Task2 - Розгортання і робота з distributed in-memory data structures на основі Hazelcast: Distributed Map

Сотнікова Поліна ФБ-41мп

Посилання на Гітхаб репозиторій: lab2

Завдання:

1. Встановити і налаштувати Hazelcast https://hazelcast.com/open-source-projects/downloads/

```
┌──(kali㉿kali)-[~/Downloads/hazelcast-5.4.0/bin]
└─$ ./hz -V
Version: 5.4.0
```

```
┌──(kali㉿kali)-[~/Downloads/hazelcast-5.4.0/bin]
└─$ ./hz-start
Warning: ./hz-start is deprecated. Please use hz start instead.
########################################
# JAVA=/usr/bin/java
# JAVA_OPTS=--add-modules java.se --add-exports java.base/jdk.internal.ref=ALL-UNNAMED --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/sun.nio.ch=ALL-UNNAM
ED --add-opens java.management/sun.management=ALL-UNNAMED --add-opens jdk.management/com.sun.management.internal=ALL-UNNAMED -Dlog4j.shutdownHookEnabled=false -Dhazelcast.lo
gging.shutdown=true -Dhazelcast.logging.type=log4j2 -Dlog4j.configurationFile=file:/home/kali/Downloads/hazelcast-5.4.0/config/log4j2.properties -Dhazelcast.config=/home/kal
i/Downloads/hazelcast-5.4.0/config/hazelcast.xml -Djet.custom.lib.dir=/home/kali/Downloads/hazelcast-5.4.0/custom-lib
# CLASSPATH=/home/kali/Downloads/hazelcast-5.4.0/lib:/home/kali/Downloads/hazelcast-5.4.0/lib/*:/home/kali/Downloads/hazelcast-5.4.0/bin/user-lib:/home/kali/Downloads/hazelc
ast-5.4.0/bin/user-lib/*
########################################
2025-03-21 07:09:03,403 [ INFO] [main] [c.h.i.c.AbstractConfigLocator]: Loading configuration '/home/kali/Downloads/hazelcast-5.4.0/config/hazelcast.xml' from System property 'hazelcast
.config'
2025-03-21 07:09:03,420 [ INFO] [main] [c.h.i.c.AbstractConfigLocator]: Using configuration file at /home/kali/Downloads/hazelcast-5.4.0/config/hazelcast.xml
2025-03-21 07:09:05,051 [ WARN] [main] [c.h.c.AbstractXmlConfigHelper]: Name of the hazelcast schema location[http://www.hazelcast.com/schema/config/hazelcast-config-5.3.xsd] is incorre
ct, using default
2025-03-21 07:09:07,705 [ INFO] [main] [c.h.i.AddressPicker]: [LOCAL] [dev] [5.4.0] Interfaces is enabled, trying to pick one address matching to one of: [127.0.0.1, 127.0.0.1]
2025-03-21 07:09:08,075 [ INFO] [main] [c.h.s.logo]: [127.0.0.1]:5701 [dev] [5.4.0]
        o    o     o     o---o   o--o o      o---o     o    o---o o--o---o
        |    |    / \       /        |     /         / \   |       |
        o---o       o     o   o---o |    o            o   o---o   |
        |   |  *     \   /          |     \      *     \      |   |
        o   o *       o o---o   o--o o---o o---o *       o o---o   o
2025-03-21 07:09:08,076 [ INFO] [main] [c.h.system]: [127.0.0.1]:5701 [dev] [5.4.0] Copyright (c) 2008-2024, Hazelcast, Inc. All Rights Reserved.
2025-03-21 07:09:08,107 [ INFO] [main] [c.h.system]: [127.0.0.1]:5701 [dev] [5.4.0] Hazelcast Platform 5.4.0 (20240415) starting at [127.0.0.1]:5701
2025-03-21 07:09:08,117 [ INFO] [main] [c.h.system]: [127.0.0.1]:5701 [dev] [5.4.0] Cluster name: dev
2025-03-21 07:09:08,119 [ INFO] [main] [c.h.system]: [127.0.0.1]:5701 [dev] [5.4.0] Integrity Checker is disabled. Fail-fast on corrupted executables will not be performed. For more inf
ormation, see the documentation for Integrity Checker.
2025-03-21 07:09:08,123 [ INFO] [main] [c.h.system]: [127.0.0.1]:5701 [dev] [5.4.0] Jet is enabled
```

```
Members {size:1, ver:1} [
        Member [127.0.0.1]:5701 - b08c548b-3421-495e-b403-e4bd3bcfcd6d this
]
```

