. Програмного забезпечення, ВНТУ,

Ольга Чолишкіна, директор Інституту комп'ютерно-інформаційних технологій і дизайну, МАУП,

Олександр Терьошин, Unity 3d developer, BlueGoji,

Павло Івасюк, Senior Snapchat JS Developer, BeVisioned,

Петро Горват, зав.каф. Комп'ютерних систем і мереж, ДВНЗ "Ужгородський національний університет".

МЕТОДИ ОПТИМІЗАЦІЇ ЗБЕРІГАННЯ ВЕЛИКИХ МАСИВІВ ДАНИХ У БАЗАХ ДАНИХ

ТЕРЕШКО Д. С. (dimon2004105@gmail.com), БАБЮК Н. П. (babiuk@vntu.edu.ua)
Вінницький національний технічний університет

Розглянуто проблематику зберігання та доступу до великих масивів структурованих даних, запропоновано ряд можливих рішень.

Вступ

Активний розвиток та поширення інформаційних технологій створює нові виклики для розробників програмних систем, адже із збільшенням навантаження на застосунки, збільшення обсягів оброблюваних даних виникають проблеми як з обробкою даних, так і з їх збереженням. Разом із дослідженням та впровадженням способів масштабування розподілених застосунків для ефективної обробки даних, створюються нові методи та покращуються існуючі рішення для зберігання інформації.

Розвиток процесорів сповільнився через існуючі технології виготовлення мікросхем на кремнії, фокус досліджень способів покращення швидкодії та доступності змістився до паралелізму – використання кількох процесорних ядер, кількох процесорів, або навіть кількох комп'ютерів, об'єднаних у кластер [1].

Напрямок паралелізму також притаманний для систем збереження та керування даними, адже швидкодія дискових накопичувачів також є обмежена, що проявляється при великих обсягах даних, і, відповідно, призводить до збільшення часу відповіді та зменшення доступності системи [2].

Проблематика та існуючі способи вирішення

З ростом обсягів даних, що необхідно обробляти, зберігати та надавати до них доступ, виникає низка проблем, пов'язаних із швидкістю, адже ресурси є обмеженими.

Від швидкості обробки запитів залежить не лише зручність користувачів, а й загальна продуктивність систем. Зростання обсягів даних, складність запитів та вимоги користувачів до швидкості доступу до інформації вимагають постійної оптимізації баз даних. Повільна робота бази даних може призвести до, наприклад, зниження ефективності бізнес-процесів та втрати клієнтів.

Серед основних методів оптимізації баз даних можна виділити такі [3]:

- Нормалізація – передбачає усунення надмірності та забезпечення цілісності даних, часто використовується для створення логічно правильної структури бази даних. Однак, надмірна нормалізація може призвести до збільшення кількості таблиць та запитів, що, в свою чергу, знижує продуктивність.

- Денормалізація – передбачає повторне зберігання даних у різних таблицях, щоб зменшити кількість запитів, необхідних для отримання необхідної інформації, може порушити цілісність даних і ускладнити підтримку бази.

- Додавання індексів – дозволяє пришвидшити пошук, замість того, щоб проводити повне сканування, за умови правильного вибору типу індексу у залежності від параметрів запиту та типу даних поля.

Крім цього, для підвищення продуктивності баз даних використовуються такі методи як кешування та створення матеріалізованих переглядів (materialized view). Кешування дозволяє зберігати результати часто використовуваних запитів у пам'яті, що значно прискорює їх виконання. Створення матеріалізованих переглядів (materialized view) дозволяє зберігати результати складних запитів у вигляді окремих таблиць, що може значно прискорити їх повторне використання.

Існує два основних шляхи для збільшення обчислювальних потужностей [4]: вертикальне та горизонтальне масштабування. Суть вертикального масштабування полягає у заміні комп'ютера на більш потужний. Натомість, горизонтальне масштабування полягає у поділі програмної системи на частини, які запускатимуться на окремих машинах.

