

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  
Климентьєв Максим  
**Перевірив:**

---

# Зміст

<b>1</b>	<b>Код реалізації</b>	<b>3</b>
<b>2</b>	<b>Результати</b>	<b>4</b>

# **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. Налаштувати 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'явилися
3. Для окремих створених ключів (каунтерів) повторити тести 1) - 4) з першої роботи, забезпечуючи коректність результатів (тобто щоб кінцевий результат був коректним) та міряючи час. Тобто тест 1 має використовувати каунтер 1, тест 2 - каунтер 2, і т.д.
  - (1) Один клієнт робить послідовно 10К викликів, кінцеве значення count = 10K - порахувати кількість запитів в секунду
  - (2) Два клієнти одночасно роблять по 10К викликів кожен, кінцеве значення count = 20K - порахувати кількість запитів в секунду
  - (3) 5 клієнтів одночасно роблять по 10К викликів кожен, кінцеве значення count = 50K - порахувати кількість запитів в секунду
  - (4) 10 клієнтів одночасно роблять по 10К 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.

```
[INFO] Connecting to Hazelcast...
[INFO] Clearing Counters...
[INFO] Data: [(1, 0), (2, 0), (3, 0), (4, 0)]
[INFO] Running Counter 1
[INFO] Counter ID: 1
[INFO] Clients: 1 | Increments per client: 10000
[RESULT] Time: 44.36 sec
[RESULT] Throughput: 225.41
[RESULT] Received: 10_000 (100.00% from Expected: 10_000)
[INFO] Running Counter 2
[INFO] Counter ID: 2
[INFO] Clients: 2 | Increments per client: 10000
[RESULT] Time: 69.05 sec
```

```

[RESULT] Throughput: 289.64
[RESULT] Received: 20_000 (100.00% from Expected: 20_000)
[INFO] Running Counter 3
[INFO] Counter ID: 3
[INFO] Clients: 5 | Increments per client: 10000
[RESULT] Time: 170.70 sec
[RESULT] Throughput: 292.91
[RESULT] Received: 50_000 (100.00% from Expected: 50_000)
[INFO] Running Counter 4
[INFO] Counter ID: 4
[INFO] Clients: 10 | Increments per client: 10000
[RESULT] Time: 354.13 sec
[RESULT] Throughput: 282.39
[RESULT] Received: 100_000 (100.00% from Expected: 100_000)
[INFO] Data: [(1, 10000), (2, 20000), (3, 50000), (4, 100000)]

```

```

[INFO] Один клієнт
[Name] Memory
[Sent] 10000 requests in 3.29 seconds
[Throughput]: 3040.52 requests/sec
[Name] DataBase
[Sent] 10000 requests in 10.75 seconds
[Throughput]: 929.95 requests/sec
[Name] Hazelcast Cash
[RESULT] Time: 44.36 sec
[RESULT] Throughput: 225.41

[INFO] Два клієнти
[Name] Memory
[Sent] 20000 requests in 6.67 seconds
[Throughput]: 2996.28 requests/sec
[Name] DataBase
[Sent] 20000 requests in 22.13 seconds
[Throughput]: 903.66 requests/sec
[Name] Hazelcast Cash
[RESULT] Time: 69.05 sec
[RESULT] Throughput: 289.64

[INFO] П'ять клієнтів
[Name] Memory
[Sent] 50000 requests in 17.83 seconds
[Throughput]: 2804.70 requests/sec
[Name] DataBase
[Sent] 50000 requests in 57.78 seconds
[Throughput]: 865.40 requests/sec
[Name] Hazelcast Cash
[RESULT] Time: 170.70 sec
[RESULT] Throughput: 292.91

[INFO] Десять клієнтів
\textbf{}
[Name] Memory
[Sent] 100000 requests in 39.65 seconds
[Throughput]: 2521.88 requests/sec
[Name] DataBase
[Sent] 100000 requests in 117.59 seconds
[Throughput]: 850.39 requests/sec
[Name] Hazelcast Cash
[RESULT] Time: 354.13 sec
[RESULT] Throughput: 282.39

```

---

**Виходить повільний запис, але надійний та можна дуже швидко читати**