2. Сконфігурувати і запустити 3 ноди (інстанси) об'єднані в кластер або як частину Java-застосування, або як окремі застосування

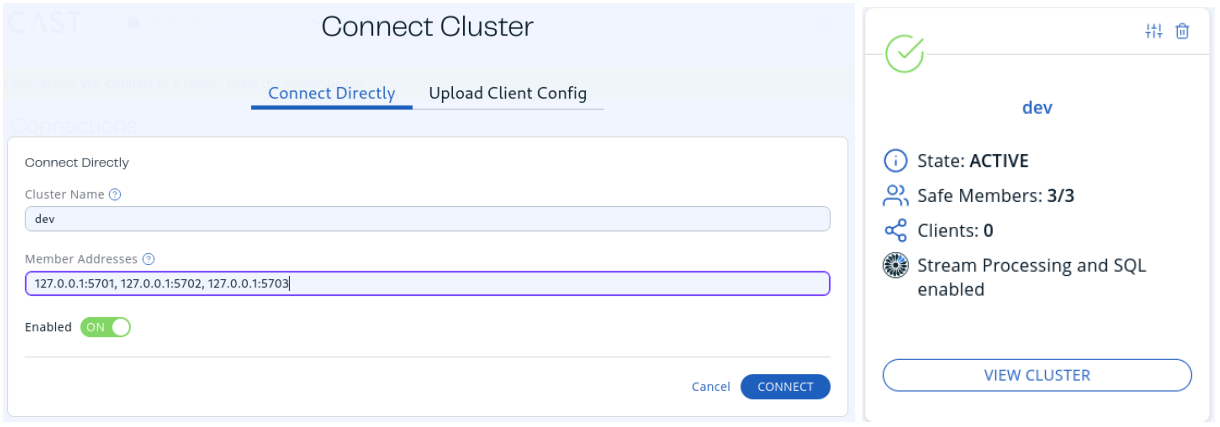
Додамо ще 2 ноди:

```
2025-03-21 13:03:59,577 [ INFO] [hz.gallant_pare.IO.thread-in-0] [c.h.i.s.t.TcpServerConnection]: [127.0.0.1]:5702 [dev] [5.4.0] Initial
ized new cluster connection between /127.0.0.1:5702 and /127.0.0.1:59011
2025-03-21 13:03:59,676 [ INFO] [hz.gallant_pare.generic-operation.thread-0] [c.h.i.c.ClusterService]: [127.0.0.1]:5702 [dev] [5.4.0]

Members {size:3, ver:3} [
        Member [127.0.0.1]:5701 - 840b5a5d-4867-4648-9653-a16a28437dfb
        Member [127.0.0.1]:5702 - fc40ef29-3013-4c8e-aa05-219cb6b748b8 this
        Member [127.0.0.1]:5703 - ed525daa-c2d5-49a2-a153-0ebe6fdc02d3
]
```

Запустимо hazelcast management center:

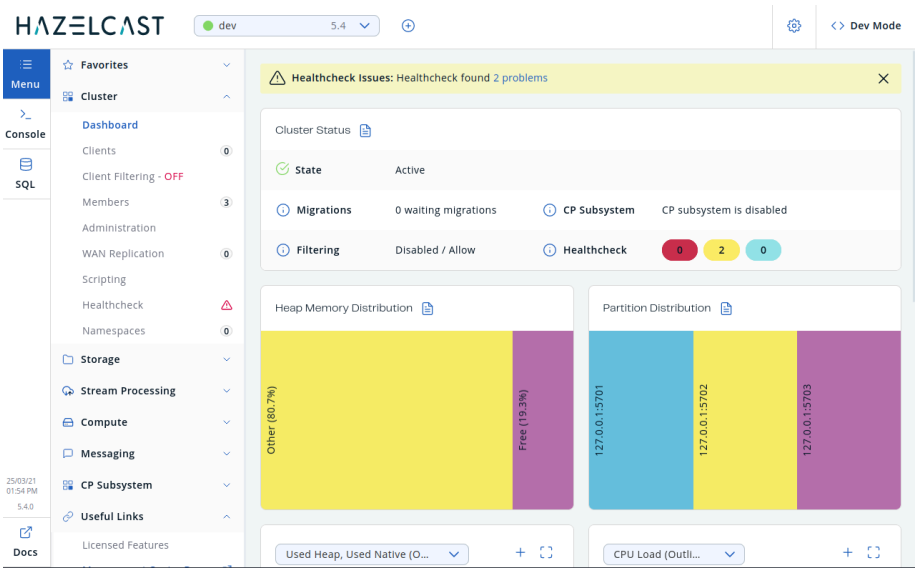
```
┌──(kali㉿kali)-[~/Downloads/hazelcast-5.4.0/management-center/bin]
└─$ ./start.sh
Warning: start.sh is deprecated. Please use hz-mc start instead.
2025-03-21 13:48:31,421 [ INFO] [main] [c.h.w.MCApplication]: Starting MCApplication using Java 21.0.6 with PID 180184 (/home/kali/Downl
oads/hazelcast-5.4.0/management-center/hazelcast-management-center-5.4.0.jar started by kali in /home/kali/Downloads/hazelcast-5.4.0/man
agement-center)
2025-03-21 13:48:31,497 [ INFO] [main] [c.h.w.MCApplication]: No active profile set, falling back to 1 default profile: "default"
2025-03-21 13:48:47,414 [ INFO] [main] [l.lockservice]: Successfully acquired change log lock
2025-03-21 13:48:50,358 [ INFO] [main] [l.util]: UPDATE SUMMARY
2025-03-21 13:48:50,358 [ INFO] [main] [l.util]: Run:                          30
2025-03-21 13:48:50,360 [ INFO] [main] [l.util]: Previously run:                0
2025-03-21 13:48:50,361 [ INFO] [main] [l.util]: Filtered out:                  0
2025-03-21 13:48:50,363 [ INFO] [main] [l.util]: -------------------------------
2025-03-21 13:48:50,366 [ INFO] [main] [l.util]: Total change sets:            30
2025-03-21 13:48:50,371 [ INFO] [main] [l.util]: Update summary generated
2025-03-21 13:48:50,391 [ INFO] [main] [l.lockservice]: Successfully released change log lock
2025-03-21 13:48:58,255 [ INFO] [main] [c.h.w.s.s.i.DisableLoginStrategy]: Login will be disabled for 5 seconds after 3 failed login att
empts. For every 3 consecutive failed login attempts, disable period will be multiplied by 10.
2025-03-21 13:48:58,757 [ INFO] [main] [c.h.w.m.i.r.RocksDBLoader]: '/tmp' is used as RocksDB shared library directory
2025-03-21 13:48:59,773 [ INFO] [main] [c.h.w.m.i.r.RocksDBLoader]: '/tmp' is used as RocksDB shared library directory
2025-03-21 13:49:12,780 [ INFO] [main] [c.h.w.MCApplication]: Started MCApplication in 44.003 seconds (process running for 48.329)
2025-03-21 13:49:12,834 [ INFO] [main] [c.h.w.MCApplication]:

Hazelcast Management Center successfully started at http://localhost:8080/
```

Підключимо кластер з 3 нодами

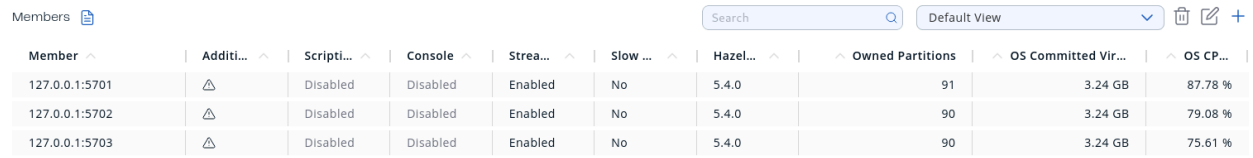



```
2025-03-21 13:51:20,954 [ INFO] [MC-Client-dev.lifecycle-1] [c.h.w.s.MCClientManager]: MC Client connected to cluster dev.
2025-03-21 13:51:20,955 [ INFO] [MC-Client-dev.lifecycle-1] [c.h.w.s.MCClientManager]: Started communication with member: Member [127.0.
0.1]:5703 - ed525daa-c2d5-49a2-a153-0ebe6fdc02d3
2025-03-21 13:51:20,955 [ INFO] [MC-Client-dev.lifecycle-1] [c.h.w.s.MCClientManager]: Started communication with member: Member [127.0.
0.1]:5702 - fc40ef29-3013-4c8e-aa05-219cb6b748b8
2025-03-21 13:51:20,958 [ INFO] [MC-Client-dev.lifecycle-1] [c.h.w.s.MCClientManager]: Started communication with member: Member [127.0.
0.1]:5701 - 840b5a5d-4867-4648-9653-a16a28437dfb
```

