

Lab 2

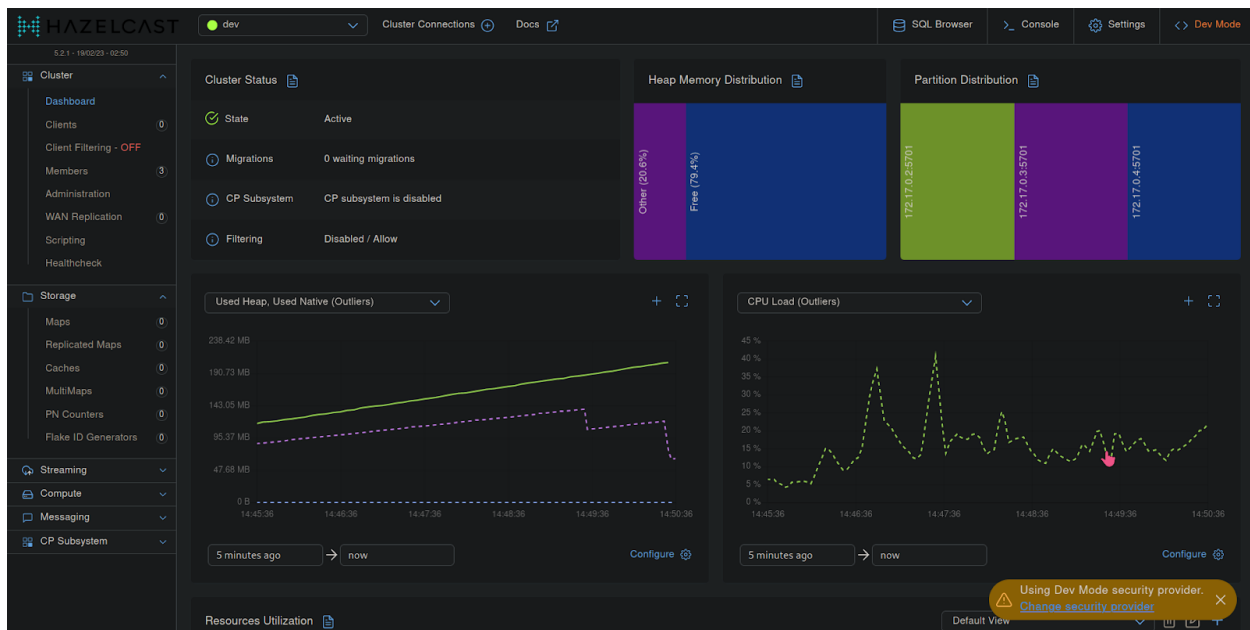
Виконав *Осман Труш*

Github

https://github.com/Adeon18/MicroServices/tree/hazelcast_basics

1. Підняв 3 ноди, як окремі застосування та запустив менеджмент центр(він тут зліва знизу). Під'єднався до інтерфесу менеджмент центру

```
e=0, executor.q.async.size=0, executor.q.client.size=0, executor.q.client.query.size=0,
executor.q.client.blocking.size=0, executor.q.query.size=0, executor.q.scheduled.size=0,
executor.q.io.size=0, executor.q.system.size=0, executor.q.operations.size=0, executor.
q.priorityOperation.size=0, operations.completed.count=32, executor.q.mapLoad.size=0, ex
ecutor.q.mapLoadAllKeys.size=0, executor.q.cluster.size=0, executor.q.response.size=0, o
perations.running.count=0, operations.pending.invocations.percentage=0.00%, operations.p
ending.invocations.count=0, proxy.count=1, clientEndpoint.count=0, connection.active.cou
nt=2, client.connection.count=0, connection.count=2
2023-02-19 12:31:44,130 [ INFO] [hz.conscending_chatelet.priority-generic-operation.th
read-0] [c.h.c.i.p.t.AuthenticationMessageTask]: [172.17.0.2]:5701 [dev] [5.2.2] Receive
d auth from Connection[id=3, /172.17.0.2:5701->/172.17.0.5:41301, qualifier=null, endpoi
nt=[172.17.0.5]:41301, remoteUid=b9d379a9-c7c0-43a5-9070-82cea58ba978, alive=true, conn
ectionType=MCJVM, planeIndex=-1], successfully authenticated, clientUid: b9d379a9-c7c0-
43a5-9070-82cea58ba978, client name: MC-Client-dev, client version: 5.2.1
2023-02-19 12:31:51,280 [ INFO] [hz.conscending_chatelet.priority-generic-operation.th
read-0] [c.h.c.i.p.t.AuthenticationMessageTask]: [172.17.0.2]:5701 [dev] [5.2.2] Receive
d auth from Connection[id=4, /172.17.0.2:5701->/172.17.0.5:58963, qualifier=null, endpoi
nt=[172.17.0.5]:58963, remoteUid=436c023f-ce7f-47f2-9028-a1abc6914bb8, alive=true, conn
ectionType=MCJVM, planeIndex=-1], successfully authenticated, clientUid: 436c023f-ce7f-
47f2-9028-a1abc6914bb8, client name: MC-Client-0-dev, client version: 5.2.1
[
null, endpoint=[172.17.0.5]:38229, remoteUid=436c023f-ce7f-47f2-9028-a1abc6914bb
8, alive=true, connectionType=MCJVM, planeIndex=-1], successfully authenticated,
clientUid: 436c023f-ce7f-47f2-9028-a1abc6914bb8, client name: MC-Client-0-dev, c
lient version: 5.2.1
2023-02-19 12:40:44,588 [ INFO] [hz.serene_carver.HealthMonitor] [c.h.i.d.HealthM
onitor]: [172.17.0.3]:5701 [dev] [5.2.2] processors=8, physical.memory.total=15.3
G, physical.memory.free=1.3G, swap.space.total=16.0G, swap.space.free=16.0G, heap
.memory.used=68.1M, heap.memory.free=397.9M, heap.memory.total=466.0M, heap.memor
y.max=12.3G, heap.memory.used/total=14.58%, heap.memory.used/max=0.54%, minor.gc.
count=21, minor.gc.time=149ms, major.gc.count=0, major.gc.time=0ms, load.process=
0.00%, load.system=25.00%, load.systemAverage=1.21, thread.count=74, thread.peakC
ount=77, cluster.timeDiff=0, event.q.size=0, executor.q.async.size=0, executor.q.
client.size=0, executor.q.client.query.size=0, executor.q.client.blocking.size=0,
