

ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ УДОСКОНАЛЕННЯ МЕТОДІВ КЕШУВАННЯ У ІНФОРМАЦІЙНИХ СИСТЕМАХ

Вінницький національний технічний університет

Анотація

За результатами проведеного дослідження виявлено можливості удосконалення методів кешування для оптимізації витрат оперативної пам'яті та процесорного часу у високонавантажених інформаційних системах.

Ключові слова: кеш, система кешування, методи кешування, інформаційна система, сервер, база даних.

Abstract

According to the results of the study, the possibility of improving caching methods to optimize the cost of RAM and CPU time in high-load information systems.

Keywords: cache, caching system, caching methods, information system, server, database.

Актуальність обраної тематики обумовлена необхідністю дослідження та на його основі отримання висновків про оптимальність використання методів кешування даних у інформаційних системах.

Одним з основних критеріїв ефективності кешування є відношення кількості оброблених запитів за одиницю часу з використанням системи кешування та кількості оброблених запитів за одиницю часу без використання системи кешування:

$$k_e = \frac{n_1 \times t_2}{n_2 \times t_1}$$

де n_1, n_2 – кількість запитів до інформаційної системи за сесію;

t_1, t_2 – тривалість сесії.

Коефіцієнт ефективності системи кешування може відрізнятися в межах однієї системи в залежності від загального навантаження системи, вільної оперативної пам'яті, часу затримки мережі, вибірки даних. Для отримання репрезентативних результатів ефективності системи кешування необхідно провести дослідження ефективності декілька разів в умовах максимально наближених до реальних, включаючи особливості нерівномірного розподілення трафіку.

$$k_c = \frac{\sum_{i=1}^n k_n}{n}$$

В реальних системах даний коефіцієнт змінюється в залежності від дня тижня та часу, адже коефіцієнт ефективності кешування напряму залежить від навантаження на інформаційну систему. Для порівняння коефіцієнтів ефективності декількох методів кешування потрібно використовувати стабільну систему, та виконувати заміри декілька разів змінюючи кількість запитів та вибірку даних. Для отримання результатів наближених до реальних вибірка даних повинна бути рандомізованою. В кінцевому результаті ефективнішим буде метод кешування при використанні якого було здійснено найменша кількість викликів до бази даних, а пошук результатів запиту потребуватиме мінімум ресурсів.

Іншим важливим критерієм ефективності кешування даних є затрати оперативної пам'яті. У багатьох інформаційних системах розмір бази даних може сягати десятків терабайт. Очевидно що неможливо зберігати у оперативній пам'яті таку кількість даних, тому потрібно фільтрувати які дані повинні затримуватися в кеші, а які ні. Зазвичай дана задача вирішується простим методом зберігання даних у кеші на певний час T , після якого дані видаляються з кешу звільнюючи місце для нових даних. Таким чином дані які запитувалися одноразово будуть видалені і не займатимуть місце в оперативній пам'яті, проте у випадку якщо дані запитуються часто виникне необхідність доступатися до них через

базу даних. Це призведе до додаткових витрат, адже замість того щоб один раз звернутися до бази даних доведеться періодично завантажувати ці дані з бази даних, де період дорівнюватиме T .

Для вирішення даної проблеми можна замінити константний час, після якого дані у кеші видаляються, на період, який буде оновлюватися кожен раз коли ці дані використовуються. Таким чином у кеші не будуть зберігатися дані які запитуються одноразово, а для збереження даних що використовуються часто буде виконуватися лише один виклик бази даних. Однак при використанні такого підходу можливі ситуації коли дані знаходяться у кеші дуже довго залишаючись неактуальними. Наприклад за час поки дані знаходяться у кеші аналогічні дані у базі даних могли бути змінені, або навіть видалені. Метод кешування Read-Through не передбачає механізму який просто вирішував дану проблему. Зазвичай для пом'якшення даної проблеми використовують комбінацію двох підходів, де встановлюють період T_1 , після якого дані видаляються з кешу і тим самим при наступному виклику актуалізуються, та період T_2 , який оновлюється кожен раз коли дані фігурують у запиті, тим самим не створюючи надмірне навантаження на базу даних. При цьому важливо виконувати умову $T_1 > T_2$, інакше сенс використання періоду T_2 втрачається.

Також для вирішення проблеми актуальності даних має сенс удосконалити метод кешування шляхом інвалідації кешованих даних при виконанні операцій над ними. Такими операціями є змінення вмісту даних, або їх видалення. Дану операцію потрібно виконати одразу після внесення змін, тоді при наступному запиті дані будуть отримані з бази даних та знову запишуться у кеш. Таким чином можна гарантувати що у кеші завжди будуть знаходитися актуальні дані.

Однак у високонавантажених інформаційних системах, що оперують великими вибірками даних, перелічених засобів оптимізації вмісту кешу може бути недостатньо. Кожен окремих запит характеризується часом виконання запиту у базі даних та кількістю виконання аналогічних запитів. Метод періодичної очистки кешу забезпечує можливість видаляти з кешу запити одноразові запити, та запити які виконуються досить рідко. Однак ефективно зберігати дані не тільки тих запитів що використовуються часто, а й дані для витягнення яких витрачається багато ресурсів бази даних. Індикаторами таких запитів перш за все є час їх виконання базою даних, також це можуть бути кількість блокувань всередині реляційних таблиць, навантаження на процесор серверу бази даних тощо. Час виконання запиту, на відміну від метрик навантаження процесору, досить легко виміряти, і водночас дана інформація дає релевантну інформацію щодо ресурсоемності запиту. Отже має сенс зберігати в системі кешування дані, запити до яких виконуються довго, навіть якщо такі запити використовуються рідко. Для цього потрібно додати додаткову метрику до об'єктів кешованих даних, що міститиме час витрачений на виконання відповідного запиту базою даних. Тоді під час періодичного процесу інвалідації кешу, такі дані не будуть видалятися, що додатково підвищить ефективність роботи системи.

При застосуванні вдосконаленого методу кешування даних, для оптимальної роботи інформаційної системи необхідно визначити періоди інвалідації кешу та вагові коефіцієнти запитів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. On Cache Invalidation – Режим доступу до ресурсу: <https://yihui.org/en/2018/06/cache-invalidation/>
2. Cache invalidation overview – Режим доступу до ресурсу: <https://cloud.google.com/cdn/docs/cache-invalidation-overview>
3. Writing, replacement, and invalidation Strategies – Режим доступу до ресурсу: <https://eximia.co/caching-basics-why-when-what-where-writing-replacement-and-invalidation-strategies/>

Горбачов Геннадій Олександрович — студент групи ІКІ-20м, кафедри комп'ютерної інженерії ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: genagorbachov1@gmail.com

Науковий керівник: **Ткаченко Олександр Миколайович** — кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, м. Вінниця, e-mail: alextk1960@gmail.com

Horbachov Hennadii Oleksandrovych — student of the Computer Techniques Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: genagorbachov1@gmail.com

Supervisor: **Tkachenko Oleksandr Mykolaiovych** — Cand. Sc. (Eng), Associate Professor of Computing Engineering Department, Vinnytsia National Technical University, Vinnytsia, e-mail: alextk1960@gmail.com