Dashboard:



Members:



3. Продемонструйте роботу Distributed Map використовуючи API на будь-якій мові яка має клієнт для Hazelcast, створіть Distributed Map

- запишіть в неї 1000 значень з ключами від 0 до 1000
- за допомогою Management Center подивиться на розподіл ключів по нодах

```
┌──(kali㉿kali)-[~/lab2]
└─$ python3 task3-5.py
Inserted: 1
Inserted: 2
Inserted: 3
Inserted: 4
Inserted: 5
```

Maps

Search | Default View | Show System Maps OFF

Name	Persistence	In Memory F...	Entries	Entry Memory	Backup Mem...	Hits	Locks	Dirty Entries	Event Jc
distributed-map	Disabled	BINARY	1,000	121.09 kB	121.09 kB	0	0	0	Disable

Map Statistics (In-Memory Format: BINARY) | RESET TIME | 1 minute ago → now | Default View

Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events	Hi
127.0.0.1:5701	344	0	344	0	0	41.66 kB	0	
127.0.0.1:5702	333	0	333	0	0	40.32 kB	0	
127.0.0.1:5703	323	0	323	0	0	39.11 kB	0	
TOTAL	1,000	0	1,000	0	0	121.09 kB	0	

1 – 3 of 3 Rows 10

- подивитись як зміниться розподіл даних по нодах:
  - якщо відключити одну ноду

```
Members {size:2, ver:4} [
        Member [127.0.0.1]:5701 - 840b5a5d-4867-4648-9653-a16a28437dfb this
        Member [127.0.0.1]:5703 - ed525daa-c2d5-49a2-a153-0ebe6fdc02d3
]
```

Map Statistics (In-Memory Format: BINARY) | RESET TIME | 1 minute ago → now | Default View

Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events	Hi
127.0.0.1:5701	518	0	344	0	0	62.73 kB	0	
127.0.0.1:5703	482	0	323	0	0	58.37 kB	0	
TOTAL	1,000	0	667	0	0	121.09 kB	0	

  - відключити послідовно дві ноди

```
Members {size:1, ver:5} [
        Member [127.0.0.1]:5701 - 840b5a5d-4867-4648-9653-a16a28437dfb this
]
```

Map Statistics (In-Memory Format: BINARY) | RESET TIME | 1 minute ago → now | Default View

Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events	Hi
127.0.0.1:5701	1,000	0	344	0	0	121.09 kB	0	
TOTAL	1,000	0	344	0	0	121.09 kB	0	

1 – 1 of 1 Rows 10

  - відключити одночасно дві ноди (емулюючи “падіння” серверів, чи використовуючи команду kill -9)

Map Statistics (In-Memory Format: BINARY) | RESET TIME | 1 minute ago → now | Default View

Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events	Hi
127.0.0.1:5701	344	0	344	0	0	41.66 kB	0	
127.0.0.1:5702	333	0	0	0	0	40.32 kB	0	
127.0.0.1:5703	323	0	0	0	0	39.11 kB	0	
TOTAL	1,000	0	344	0	0	121.09 kB	0	

1 – 3 of 3 Rows 10

```
kali      239586 79.2  4.7 3352336 143412 pts/0  Sl+  17:20   0:05 /usr/bin/java -cp /home/kali/Downloads/hazelcast-5.4.0/lib:/home/kal
kali      239654 56.6  3.8 3352316 114432 pts/1  Sl+  17:20   0:02 /usr/bin/java -cp /home/kali/Downloads/hazelcast-5.4.0/lib:/home/kal
kali      239738 50.0  0.1   9948  4768 pts/4    R+   17:20   0:00 ps aux

┌──(kali㉿kali)-[~/lab2]
└─$ sudo kill -9 239586 239654
```

Map Statistics (In-Memory Format: BINARY) | RESET TIME | 1 minute ago → now | Default View

Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events	Hi
127.0.0.1:5701	671	0	344	0	0	81.25 kB	0	
TOTAL	671	0	344	0	0	81.25 kB	0	

1 – 1 of 1 Rows 10

- Чи буде втрата даних?