executor.q.query.size=0, executor.q.scheduled.size=0, executor.q.io.size=0, execu
tor.q.system.size=0, executor.q.operations.size=0, executor.q.priorityOperation.
size=0, operations.completed.count=29251, executor.q.mapLoad.size=0, executor.q.m
apLoadAllKeys.size=0, executor.q.cluster.size=0, executor.q.response.size=0, oper
ations.running.count=0, operations.pending.invocations.percentage=0.00%, operatio
ns.pending.invocations.count=0, proxy.count=3, clientEndpoint.count=2, connection
.active.count=5, client.connection.count=2, connection.count=4
[
2023-02-19 12:31:52,310 [ WARN] [MetricsConsumer-1] [c.h.w.m.j.JetClusterMetricsStore]:
Received non-monotonic timestamp from Member [172.17.0.2]:5701 - 35bdcf2-2212-4fe0-a1ff
-0515e4c900c5. previous=1676809911864 current=1676809911864
2023-02-19 12:31:52,336 [ WARN] [MetricsConsumer-0] [c.h.w.m.j.JetClusterMetricsStore]:
Received non-monotonic timestamp from Member [172.17.0.3]:5701 - 5ecbb86b-8fd0-4586-bb3f
-13b79d6622bf. previous=1676809911419 current=1676809911419
2023-02-19 12:31:52,337 [ WARN] [MetricsConsumer-1] [c.h.w.m.j.JetClusterMetricsStore]:
Received non-monotonic timestamp from Member [172.17.0.4]:5701 - c3ffe625-ab3b-41a3-a2a8
-775fa881f442. previous=1676809908994 current=1676809908994
2023-02-19 12:31:54,019 [ WARN] [MetricsConsumer-0] [c.h.w.m.j.JetClusterMetricsStore]:
Received non-monotonic timestamp from Member [172.17.0.4]:5701 - c3ffe625-ab3b-41a3-a2a8
-775fa881f442. previous=1676809913994 current=1676809913994
[
8, alive=true, connectionType=MCJVM, planeIndex=-1], successfully authenticated,
clientUid: 436c023f-ce7f-47f2-9028-a1abc6914bb8, client name: MC-Client-0-dev, c
lient version: 5.2.1
2023-02-19 12:38:24,570 [ INFO] [hz.jovial_napier.HealthMonitor] [c.h.i.d.HealthM
onitor]: [172.17.0.4]:5701 [dev] [5.2.2] processors=8, physical.memory.total=15.3
G, physical.memory.free=1.6G, swap.space.total=16.0G, swap.space.free=16.0G, heap
.memory.used=50.2M, heap.memory.free=469.8M, heap.memory.total=520.0M, heap.memor
y.max=12.3G, heap.memory.used/total=9.62%, heap.memory.used/max=0.40%, minor.gc.c
ount=17, minor.gc.time=156ms, major.gc.count=0, major.gc.time=0ms, load.process=0
.00%, load.system=37.50%, load.systemAverage=1.34, thread.count=77, thread.peakCo
unt=77, cluster.timeDiff=0, event.q.size=0, executor.q.async.size=0, executor.q.c
lient.size=0, executor.q.client.query.size=0, executor.q.client.blocking.size=0,
executor.q.query.size=0, executor.q.scheduled.size=0, executor.q.io.size=0, execu
tor.q.system.size=0, executor.q.operations.size=0, executor.q.priorityOperation.s
ize=0, operations.completed.count=22819, executor.q.mapLoad.size=0, executor.q.ma
pLoadAllKeys.size=0, executor.q.cluster.size=0, executor.q.response.size=0, opera
tions.running.count=0, operations.pending.invocations.percentage=0.00%, operation
s.pending.invocations.count=0, proxy.count=3, clientEndpoint.count=2, connection
.active.count=5, client.connection.count=2, connection.count=4
[
```



