

Проектування високонавантажених систем

Лабораторна робота 2

Торгало Ігнатій ФБ-51мп

1. Конфігурація середовища

Docker-compose.yml

```
1 version: '3.8'
2
3 services:
4   hz1:
5     image: hazelcast/hazelcast:5.4.0
6     container_name: hz1
7     environment:
8       - HZ_CLUSTERNAME=lab-cluster
9       - HAZELCAST_CONFIG=/opt/hazelcast/config/hazelcast.yaml
10    volumes:
11      - ./hazelcast.yaml:/opt/hazelcast/config/hazelcast.yaml
12    ports:
13      - "5701:5701"
14
15   hz2:
16     image: hazelcast/hazelcast:5.4.0
17     container_name: hz2
18     environment:
19       - HZ_CLUSTERNAME=lab-cluster
20       - HAZELCAST_CONFIG=/opt/hazelcast/config/hazelcast.yaml
21     volumes:
22       - ./hazelcast.yaml:/opt/hazelcast/config/hazelcast.yaml
23     ports:
24       - "5702:5701"
25
26   hz3:
27     image: hazelcast/hazelcast:5.4.0
28     container_name: hz3
29     environment:
30       - HZ_CLUSTERNAME=lab-cluster
31       - HAZELCAST_CONFIG=/opt/hazelcast/config/hazelcast.yaml
32     volumes:
33       - ./hazelcast.yaml:/opt/hazelcast/config/hazelcast.yaml
34     ports:
35       - "5703:5701"
```

Hazelcast.yaml

```
1 hazelcast:
2   cluster-name: lab-cluster
3
4   network:
5     join:
6       multicast:
7         enabled: true
8
9   cp-subsystem:
10    cp-member-count: 3
11    group-size: 3
12    session-time-to-live-seconds: 30
13    session-heartbeat-interval-seconds: 5
```

2. Запуск кластера Hazelcast

```

Members {size:3, ver:3} [
    Member [172.20.0.2]:5701 - d9c81743-17a1-4ba4-9784-cdf1cf5a4b1d this
    Member [172.20.0.3]:5701 - 26fd1dbd-e6c2-420f-8b9c-7a57cc73cf30
    Member [172.20.0.4]:5701 - 39066c56-487c-4c9a-9251-fe757157fe21
]

```

```

└─ sudo docker logs hz1 | grep "CP Subsystem"
2025-12-08 13:26:52,673 [ INFO] [main] [c.h.c.CPSubsystem]: [172.20.0.2]:5701 [lab-cluster] [5.4.0] CP Subsystem is enabled with 3 members.
2025-12-08 13:26:58,116 [ INFO] [hz.suspicious_morse.cached.thread-2] [c.h.c.i.MetadataRaftGroupManager]: [172.20.0.2]:5701 [lab-cluster] [5.4.0] CP Subsystem is initialized with: [CPMember{uuid=d9c81743-17a1-4ba4-9784-cdf1cf5a4b1d, address=[172.20.0.2]:5701}, CPMember{uuid=26fd1dbd-e6c2-420f-8b9c-7a57cc73cf30, address=[172.20.0.3]:5701}, CPMember{uuid=39066c56-487c-4c9a-9251-fe757157fe21, address=[172.20.0.4]:5701}]

```

3. Код клієнта

Клієнт підключається до кластеру з трьох нод, отримує доступ до розподілених структур даних (Distributed Map та IAtomicLong) та виконує інкрементацію каунтера у 10 потоках для демонстрації різних механізмів синхронізації.

client.py

```

1 import hazelcast
2 import threading
3 import time
4
5 ITERATIONS = 2000
6 THREADS = 10
7
8 hz = hazelcast.HazelcastClient(
9     cluster_name="lab-cluster",
10    cluster_members=["localhost:5701", "localhost:5702", "localhost:5703"],
11 )
12
13 dist_map = hz.get_map("counter_map").blocking()
14 atomic = hz.cp_subsystem.get_atomic_long("counter_long").blocking()
15
16 def run_test(name, setup, worker):
17     setup()
18
19     threads = []
20     start = time.time()
21 |
22     for _ in range(THREADS):
23         t = threading.Thread(target=worker)
24         t.start()
25         threads.append(t)
26
27     for t in threads:
28         t.join()
29
30     duration = time.time() - start
31     print(f"{name} RESULT:", dist_map.get("cnt") if "cnt" in dist_map.key_set() else atomic.get())
32     print(f"{name} TIME:", duration)
33     print("-" * 50)
34
35 def test_no_lock():
36     def setup():
37         dist_map.put("cnt", 0)
38
39     def worker():
40         for i in range(ITERATIONS):
41             val = dist_map.get("cnt")
42             dist_map.put("cnt", val + 1)
43
44     run_test("NO LOCK", setup, worker)
45

```

```

46 def test_pessimistic():
47     def setup():
48         dist_map.put("cnt", 0)
49
50     def worker():
51         for i in range(ITERATIONS):
52             dist_map.lock("cnt")
53             val = dist_map.get("cnt")
54             dist_map.put("cnt", val + 1)
55             dist_map.unlock("cnt")
56
57     run_test("PESSIMISTIC", setup, worker)
58
59 def test_optimistic():
60     def setup():
61         dist_map.put("cnt", 0)
62
63     def worker():
64         for _ in range(ITERATIONS):
65             while True:
66                 old = dist_map.get("cnt")
67                 if dist_map.replace_if_same("cnt", old, old + 1):
68                     break
69
70     run_test("OPTIMISTIC", setup, worker)
71

```

```

72 def test_atomic_long():
73     def setup():
74         atomic.set(0)
75
76     def worker():
77         for _ in range(ITERATIONS):
78             atomic.increment_and_get()
79
80     threads = []
81     start = time.time()
82
83     for _ in range(THREADS):
84         t = threading.Thread(target=worker)
85         t.start()
86         threads.append(t)
87
88     for t in threads:
89         t.join()
90
91     duration = time.time() - start
92     print("IATOMICLONG RESULT:", atomic.get())
93     print("IATOMICLONG TIME:", duration)
94     print("-" * 50)
95
96
97 test_no_lock()
98 test_pessimistic()
99 test_optimistic()
100 test_atomic_long()
101
102 hz.shutdown()

```

4. Результати роботи

```

python3 client.py
NO LOCK RESULT: 2890
NO LOCK TIME: 24.743112564086914
-----
PESSIMISTIC RESULT: 20000
PESSIMISTIC TIME: 689.4249777793884
-----
OPTIMISTIC RESULT: 20000
OPTIMISTIC TIME: 32.552724838256836
-----
IATOMICLONG RESULT: 20000
IATOMICLONG TIME: 4.568831920623779
-----
```

5. Висновки

Тестування чотирьох моделей інкрементації показало суттєву різницю як у точності, так і у швидкості. Підхід без блокувань дав найгірший результат — значна частина операцій губиться через одночасний доступ. Песимістичне блокування забезпечує правильний підсумок, але працює надзвичайно повільно через постійне захоплення та звільнення блокування. Оптимістичний підхід працює значно швидше і теж дає правильний результат, але вимагає повторних спроб при конфліктах. Найкращим рішенням став IAtomicLong, який поєднує коректність і максимальну швидкість завдяки використанню CP Subsystem та Raft. Це підтверджує, що для розподілених лічильників найефективніше використовувати спеціалізовані атомарні структури, а не стандартні операції мапи.