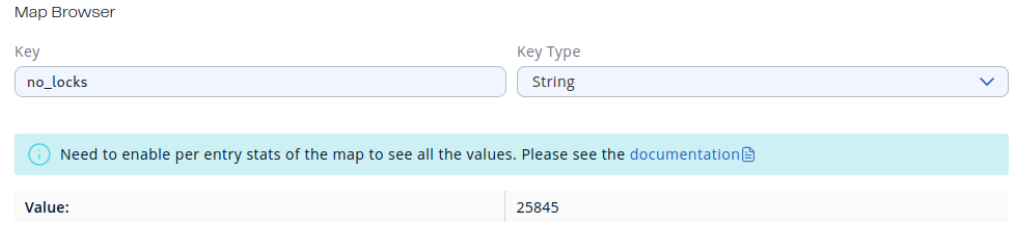
Якщо резервні копії (backup-count) не налаштовані або їх недостатньо, при одночасному падінні кількох нод може статися втрата даних, як бачимо, у ситуації з одночасних відключенням двох нод ми втратили велику кількість даних.

- Яким чином зробити щоб не було втрати даних?

Збільшити backup-count (наприклад, до 2), увімкнути Persistence, додати більше нод у кластер або налаштувати Partition Groups.

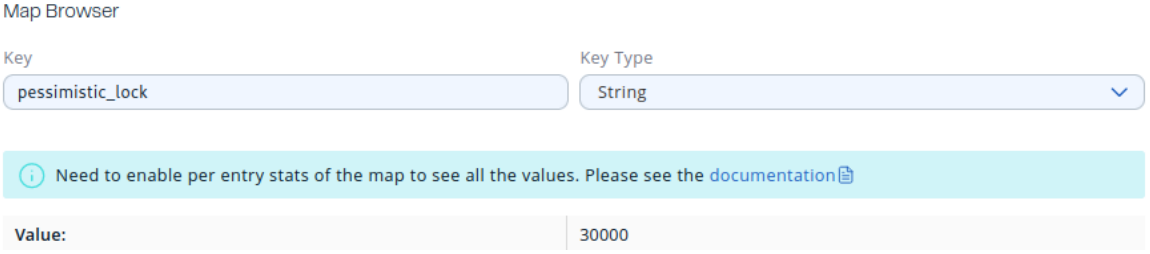
4. Продемонструйте роботу Distributed Map without locks
   - використовуючи 3 клієнта, на кожному з них одночасно запустіть інкремент значення для одного й того самого ключа в циклі на 10К ітерацій:
   - подивиться яке кінцеве значення для ключа “key” буде отримано (чи вийде 30К?)

```
┌──(kali㉿kali)-[~/lab2]
└─$ python3 lab2.py & python3 lab2.py & python3 lab2.py
[1] 251137
[2] 251138
Final value (No Locks): 18807
Time taken: 13.5962 seconds

┌──(kali㉿kali)-[~/lab2]
└─$ Final value (No Locks): 25424
Time taken: 24.1674 seconds

[1]  - done       python3 lab2.py
┌──(kali㉿kali)-[~/lab2]
└─$ Final value (No Locks): 25845
Time taken: 24.9612 seconds

[2]  + done       python3 lab2.py
```



