### ЛЕКЦІЯ 8. КЕШУВАННЯ В КЛІЄНТ-СЕРВЕРНІЙ АРХІТЕКТУРІ. ОСНОВИ КЕШУВАННЯ. КЛІЄНТСЬКЕ ТА СЕРВЕРНЕ КЕШУВАННЯ. ВИКОРИСТАННЯ КЕШІВ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ

Кешування — це одна з основних технологій для підвищення продуктивності клієнт-серверної архітектури. Воно дозволяє зменшити завантаження на сервери, покращити час відгуку та оптимізувати використання ресурсів мережі. У цій лекції ми розглянемо основи кешування, його принципи, різні типи кешування (клієнтське та серверне) та приклади використання кешів для оптимізації взаємодії між клієнтом та сервером.

**1. Основи кешування**

Кешування — це процес тимчасового збереження результатів виконання операцій або даних для швидкого доступу в майбутньому. Кеші використовуються для зменшення кількості звернень до вільних або видалених джерел даних, таких як бази даних або зовнішні веб-сервіси.

**1.1 Принцип роботи кешу**

Основна ідея кешування виникає в тому, що коли клієнт запитує ресурс (наприклад, веб-сторінку або сайт), система перевіряє, чи збережений цей ресурс у кеші. Якщо він доступний у кеші, результат повертається зараз без необхідності взаємодії з основним сервером. Якщо ресурс не знайдений у кеші, сервер отримує його зі сховища, обробляє та зберігає в кеші для подальшого використання.

**1.2 Види кешування**

Залежно від місця зберігання кешу, існує кілька видів кешу:

* **Клієнтське кешування** : збереження даних на боці клієнта (наприклад, у браузері або на мобільному пристрої).
* **Серверне кешування** : збереження даних на боці сервера (наприклад, в RAM або спеціалізованих сховищах).
* **Кешування на проміжному рівні** : кешування даних на стороні проксі-серверів або CDN (Content Delivery Network).

**1.3 Переваги кешування**

Основні переваги використання кешування включають:

* **Покращення швидкості завантаження** : зменшення затримок при зверненні до ресурсів.
* **Змінення навантаження на сервери** : серверу не потрібно обробляти одні й ті ж запити повторно.
* **Економія ресурсів мережі** : кількість запитів до видалених серверів зменшується.
* **Збільшення стійкості системи** : у випадку проблеми з доступом до основного сервера, кешовані дані можуть бути використані для тимчасового забезпечення функціональності.

**2. Клієнтське кешування**

Клієнтське кешування — це механізм, за допомогою якого зберігаються деякі пристрої користувача (наприклад, у браузері). Клієнтське кешування є ефективним для зменшення часу завантаження веб-сторінок або мобільних додатків, після збережених раніше даних не потрібно запитувати повторно.

**2.1 HTTP-заголовки для кешування**

У клієнтському кешуванні широко використовують HTTP-заголовки для контролю того, як і коли кешують дані. Основні заголовки включають:

* **Cache-Control** : використання для вказівки політики кешування. Наприклад, заголовок Cache-Control: max-age=3600 інструктує зберегти ресурс браузера в кеші протягом першої години (3600 секунд).
* **Expires** : вибрати дату й час, після якого ресурс застарілим.
* **ETag** : використання для ідентифікації версії ресурсу. Якщо версія ресурсу змінилася, кеш повинен оновитися.
* **Last-Modified** : вказує час останньої модифікації ресурсу. Браузер використовує цей заголовок для перевірки, чи змінився ресурс після попереднього завантаження.

**2.2 Приклад клієнтського кешування в браузері**

Припустимо, що клієнт відвідує веб-сайт, і його браузер завантажує зображення. Сервер може відповісти з наступними заголовками:

http

HTTP/1.1 200 OK

Content-Type: image/png

Cache-Control: max-age=86400

ETag: "abc123"

Це інструктує браузер зберігати зображення в кеші протягом 24 годин. При наступному запиті на це зображення браузер перевірити кеш, а не відправити новий запит на сервер.

**2.3 Проблеми клієнтського кешування**

Одна з проблем клієнтського кешування виникає в тому, що збережені в кеші дані можуть застаріти. Щоб уникнути цього, розробники можуть використовувати механізми, які спонукають клієнта до актуальності кешу (наприклад, через перевірку ETag або Last-Modified). Ще одна проблема — це конфлікти, коли клієнтський кеш може надрукувати невірні дані через локальні зміни або конфігурації, які потребують очищення кешу.

**3. Серверне кешування**

Серверне кешування створюється на стороні сервера і може включати збереження результатів запитів до бази даних або обробки в оперативній пам’яті або на диску. Серверне кешування зазвичай використовується для зменшення завантаження бази даних або оптимізації взаємодії з віддаленими сервісами.

**3.1 Кешування баз даних**

Один із найпоширеніших випадків серверного кешування — це кешування результатів запитів до бази даних. Наприклад, якщо клієнт часто запитує один і той самий список продуктів, сервер може зберегти цей список у кеші, щоб не виконати той самий SQL-запит щоразу.

**3.2 Інструменти для серверного кешування**

Існує багато технологій та інструментів, які використовують для серверного кешування. Одними з найпопулярніших є:

* **Memcached** : високопродуктивна система кешування, яка зберігає дані в оперативній пам'яті для швидкого доступу.
* **Redis** : інша система кешування, яка також працює в оперативній пам'яті, але має більше функціональних можливостей, включаючи підтримку складних типів даних і стійкість даних на диску.

**3.3 Приклад використання Memcached**

Припустимо, що сервер обробляє запит на отримання списку користувачів:

1. Спочатку перевіряє сервер, чи є цей список у кеші Memcached.
2. Якщо список є у кеші, він знову повертається.
3. Якщо списку немає в кеші, сервер виконує запит до бази даних, зберігає результат у Memcached і повертає його клієнту.

**4. Проміжне кешування (CDN)**

Для великих веб-сайтів, які обслуговують велику кількість користувачів з різних географічних регіонів, ефективним рішенням є використання мережі доставки вмісту (CDN). CDN зберігають копії статичного контенту (зображень, стилів, скриптів) на своїх серверах по всьому світу, що дозволяє доставляти контент ближче до кінцевого користувача.

**4.1 Принцип роботи CDN**

Коли клієнт запитує ресурс через CDN, система шукає найближчий сервер CDN до клієнта та доставляє йому кешований контент із цього сервера. Знижує затримки та прискорює доставку контенту.

**4.2 Переваги використання CDN**

* **Зменшення часу завантаження** : ресурси доставляються з географічно ближчих серверів.
* **Зниження навантаження на основний сервер** : статичний контент обробляється CDN.
* **Стійкість до високих навантажень** :

**5.** **Використання кешів для оптимізації роботи**

Ефективне використання кешування дозволяє значно оптимізувати роботу веб-сайтів та додатків

* **Оптимізація часу завантаження сторінок** :
* **Зміна навантаження на сервер**: сервер
* **Поліпшення користувацького досвіду** :

Кешування є продуктом елемента в архітектурі клієнт-серверних систем, який дозволяє підвищити продуктивність, втратити затримки і забезпечити стабільну роботу системи навіть при високих навантаженнях. Розуміння механізмів кешування, використання відповідних інструментів і технологій, таких як Memcached, Redis або CDN, додатково розробникам створюють більш ефективні та масштабовані рішення.