Вищезгадані способи також є актуальними і для баз даних. Вертикальне масштабування передбачає збільшення обчислювальних ресурсів однієї машини, горизонтальне масштабування або шардинг – розподілення даних та навантаження на декілька менших машин, воно дозволяє

плавно збільшувати потужність системи шляхом додавання нових машин, що робить його більш масштабованим і економічно ефективним у довгостроковій перспективі.

Розбиття даних (шардинг) [2] – це техніка, яка полягає у поділі великої таблиці або набору даних на менші, більш керовані частини, звані секціями.

Метою розбиття є рівномірний розподіл даних навантаження запитів між декількома машинами, при цьому уникаючи так званих «гарячих точок» – вузлів з непропорційно високим навантаженням). Для цього необхідно вибрати схему розбиття, яка підходить для даних, і повторно балансувати секції, коли вузли додаються або видаляються з кластера.

Основними підходами розбиття є [2]:

- Розбиття за діапазоном ключів – ключі сортуються, і секція володіє ключами від певного мінімуму до певного максимуму. Сортування має перевагу в тому, що можливі ефективні запити за діапазоном, але існує ризик «гарячих точок» у випадках, коли запит здійснюється до ключів, які розташовані близько один до одного в відсортованому порядку. У цьому підході секції зазвичай перебалансуються динамічно шляхом розділення діапазону на два піддіапазони, коли секція стає занадто великою.

- Хеш-розбиття – застосовується хеш-функція до кожного ключа, і секція володіє діапазоном хешів. Цей метод порушує порядок ключів, що робить неефективними запити за діапазоном, але цим може більш рівномірно розподіляти навантаження. При хеш-розбитті зазвичай створюється фіксована кількість секцій, присвоюється кілька секцій кожному вузлу і цілі секції переміщуються з одного вузла на інший, коли вузли додаються або видаляються.

Можливі також гібридні підходи, наприклад, зі складовим ключем: використання однієї частини ключа для ідентифікації секції та іншої частини для порядку сортування.

Важливо розуміти, що розбиття вимагає ретельного планування та аналізу сценаріїв використання бази даних, адже неправильне розбиття може призвести до проблем з продуктивністю та складності управління даними.

Висновки

Користувачі сучасних програмних систем очікують миттєвої реакції на свої дії. Щоб задовольнити ці очікування, необхідно забезпечити високу швидкість роботи системи. Вибір правильного типу організації даних залежно від потреб може зробити процес збереження та доступу до даних ефективнішим, що, у свою чергу, веде до зменшення часу відгуку системи на запити користувачів, підвищення її доступності та надійності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Single-Chip Processors Have Reached Their Limits [Електронний ресурс]. Режим доступу: <https://spectrum.ieee.org/single-chip-processors-have-reached-their-limits>
2. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol: O'Reilly Media Inc, 2017. 590 p.
3. Richard Niemiec. Oracle Database 12c Release 2 Performance Tuning Tips & Techniques (Oracle Press). New York: McGraw Hill, 2017. 1136 p.
4. A Guide to Horizontal vs Vertical Scaling [Електронний ресурс]. Режим доступу: <https://www.mongodb.com/resources/basics/horizontal-vs-vertical-scaling>

УДК 37.091.33:004.8

РОЗШИРЕННЯ МОЖЛИВОСТЕЙ ОСВІТИ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ
УМАНЕЦЬ В.О., РОЗПУТНЯ Б.М. (b.rozputnia@vspu.edu.ua, umanets@vspu.edu.ua)
Вінницький державний педагогічний університет імені Михайла Коцюбинського

Сучасні освітні системи потребують персоналізації для задоволення різноманітних потреб здобувачів освіти, зокрема навчальних, пізнавальних, мотиваційних та індивідуальних. Штучний інтелект пропонує інноваційні рішення, дозволяючи адаптувати навчальний процес під індивідуальні особливості кожного.