2. Запускаю код для запуску 1000 значень в "my-distributed-map"

Map Statistics (In-Memory Format: BINARY)

RESET TIME 1 minute ago → now Default View

Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events
172.17.0.2:5701	325	0	325	0	0	42.47 kB	0
172.17.0.3:5701	352	0	352	0	0	45.98 kB	0
172.17.0.4:5701	323	0	323	0	0	42.20 kB	0
TOTAL	1000	0	1000	0	0	130.64 kB	0

1 - 3 of 3 Rows 10

- Дані +- рівномірно розподілилися по трьох мапах, при пошуку ключів по мапі, отримую правильний результат.

Map Browser

Key Key Type

Value:	865
Memory Cost:	67.00 B
Expiration Time:	N/A
Last Access Time:	N/A
Last Stored Time:	N/A
Time to Live:	Unlimited
Key Owner Member:	172.17.0.3:5701
Class:	java.lang.String
Creation Time:	N/A
Hits:	-1
Last Update Time:	N/A
Version:	0
Max Idle:	Unlimited

Cancel **BROWSE**

- Зупиняю першу мапу - дані не втрапилися. Браузинг по контентях мапи далі працює.

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

Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events
172.17.0.3:5701	545	0	352	0	0	71.20 kB	0
172.17.0.4:5701	455	0	323	0	0	59.44 kB	0
TOTAL	1000	0	675	0	0	130.64 kB	0

- Зупиняю другу мапу - дані не втрапилися, всі дані зберігаються в одній ноді.

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

Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events
172.17.0.4:5701	1000	0	323	0	0	130.64 kB	0
TOTAL	1000	0	323	0	0	130.64 kB	0

3. Продемонструйте роботу Distributed Map with locks

Я запускатииму багатопоточну програму написану на C++ і з 3 потоків одночасно писатиму в мапу. Програма є на гілці `hazelcast_basics` а саме `task2_datarace.cpp` там

можна вибрати яку опцію(з яким локінгом взяти), та заранити.

- Без локінгу - бачимо результат далекий від 3000, а саме 1223.

```
Starting threads...
Thread 0 Started
Thread 1 Started
Thread 2 Started
Thread 0 Finished
Thread 1 Finished
Thread 2 Finished
Finished! Result = 1223
Done
```

- З песимістичним локінгом - бачимо що 3000 елементів попадають в мапу

```
Starting threads...
Thread 0 Started
Thread 1 Started
Thread 2 Started
Thread 0 Finished
Thread 1 Finished
Thread 2 Finished
Finished! Result = 3000
Done
```

- З оптимістичним локінгом - також всі 3000 елементів попадають у мапу.

```
Starting threads...
Thread 0 Started
Thread 1 Started
Thread 2 Started
Thread 0 Finished
Thread 1 Finished
Thread 2 Finished
Finished! Result = 3000
Done
```

Документація по тому як це запустити та відтворити є в проєкті

4. Налаштуйте Bounded queue

- з однієї ноди (клієнта) йде запис, а на двох інших читання

Один клієнт записує кожну 0.1 секунду(тотал записує 20 елементів), два клієнта читають кожних 2 секунди (читають по 10 елементів кожен), ліміт черги - 10.