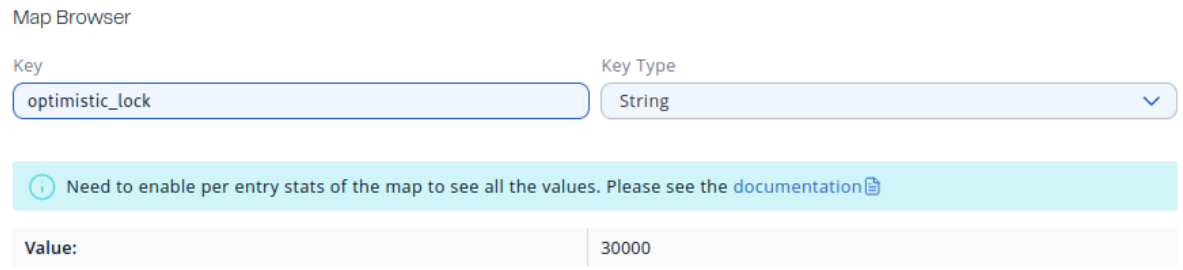
5. Зробіть те саме з використанням песимістичним блокування та поміряйте час:

```
┌──(kali㉿kali)-[~/lab2]
└─$ python3 lab2.py & python3 lab2.py & python3 lab2.py
[1] 253150
[2] 253151
Final value (Pessimistic Locking): 22041
Time taken: 68.1011 seconds

┌──(kali㉿kali)-[~/lab2]
└─$ Final value (Pessimistic Locking): 29938
Time taken: 83.7305 seconds

[2]  + done       python3 lab2.py
┌──(kali㉿kali)-[~/lab2]
└─$ Final value (Pessimistic Locking): 30000
Time taken: 84.1260 seconds
```



6. Зробіть те саме з використанням оптимістичним блокуванням та поміряйте час:

```
┌──(kali㉿kali)-[~/lab2]
└─$ python3 lab2.py & python3 lab2.py & python3 lab2.py
[1] 254762
[2] 254763
Final value (Optimistic Locking): 12103
Time taken: 18.2511 seconds

┌──(kali㉿kali)-[~/lab2]
└─$ Final value (Optimistic Locking): 29285
Time taken: 45.3758 seconds

[1]  - done       python3 lab2.py
┌──(kali㉿kali)-[~/lab2]
└─$ Final value (Optimistic Locking): 30000
Time taken: 46.5873 seconds

[2]  + done       python3 lab2.py
```



7. Порівняйте результати кожного з запусків

- для реалізації без блокувань маєте спостерігату втрату даних; для реалізації з песимістичним та оптимістичним блокуванням мають бути однакові результати

Під час використання Distributed Map without locks **не отримали** частину інкрементів (результат 25845), оскільки одночасний доступ кількох потоків до одного ключа призводив до умов гонки. А для песимічного та оптимістичного блокувань **отримали** 30000 ітерацій, бо механізми блокування гарантували цілісність даних,

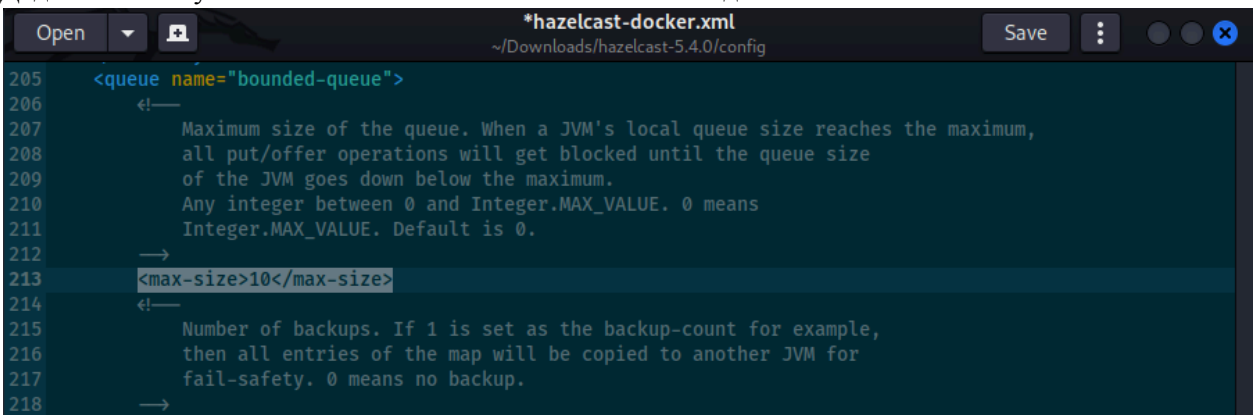
- песимістичний чи оптимістичний підхід працює швидше?

Оптимістичний підхід працює майже у **2 рази швидше**, ніж песимістичний, оскільки він не утримує блокування протягом усього часу оновлення, а перевіряє версію значення перед записом.

