

МЕТОД КЕШУВАННЯ ІНДЕКСІВ ПРИ ОПТИМІЗАЦІЇ ПОШУКУ В БАЗАХ ДАНИХ

Анотація: Запропоновано новий метод оптимізації пошуку інформації в базах даних. Проведено експериментальне дослідження збільшення швидкості запису пошуку інформації та виконано порівняння швидкості пошуку з класичними методами, що використовують алгоритми завантаження індексів в оперативну пам'ять СУБД.

Ключові слова: пошук даних, індекси, оперативна пам'ять, кешування.

Вступ

Швидкість пошуку інформації є ключовою характеристикою в сучасних СУБД та пошукових системах. Використання індексів при кожному зверненні до даних призводить до збільшення швидкості пошуку інформації, адже індекси завантажуються в оперативну пам'ять та зберігаються там доти, поки їх не відтіснять інші індекси [1,2]. Аналогічні процеси повторюються із даними, які після опрацювання запитів також завантажуються в пам'ять та знаходяться там до витіснення їх іншими даними. Така система дозволяє пришвидшити пошук інформації та обробляти її зазвичай в ОП, адже обробка даних в ОП на кілька порядків швидше, чим при зчитуванні з дискового простору [3,4,5].

Постановка задачі

Необхідно створити новий метод підвищення швидкості пошуку інформації в базах даних, який б дозволив завантажувати в оперативну пам'ять тільки індекси, кількість використань яких при виконанні запитів за час збору статистики найбільша, а по іншим проводити пошук тільки завантаживши їх з диску.

Класичний алгоритм завантаження індексів в оперативну пам'ять СУБД

Пошук в СУБД проходить в кількох напрямках, спочатку проводиться аналіз даних, які знаходяться в оперативній пам'ять, а потім вже інформація зчитується з дискового простору.

Приведемо класичний алгоритм завантаження індексів, що використовуються у запитах, в оперативну пам'ять СУБД:

1. Після отримання запиту СУБД визначає список індексів, які потрібні для пошуку відповідних даних.
2. Загружає відповідні індекси в оперативну пам'ять.

3. Після обробки наступного запиту СУБД перевіряє, чи індекс знаходиться в ОП. Якщо індекс знаходиться, то провидить пошук по ньому і він переміщається на перше місце в списку індексів ОП, а якщо ні – то загрузає його з індексного файлу диску та розміщає його на перше місце в списку, а решту зсуває на рівень нижче.
4. Вразі, якщо відсутня пам'ять для завантаження наступного індексу, то СУБД видаляє з пам'яті індекс, який займає остання місце у списку індексів.

Даний алгоритм має свої переваги та недоліки. Серед переваг є те, що він дуже простий в використанні та дозволяє зберігати індекси в оперативній пам'яті. Але недоліком його є те, що в пам'ять попадають індекси, які використовуються надзвичайно рідко та займають там місце, аж поки їх не відтіснять інші.

Алгоритм завантаження індексів в оперативну пам'ять СУБД на основі методу кешування індексів

Потрібно створити метод, який би дозволяв завантажувати в оперативну пам'ять СУБД тільки ті індекси, які використовуються найчастіше.

Алгоритм завантаження індексів в оперативну пам'ять СУБД на основі методу кешування індексів (МКІ):

1. Для збору статистики по кількості використань індексів за період ΔT СУБД завантажує індекси в ОП за класичним алгоритмом.
2. Після збору статистики за відповідний період СУБД загрузає в ОП ті індекси, які найчастіше використовувались за цей період.
3. Вразі, якщо індекс відсутній в ОП, то пошук по ньому проводиться безпосереднім зчитуванням його вузлів з дискового індексного простору БД.
4. Після проходження відповідного ΔT в оперативну пам'ять завантажуються ті індекси, які використовувались найчастіше при умові, якщо вони не були завантажені раніше.

Такий алгоритм реалізується в два етапи. Перший етап, час якого дорівнює ΔT , застосовує класичний алгоритм, адже потрібен час на збір статистики та визначення індексів, які максимально використовувались за відповідний період. Під час другого етапу в оперативну пам'ять завантажуються ті індекси, які використовувалися найчастіше за попередній ΔT .

Формула для розрахунку:

$$\Delta T = \frac{K_i T}{K},$$

де K_i – кількість індексів, K – кількість використань індексів, а T – час збору статистики.

Експериментальні оцінки параметрів методів запису та пошуку даних

Період збору статистики визначається експериментально для відповідної СУБД. Наприклад, візьмемо базу студентів НТУУ “КПІ”, яка розміщена на сервері СУБД Oracle 10g та проведемо розрахунки для отримання ΔT .

Створимо таблицю, в яку збережемо статистику використання індексів за відповідний період:

```
create table назва_таблиці as
select vplan.object_name, vplan.timestamp, v.module, di.table_name
from dba_indexes di, v$sql_plan_statistics_all vplan
join v$sql v using (sql_id, child_number)
where vplan.operation = 'INDEX' and
      vplan.object_name=di.index_name and
      di.table_type='TABLE' and
      vplan.timestamp>'початок_періоду' and
      vplan.timestamp>'кінець_періоду'
```

Відповідно ми отримали таблицю з даними для аналізу виконань індексів в нашій базі даних.

Як показав експеримент, ми отримали наступні числа:

$K_i = 170$ шт., $K = 4072$ шт., $T = 8$ годин.

Відповідно:

$$\Delta T = 1202c \approx 20xв$$

Отже, для нашої бази даних період, за який повинна збиратися статистика буде рівний 20 хв.

Покажемо на графіках статистику використань індексів за 8 робочих годин (рисунок 1) та за кожен годину окремо (рисунок 2).

