

К.т.н., доцент Замятін Д.С., магістрант Васильєв В.О.

**Національний технічний університет України
«Київський політехнічний інститут»**

ВИБІР СТРАТЕГІЇ КЕШУВАННЯ ДЛЯ СЕРВЕРІВ ПІДТРИМКИ СОЦІАЛЬНИХ МЕРЕЖ

Вступ

На сьогодні кількість користувачів соціальних мереж неухильно збільшується, тому постають вимоги щодо часу відгуку web-систем. Задоволення вимог ефективності роботи системи можливе шляхом застосування технології багаторівневого кешу. При використанні кешування істотно знижується навантаження на базу даних (БД) [1], зменшується кількість дискових операцій, які замінюються швидкішими зверненнями до оперативно-запам'ятовуючого пристрою (ОЗП). Також кешування дозволяє зменшити кількість викликів алгоритмів при побудові блоку html-коду об'єкта до його наступної зміни. В цілому, ефект у роботі web-систем виражається в кількості запитів за одиницю часу, що виправдовує витрати на створення кеша й збільшення обсягу ОЗП.

Постановка задачі

Задачею дослідження є зниження часу відгуку в web-орієнтованих системах підтримки соціальних мереж за допомогою кешування.

Основний матеріал

Розглянемо схему побудови web-сторінки за запитом користувача для системи без використання кешу. В загальному випадку можна виділити основні етапи роботи: запити до БД та формування результату у вигляді html-сторінки. В системі з кешуванням робота буде полягати у запиті до кеш-серверу [3], причому загальний час відгуку набагато менший за час у системі без кешування, тому що використовується швидкий ОЗП. Таким чином, системи з кешуванням дозволяють замінити послідовність високовартісних мережених запитів (наприклад, до БД) на одноразове звернення до ОЗП. Отже, система без кешування не дозволяє задовольнити вимогам щодо застосування часу відгуку, тому в даній роботі не розглядаються.

В соціальних мережах за частотою оновлення можна виділити три основні види даних: рідко оновлюванні дані (статичні сторінки, меню користувача і т.д.); часто оновлюванні дані (загальна інформація користувача); миттєві дані (приватні повідомлення, коментарі). Відповідно до цих рівнів, в роботі запропоновано побудувати систему з 3-ма рівнями кешу.

Загальна схема системи з трирівневим кешем подана на рис. 1.

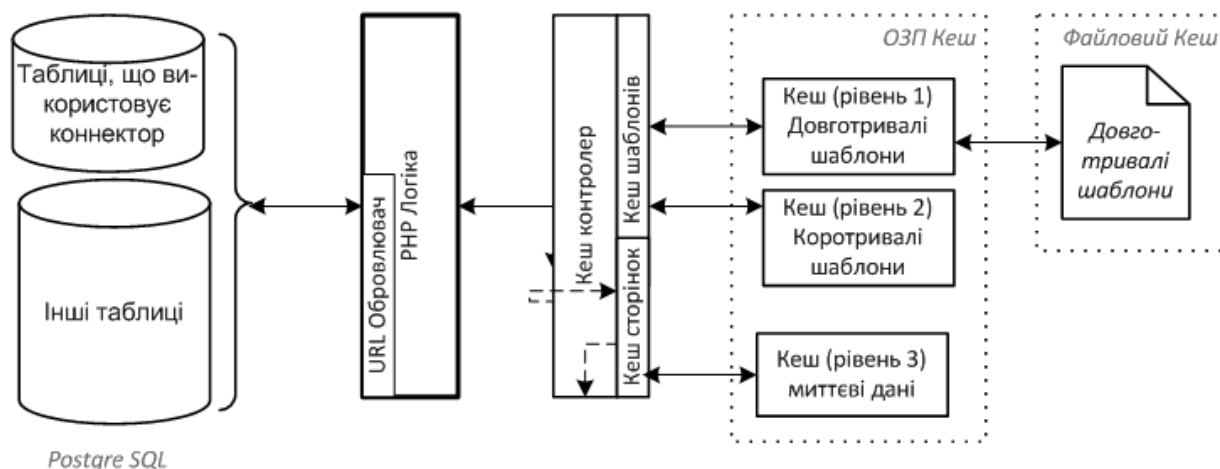


Рис. 1. Загальна схема системи

Кеш розшарований на 3 рівні залежно від частоти оновлення вихідних даних [2], а саме: рідко оновлюванні дані (кеш рівня 1 – L1); часто оновлюванні дані (кеш рівня 2 – L2) і дані миттєві (надоперативний кеш 3-го рівня – L3). Рівні кешу та характеристика даних подані у таблиці 1.