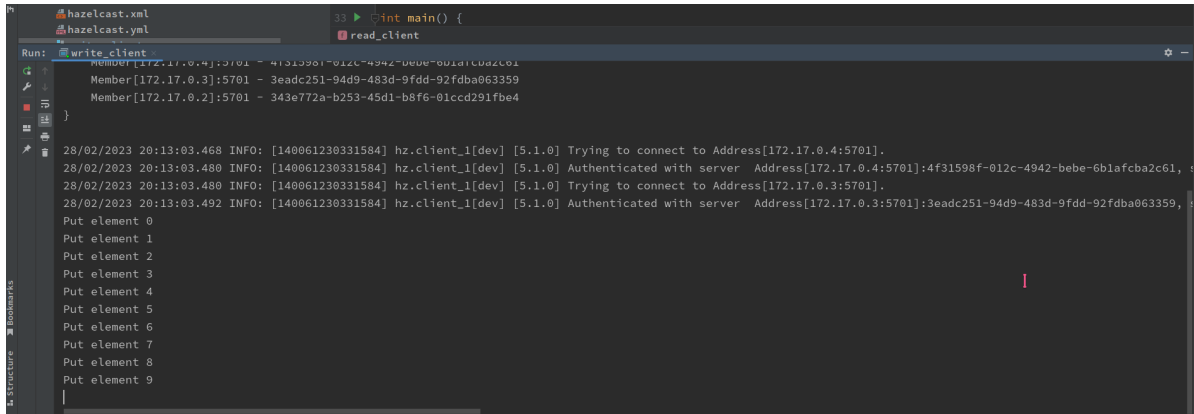
```
28/02/2023 22:40:19.544 INFO: [14
Thread Write 1: Put element 0
Thread Read 2: Took element 0
Thread Read 1: Took element 1
Thread Write 1: Put element 1
Thread Write 1: Put element 2
Thread Write 1: Put element 3
Thread Write 1: Put element 4
Thread Write 1: Put element 5
Thread Write 1: Put element 6
Thread Write 1: Put element 7
Thread Write 1: Put element 8
Thread Write 1: Put element 9
Thread Write 1: Put element 10
Thread Write 1: Put element 11
Thread Write 1: Put element 12
Thread Read 2: Took element 2
Thread Read 1: Took element 3
Thread Write 1: Put element 13
Thread Write 1: Put element 14
Thread Read 2: Took element 4
Thread Read 1: Took element 5
```

```
↓ Thread Read 1: Took element 5
⏮ Thread Write 1: Put element 15
⏮ Thread Write 1: Put element 16
🖨 Thread Read 2: Took element 6
🗑 Thread Read 1: Took element 7
Thread Write 1: Put element 17
Thread Read 2: Took element 8
Thread Write 1: Put element 18
Thread Read 1: Took element 9
Thread Write 1: Put element 19
28/02/2023 22:40:27.760 INFO: [1405
28/02/2023 22:40:27.769 INFO: [1405
28/02/2023 22:40:27.769 INFO: [1405
28/02/2023 22:40:27.769 INFO: [1405
28/02/2023 22:40:27.769 INFO: [1405
28/02/2023 22:40:27.774 INFO: [1405
Thread Read 2: Took element 10
Thread Read 1: Took element 11
Thread Read 2: Took element 12
Thread Read 1: Took element 13
Thread Read 2: Took element 14
Thread Read 1: Took element 15
Thread Read 2: Took element 16
Thread Read 1: Took element 17
Thread Read 2: Took element 18
Thread Read 1: Took element 19
28/02/2023 22:40:39.567 INFO: [1405
28/02/2023 22:40:39.567 INFO: [1405
```