Як показує рисунок 1, перші 30 індексів використовуються, в нашому випадку, набагато частіше ніж інші, тому саме вони будуть завантажені в оперативну пам'ять СУБД. Крім того визначною характеристикою є кількість оперативної пам'яті, яку виділяє СУБД на зберігання індексів. Як показав розрахунок об'ємів індексів, в середньому в ОП їх зберігалось 27 індексів.

Рисунок 2 показує відношення кількості індексів за відповідний час до кількості використань цих індексів. Окремим висновком є те, що співробітники нашого університету, які працюють з базою даних студентів найбільш інтенсивно звертаються до бази даних після 13 години.

Після аналізу отриманих даних виконується підрахунок кількості завантажень з дискового простору в оперативну пам'ять індексів, які використовуються за відповідну робочу годину.

Рисунок 3 показує, що використання методу кешування індексів ефективно на ділянках в яких кількість використань індексів на порядок більша за кількість індексів, що використовуються.

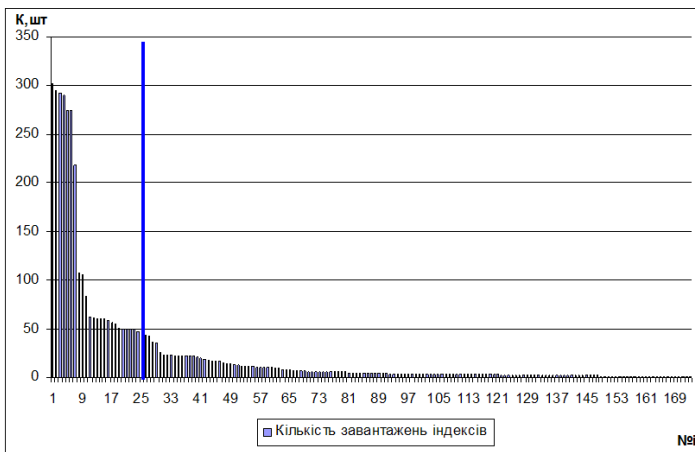


Рис. 1 – Розподіл кількості завантажень відповідного індексу за експериментальний час

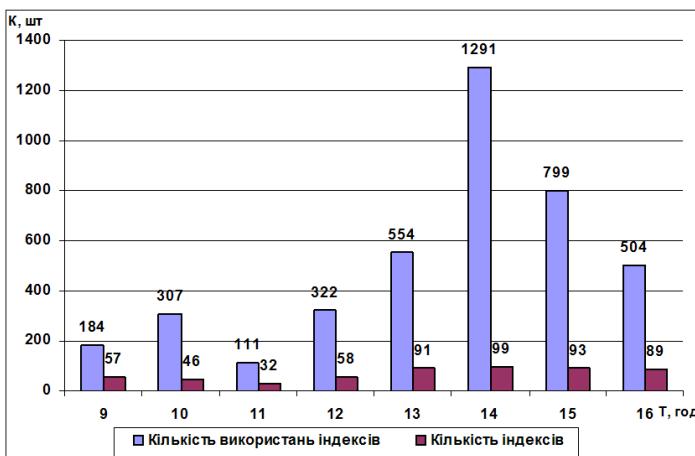


Рис. 2 – Розподіл кількості індексів та кількості використань індексів за кожну годину експериментального часу

Висновки

Пошук на основі алгоритму, який базується на методі кешування індексів доцільно використовувати для баз даних, в яких кількість використань індексів на порядок більша за кількість самих індексів. Швидкість пошуку при застосуванні МКІ збільшується приблизно на 15%. Таким чином, реалізується оптимізація викори-

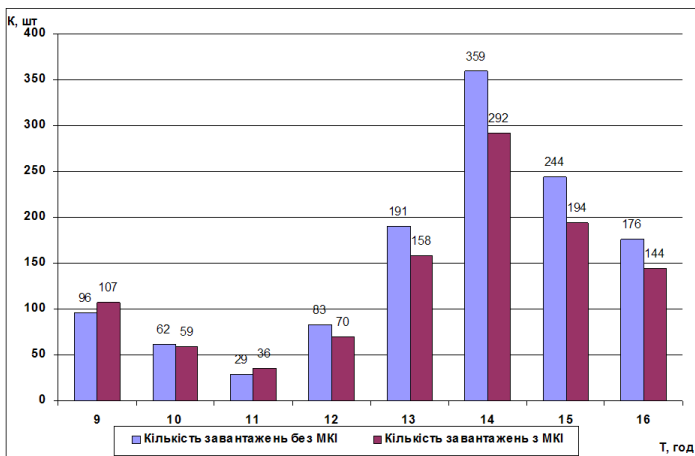


Рис. 3 – Експериментальні дані по кількості завантажень індексів в оперативну пам'ять за кожну годину робочого часу

стання індексів, які зберігаються в оперативній пам'яті та підвищується ефективність системи пошуку в СУБД.

Вибір певного алгоритму завантаження індексів в оперативну пам'ять СУБД залежить від кількості індексів, які створені в таблицях БД та кількості використань цих індексів при проведенні пошуку.

Бібліографічний список

1. Фуфаев Э.В., Фуфаев Д.Э. Базы данных / Академия, 2012. – 320 с.
2. Ицик Бен-Ган Microsoft SQL Server 2008. Основы T-SQL / БХВ-Петербург, 2009. – 430 с.
3. Маркин А. В. Построение запросов и программирование на SQL / Питер Кому, 2008. – 704 с.
4. Дейв Энсор, Йен Стивенсон Oracle. Проектирование баз данных / Лорі, 2006. – 560 с.
5. Каролин Бегг, Томас Коннолли. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Вильямс, 2006. – 532 с.

Отримано 22.10.2013 р.