8. Робота з Bounded queue

- на основі Distributed Queue налаштуйте Bounded queue на 10 елементів

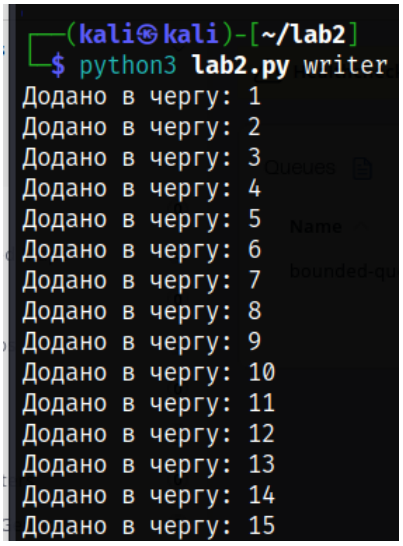
Додамо налаштування в hazelcast-docker.xml на кожній ноді:

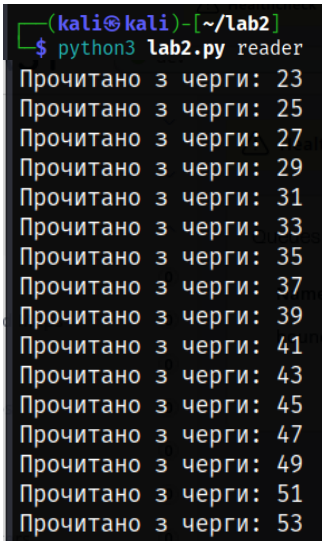


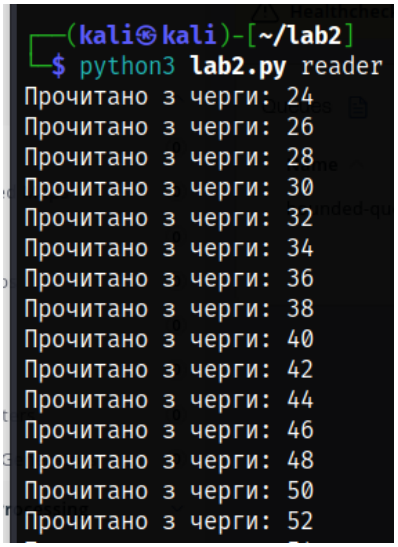
Та перезапустимо ноди, щоб змінити налаштування.

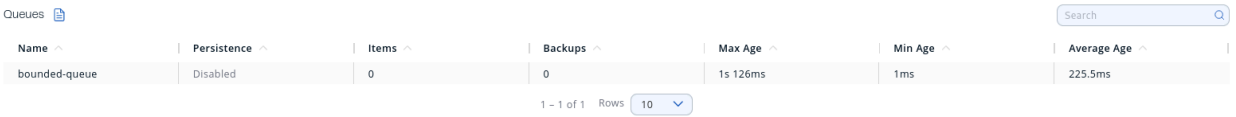
- запустіть одного клієнта який буде писати в чергу значення 1..100, а двох інших які будуть читати з черги
- під час вичитування, кожне повідомлення має вичитуватись одразу
- яким чином будуть вичитуватись значення з черги двома клієнтами?

Значення з черги будуть читатись по черзі двома клієнтами. Оскільки обидва клієнти, звертаються до однієї і тієї ж черги, вони будуть забирати елементи один за одним, тому елементи розподілятимуться між ними.



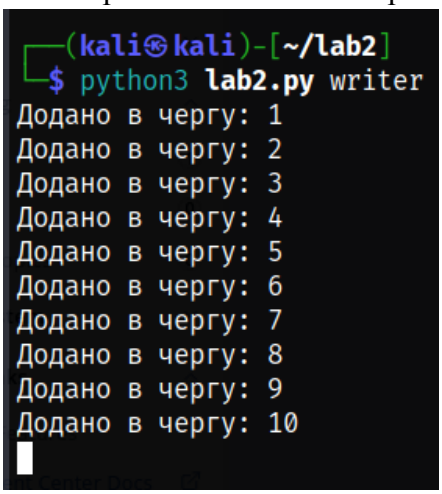




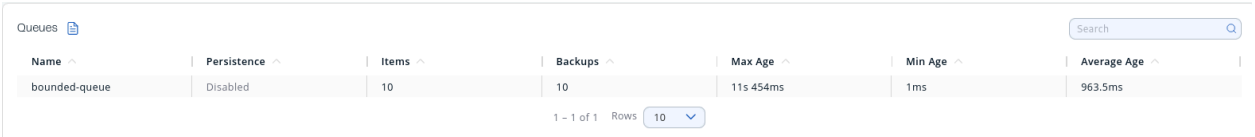


- перевірте яка буде поведінка на запис якщо відсутнє читання, і черга заповнена

Запис припиняється на номері черги, вказаному в <max-size>.:



Якщо черга заповнена і немає читання, операція запису блокується до того моменту, поки не з'явиться вільне місце в черзі (поки елемент не буде вилучений). Це допомагає запобігти втраті даних.



Код:

```
import hazelcast
import time
from hazelcast import client

CONFIG = {
    "cluster_name": "dev",
    "cluster_members": [],
}

def task3():
    client = hazelcast.HazelcastClient(**CONFIG)
    try:
        distributed_map = client.get_map("distributed-map").blocking()
        for index in range(1, 1001):
            distributed_map.put(index, index)
            print(f"Inserted: {index}")
    finally:
        client.shutdown()

def task4():
    client = hazelcast.HazelcastClient(**CONFIG)
    key = "no_locks"
    distributed_map = client.get_map("distributed-map").blocking()

    if not distributed_map.contains_key(key):
        distributed_map.put(key, 0)

    start_time = time.time()
    for _ in range(10_000):
        value = distributed_map.get(key)
        distributed_map.put(key, value + 1)
    end_time = time.time()

    print(f"Final value (No Locks): {distributed_map.get(key)}")
    print(f"Time taken: {end_time - start_time:.4f} seconds")

    client.shutdown()

def task5():
    client = hazelcast.HazelcastClient(**CONFIG)
    key = "pessimistic_lock"
    distributed_map = client.get_map("distributed-map").blocking()

    if not distributed_map.contains_key(key):
        distributed_map.put(key, 0)

    start_time = time.time()
    for _ in range(10_000):
        distributed_map.lock(key)
        try:
            value = distributed_map.get(key)
            distributed_map.put(key, value + 1)
        finally:
            distributed_map.unlock(key)
    end_time = time.time()

    print(f"Final value (Pessimistic Locking): {distributed_map.get(key)}")
    print(f"Time taken: {end_time - start_time:.4f} seconds")

    client.shutdown()

def task6():
    client = hazelcast.HazelcastClient(**CONFIG)
    key = "optimistic_lock"
```

```
    distributed_map = client.get_map("distributed-map").blocking()

    if not distributed_map.contains_key(key):
        distributed_map.put(key, 0)

    start_time = time.time()
    for _ in range(10_000):
        while True:
            old_value = distributed_map.get(key)
            new_value = old_value + 1
            if distributed_map.replace_if_same(key, old_value, new_value):
                break
    end_time = time.time()

    print(f"Final value (Optimistic Locking): {distributed_map.get(key)}")
    print(f"Time taken: {end_time - start_time:.4f} seconds")

    client.shutdown()

def writer():
    client = hazelcast.HazelcastClient(**CONFIG)
    queue = client.get_queue("bounded-queue").blocking()

    try:
        for value in range(1, 101):
            queue.put(value)
            print(f"Додано в чергу: {value}")
            time.sleep(0.1)
    finally:
        client.shutdown()

def reader():
    client = hazelcast.HazelcastClient(**CONFIG)
    queue = client.get_queue("bounded-queue").blocking()

    try:
        while True:
            item = queue.take()
            print(f"Прочитано з черги: {item}")
    finally:
        client.shutdown()

if __name__ == "__main__":
    import sys
    if len(sys.argv) < 2:
        print("Вкажіть 'writer', 'reader', або одну з task3-task6 для запуску відповідного
клієнта.")
    elif sys.argv[1] == "writer":
        writer()
    elif sys.argv[1] == "reader":
        reader()
    elif sys.argv[1] == "task3":
        task3()
    elif sys.argv[1] == "task4":
        task4()
    elif sys.argv[1] == "task5":
        task5()
    elif sys.argv[1] == "task6":
        task6()
    else:
        print("Неправильний аргумент. Використовуйте 'writer', 'reader', 'task3', 'task4',
'task5' або 'task6'.")
```