

АНАЛІЗ МЕТОДІВ КЕШУВАННЯ ДАНИХ У ВИСОКОНАВАНТАЖЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Вінницький національний технічний університет

Анотація

За результатами проведеного аналізу виявлено недоліки та переваги основних підходів до кешування даних та визначено доцільність використання кешування у високонавантажених інформаційних системах.

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

Abstract

According to the results of the analysis, the disadvantages and advantages of the main approaches to data caching are identified and the feasibility of using caching in highly loaded information systems is determined.

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

Актуальність обраної тематики обумовлена необхідністю аналізу та на його основі отримання певних висновків про використання методів кешування даних та їх вплив на швидкодію та надійність роботи інформаційних систем. Кешування широко використовується досить часто у наш час для підвищення ефективності доступу до даних та підвищення швидкодії запитів у різноманітних інформаційних системах. Однак існує багато методів кешування кожен з яких має свої недоліки та переваги у порівнянні з іншими, а також може комбінуватися з іншими методами для оптимізації роботи системи.

Для різних цілей система кешування може розміщуватися на різних рівнях інформаційної системи, але основною задачею кешування на будь-якому рівні є запобігання надлишкових викликів до нижчих рівнів системи. Найнижчим рівнем системи зазвичай є база даних, і у випадку з реляційними базами даних популярні системи управління базами даних такі як MySQL, MsSQL, PostgreSQL вже містять в собі реалізовані методи кешування запитів на рівні таблиці. У вищих шарах архітектури додатку також є багато реалізацій що пропонуються фреймворками, операційними системами та сторонніми бібліотеками, однак дані засоби зазвичай є лише базою для побудови системи кешування що відповідає вимогам інформаційної системи [1]. Окрім того налаштування та використання систем кешування вимагають глибоких знань теми, адже при неправильному використанні кешування може навпаки знизити ефективність роботи системи у декілька разів або призвести до різноманітних програмних помилок. Основними методами кешування у інформаційних системах є наступні:

Cache-Aside – метод кешування що користується найбільшою популярністю, основною ідеєю якого є розміщення системи кешування окремо від сервера, подібно базі даних. Система кешування такого типу постійно оновлює кеш через додаток асинхронно. При необхідності доступу до даних додаток перевіряє, чи існують дані в кеші, і у випадку якщо вони записані зчитує їх, у іншому випадку зчитування відбувається напряму з бази даних. Системи кешування даного типу добре підходять для інформаційних систем що працюють з великою кількістю даних та високими навантаженнями. Така архітектура досить стійка до збоїв кешу, адже у випадку неполадки інформаційна система може звертатися до бази даних. Ще однією перевагою даного підходу є те що модель даних системи кешу може відрізнитись від моделі даних у базі даних [2]. Наприклад, відповідь, сформована в результаті декількох запитів, може зберігатись по спеціальному ідентифікатору запиту. Однак у даного підходу є свої недоліки, через те що система кешування працює незалежно від бази даних інформація у них з часом може відрізнитися. Такий метод кешування дуже простий у реалізації, але складний в використанні та управлінні. Для оновлення даних у кеші доводиться проводити інвалідацію даних щоразу при оновленні даних у базі даних або використовувати токени валідності даних (TTL) і продовжувати обслуговувати застарілі дані до закінчення терміну дії TTL.

Read-Through – метод кешування де система кешування розміщується між сервером та базою

даних, або клієнтським додатком та сервером. При використанні кешу такого типу вся інформація що завантажується з бази даних або сервера зберігається в кеші. Таким чином інформація що повертається у такій системі завжди надсилається з кешу, а сам кеш там база даних тісно пов'язані. Такий тип кешування добре підходить у випадках коли дані не змінюються при цьому зчитуються багато раз, як це відбувається наприклад з архівами або новинами. На відміну від методу Cache-Aside у даній реалізації кешу модель даних що зберігається у кеші не може відрізнитися від моделі даних бази даних. Недоліком такого методу є те що при першому зчитуванні інформації швидкість запиту тільки впаде. Окрім того у даній реалізації також присутня проблема свіжості даних, яка може вирішуватися за допомогою використання токенів валідності даних або доповненням реалізації іншими методами кешування даних.

Write-Through – метод кешування що часто використовується для доповнення методу Read-Through та гарантує актуальність даних у кеші. Сам по собі він лише збільшує затримку перед записом інформації у базу даних, однак при використанні разом з методом Read-Through ми отримуємо всі переваги при зчитуванні інформації а також узгоджуємо дані між базою даних та кешем. Такий метод кешування є досить ресурсоємні і його неправильне використання є особливо небезпечним. Write-Through метод варто застосовувати лише у системах де даних небагато та вони частіше зчитуються ніж оновлюються.

Write-Back – ще один метод кешування що розміщується між базою даних та сервером. При використанні даного методу кешування інформація завжди спочатку записується в кеш і лише тоді система кешування записує дані у базу даних. Write-Back стійкий до збоїв у роботі бази даних і може сам якийсь час слугувати у якості бази даних. Окрім того за допомогою даного методу можливо буферизувати дані перед записом у базу даних, таким чином захищаючи базу даних від перенавантаження під час пікових годин навантаження на систему. Також значною перевагою даної реалізації є те що дані у кеші завжди актуальні. Однак при використанні даного типу кешування система дуже залежна від кешу, і при виході кешу із ладу дані можуть не потрапити у базу даних або може постраждати цілісність даних [3].

При проектуванні кожної інформаційної системи необхідно індивідуально проаналізувати переваги та недоліки при використанні різних методів кешування та кешування в цілому. При роботі з цінною інформацією краще використовувати метод з винесенням кешу за межі системи щоб гарантувати стабільну роботи систему навіть у випадку виходу кешу із ладу. При роботі з даними що часто зчитуються та рідше оновлюються краще обрати Read-Through та Write-Through підходи. Для подолання проблем пікового навантаження та корекції навантаження на базу даних варто звернути увагу на метод кешування Write-Back. Інколи ж кешування є надлишковим та лише ускладнює розробку та підтримку системи, а проблеми швидкодії системи потрібно вирішувати іншими способами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Caching Strategies and How to Choose the Right One – Режим доступу до ресурсу: <https://codeahoy.com/2017/08/11/caching-strategies-and-how-to-choose-the-right-one/>
2. Caching challenges and strategies – Режим доступу до ресурсу: <https://aws.amazon.com/builders-library/caching-challenges-and-strategies/>
3. Deploy ASP.NET Core apps to Azure App service – Режим доступу до ресурсу: <https://medium.com/the-cloud-architect/patterns-for-resilient-architecture-part-4-85afa66d6341>

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