Таблиця 1. Зведена таблиця рівнів кеша

Характеристика даних	L1	L2	L3
Час життя об'єкта	Максимальне	Середнє	Мінімальне
Ступінь «зібраності» об'єкта	Блок з можливими включеннями в нього	Блок без включень	Повністю готова сторінка
Спосіб зберігання	ОЗУ, можливо файли	ОЗУ	ОЗУ

Таким чином, запит, що надходить від клієнта, спочатку втрапляє в кеш 3-ого рівня; якщо там відсутня сторінка, що задовольняє параметрам запиту, то система шукає її в кеші 2-ого рівня, а потім в 1-ому рівні. В середньостатистичних запитах, далі 2-ого рівня контролер кеша рідко передає управління, тому що на 1-ому рівні дані оновлюється вкрай рідко, тобто завжди актуальні.

Нехай кількість записів кешу 1-ого рівня n_1 , кількість записів кешу 2-ого рівня n_2 , кількість кешу 3-ого рівня n_3 . Час запиту до кешу 1-ого рівня – t_1 , до 2-ого рівня – t_2 , до 3-ого рівня – t_3 . Тоді загальний час обробки буде рівнятися $(t_1 + t_2 + t_3) \cdot (n_1 + n_2 + n_3)$ у кеші з загальною кількістю записів $(n_1 + n_2 + n_3)$.

Якщо виділити в кеші два рівня (перший рівень повторює рівень з 3-ох рівневого кеша, другий рівень отримується шляхом об'єднання 2-ого і 3-ого рівней 3-рівневого кеша), тоді загальний час обробки буде дорівнювати $(t_1 + t_2) \cdot (n_1 + n_2 + n_3)$ у кеші з загальною кількістю записів $(n_1 + n_2 + n_3)$.

Якщо виділити в кеші тільки один рівень (отримується шляхом об'єднання всіх рівнів з трьохрівневого кеша), тоді загальний час обробки буде дорівнювати $t_1 \cdot (n_1 + n_2 + n_3)$ у кеші з загальною кількістю записів $(n_1 + n_2 + n_3)$.

Отже, кеш з одним рівнем визначається найменшим часом, але найбільшою кількістю записів; кеш з трьома рівнями визначається найбільшим часом, але найменшою кількістю записів. Але кеш знаходиться у

швидкому ОЗП, тому переваги по часу однорівневого кешу незначні і не співвідносяться з перевагами кешу з трьома рівнями по кількості записів.

Проведемо порівняння по часу доступу трьохрівневого кешу та однорівневого кешу. Нехай кількість кеш-промахів за одиницю часу для першого рівня складає k_1 , кількість кеш-промахів для другого рівня k_2 , для третього рівня – k_3 . Час доступу для першого рівня p_1 , для другого – p_2 , для третього – p_3 .

Загальний час доступу для системи з трьома рівнями кешу:

$$k_1(p_1 + p_2 + p_3) + k_2(p_2 + p_3) + k_3 p_3 = k_1 p_1 + k_1 p_2 + k_1 p_3 + k_2 p_2 + k_2 p_3 + k_3 p_3$$

Загальний час доступу для системи з одним рівнем кешу:

$$(k_1 + k_2 + k_3)(p_1 + p_2 + p_3) = k_1 p_1 + k_1 p_2 + k_1 p_3 + k_2 p_1 + k_2 p_2 + k_2 p_3 + k_3 p_1 + k_3 p_2 + k_3 p_3$$

$$A = k_1 p_1 + k_1 p_2 + k_1 p_3 + k_2 p_2 + k_2 p_3 + k_3 p_3$$

Знайдемо відношення часу доступу для системи з трьома рівнями кешу до часу доступу для системи з одним рівнем кешу:

$$\frac{A}{A + k_2 p_1 + k_3 p_1 + k_3 p_2} < 1$$

З цього відношення видно, що час доступу до кешу з трьома рівнями менший, ніж час доступу до кешу з одним рівнем.

Висновки

Таким чином, у даній статті запропоновано стратегію кешування на основі трирівневого кешу. Показано, що застосування кешу з трьома рівнями для систем підтримки соціальних мереж дає переваги по часу відгуку та кількості записів у порівнянні з кешами одно- та дворівневою стратегією кешування. Запропонована архітектура використана при розробці системи, яка в даний момент застосовується з метою експериментального підтвердження наведених результатів.

Література

1. *Hong-Tai Chou, David J. De Witt. An Evaluation of Buffer Management Strategies for Relational Database Systems. Algorithmica 1 (3), 1986.– P. 311-336.*
2. *S. Dar, M.J. Franklin, B.T. Jonsson, D. Shrivastava, M. Tan. Semantic Data Caching and Replacement. Proc. VLDB, 1996. – P. 330-341.*
3. *Ramakrishna Karedla, J. Spencer Love, Bradley G. Wherry. Caching strategies to improve disk system performance, Computer, v.27 n.3, March, 1994. - P. 38-46.*