Можна побачити що елементи поклалися, ліміт черги досягнувся, і тоді вже тредами що читають витягуються елементи, як тільки витягнулися 2 елементи, з'явилося вільне місце, і туди той тред що пише зразу значення записав і забив чергу. Оскільки тред запису записує 20 значень, а треди читання читають по 10 жоден тред не лочиться

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

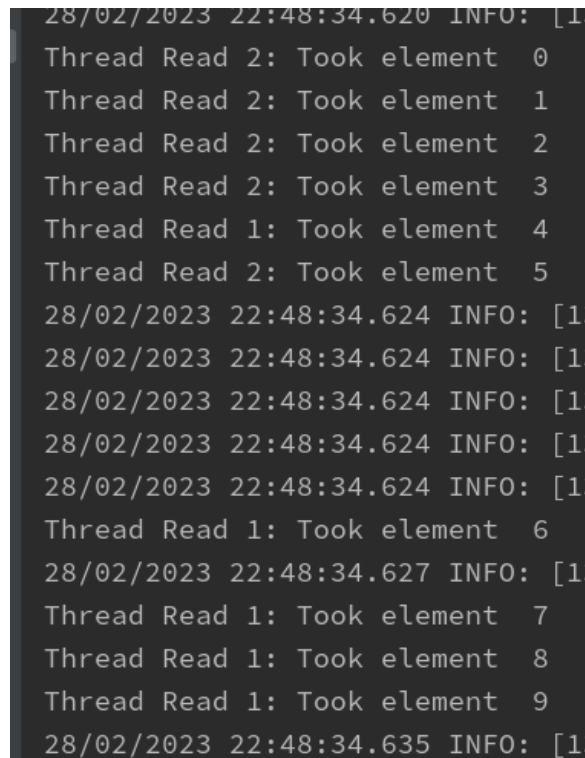
При записі з одного потоку без читання в queue, при спробі запису 100 значень в чергу, ліміт якої 10 - мапа блокується, та при наступному колі до put потік лочиться та чекає допоки в черзі звільниться місце.



```
hazelcast.xml
hazelcast.yml
Run: write_client
Member[172.17.0.4]:5701 - 4f31598f-012c-4942-bebe-6b1afcba2c61
Member[172.17.0.3]:5701 - 3eadc251-94d9-483d-9fdd-92fdbae63359
Member[172.17.0.2]:5701 - 343e772a-b253-45d1-b8f6-01ccd291fba4
}
}
28/02/2023 20:13:03.468 INFO: [140061230331584] haz.client_l[dev] [5.1.0] Trying to connect to Address[172.17.0.4:5701].
28/02/2023 20:13:03.480 INFO: [140061230331584] haz.client_l[dev] [5.1.0] Authenticated with server Address[172.17.0.4:5701]:4f31598f-012c-4942-bebe-6b1afcba2c61.
28/02/2023 20:13:03.480 INFO: [140061230331584] haz.client_l[dev] [5.1.0] Trying to connect to Address[172.17.0.3:5701].
28/02/2023 20:13:03.492 INFO: [140061230331584] haz.client_l[dev] [5.1.0] Authenticated with server Address[172.17.0.3:5701]:3eadc251-94d9-483d-9fdd-92fdbae63359.
Put element 0
Put element 1
Put element 2
Put element 3
Put element 4
Put element 5
Put element 6
Put element 7
Put element 8
Put element 9
```

- як будуть вичитуватись значення з черги якщо є декілька читачів

В чергу було запхано 10 елементів, і 2 потоки(клієнти) читання їх звіди читають одночасно(без зупинки на таймері). Результат наступний.



```
28/02/2023 22:48:34.620 INFO: [1
Thread Read 2: Took element 0
Thread Read 2: Took element 1
Thread Read 2: Took element 2
Thread Read 2: Took element 3
Thread Read 1: Took element 4
Thread Read 2: Took element 5
28/02/2023 22:48:34.624 INFO: [1
28/02/2023 22:48:34.624 INFO: [1
28/02/2023 22:48:34.624 INFO: [1
28/02/2023 22:48:34.624 INFO: [1
28/02/2023 22:48:34.624 INFO: [1
Thread Read 1: Took element 6
28/02/2023 22:48:34.627 INFO: [1
Thread Read 1: Took element 7
Thread Read 1: Took element 8
Thread Read 1: Took element 9
28/02/2023 22:48:34.635 INFO: [1
```

