

Міністерство освіти і науки України
НТУУ «Київський політехнічний інститут»
Фізико-технічний інститут

Проектування високонавантажених систем

Лабораторна робота №6
Кешування з використанням Hazelcast

Виконав:

Студент 4-го курсу
групи ФІ-21
Климент'єв Максим
Перевірив:

Зміст

1	Код реалізації	3
2	Результати	4

1 Код реалізації

2 Результати

На основі першого завдання з Web-counter, додати до нього Read-through та Write-through кешування.

У якості системи кешування, використати кластер Hazelcast з відповідними налаштуваннями <https://docs.hazelcast.com/hazelcast/5.6/mapstore/working-with-external-data>

Ідея полягає у тому, що значення каунтера зберігається у базі даних, але операції з ним відбувались не на пряму, а крізь кеш.

1. Налаштuvати Read-through та Write-through на основі generic MapLoader та generic MapStore для БД PostgreSQL (або MySQL)

2. Припускаючи, що у нас буде 4 різних каунтерів, додати в Hazelcast Map, яка відповідає таблиці БД, 4 окремих записи з різними ID (1-4) та 0 у якості початкового значення. Перевірити через прямий запит до БД, що ці записи в ній з'явились

3. Для окремих створених ключів (каунтерів) повторити тести 1) - 4) з першої роботи, забезпечуючи коректність результатів (тобто щоб кінцевий результат був коректним) та мірюючи час. Тобто тест 1 має використовувати каунтер 1, тест 2 - каунтер 2, і т.д.

- (1) Один клієнт робить послідовно 10K викликів, кінцеве значення count = 10K - порахувати кількість запитів в секунду
- (2) Два клієнти одночасно роблять по 10K викликів кожен, кінцеве значення count = 20K - порахувати кількість запитів в секунду
- (3) 5 клієнтів одночасно роблять по 10K викликів кожен, кінцеве значення count = 50K - порахувати кількість запитів в секунду
- (4) 10 клієнтів одночасно роблять по 10K викликів кожен, кінцеве значення count = 100K - порахувати кількість запитів в секунду

4. Перевірити кінцеві значення каунтерів у БД та порівняти отримані результати з продуктивності з отриманими у першій роботі.