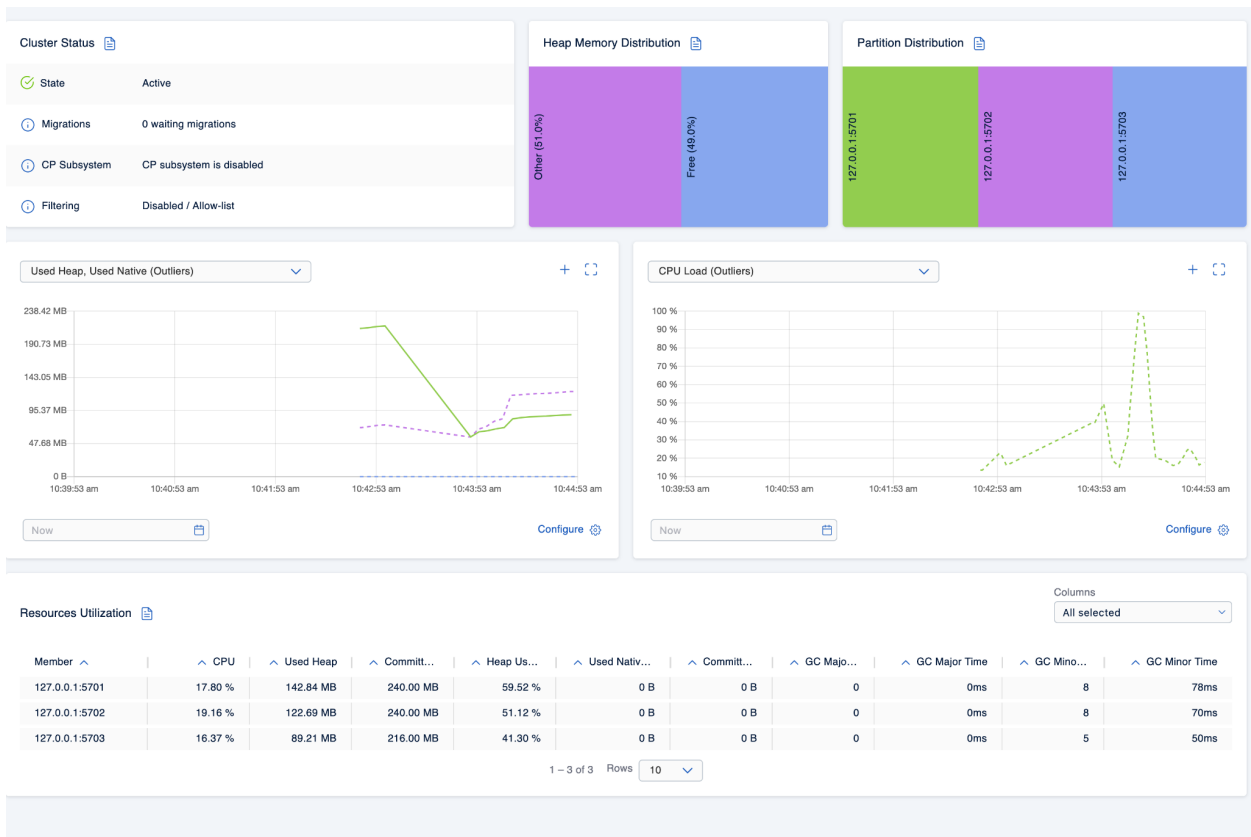


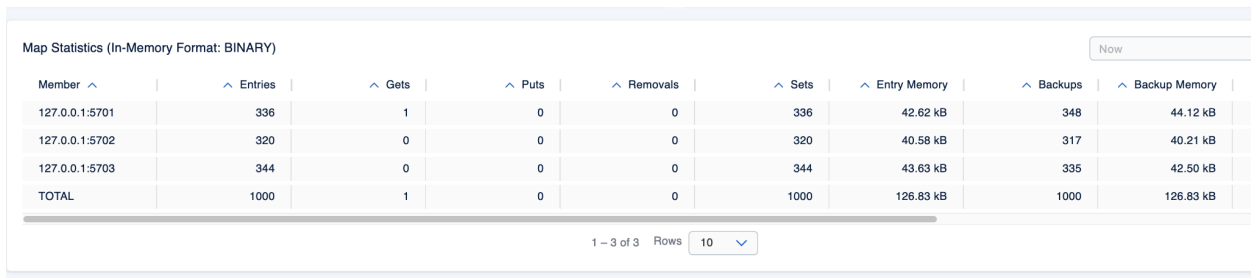
Authors: Maxym Kuzyshyn, Nazarii Kuspys

TASK 2, 3 nodes and management cluster created:



TASK 3, Creating the distributed map

- 1. Added 1000 values to the map



Бачимо що дані розподілені рівномірно (40-45 Кб на ноду)

2. Behavior after suddenly removing 3-rd cluster:

Map Statistics (In-Memory Format: BINARY) Now

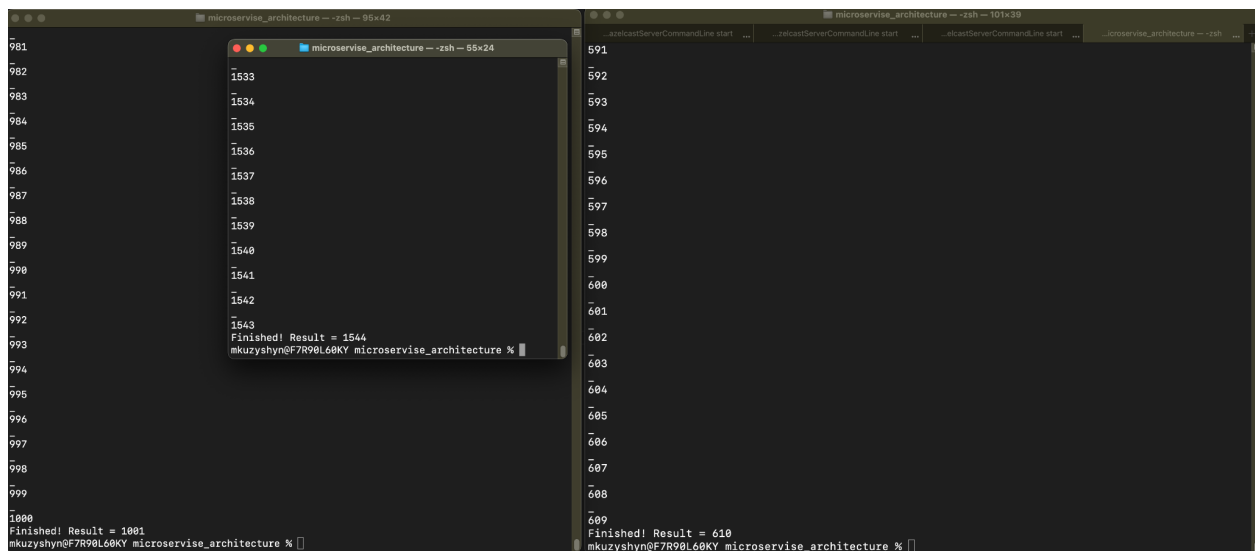
Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Backups	Backup Memory
127.0.0.1:5701	513	1	0	0	336	65.06 kB	487	61.77 kB
127.0.0.1:5702	487	0	0	0	320	61.77 kB	513	65.06 kB
TOTAL	1000	1	0	0	656	126.83 kB	1000	126.83 kB

1 - 2 of 2 Rows 10

Бачимо що втрата даних не відбулась адже кожен байт даних одного з трьох кластерів був реплікований 1 раз на інший, тому при видаленні третього кластера, перші два змогли зарекаверити інформацію.

TASK 4, Map Locks

1. No lock



```
microservice_architecture -- zsh -- 95x42
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
Finished! Result = 1544
mkuzyszyn@F7R90L60KY microservice_architecture %

microservice_architecture -- zsh -- 101x39
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
Finished! Result = 610
mkuzyszyn@F7R90L60KY microservice_architecture %
```

Бачимо що сума трьох підрахунків не співпала з реальним значенням через race issue

2. Pessimistic locking

```
2237
2240
2243
2246
2249
2252
2255
2258
2261
2264
2267
2270
2273
^CTraceback (most recent call last):
  File "map_locks/pessimistic_lock.py", line 31, in
    main()
  File "map_locks/pessimistic_lock.py", line 17, in main
    distributed_map_lock(key)
  File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast
line 325, in f
    return result.result()
  File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast
line 58, in result
    self._event.wait()
  File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast
line 191, in wait
    self.condition.wait()
  File "/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.8/lib/python3.8/threading.py", line 302, in wait
    waiter.acquire()
KeyboardInterrupt
mkuzyshyn@F7R90L60KY microservice_architecture % python3 map_locks/pessimistic_lock.py
at: 0
at: 100
at: 200
at: 300
at: 400
at: 500
at: 600
at: 700
at: 800
at: 900
Finished! Result = 2696
mkuzyshyn@F7R90L60KY microservice_architecture %
```

```
2274
^CTraceback (most recent call last):
  File "map_locks/pessimistic_lock.py", line 31, in <module>
    main()
  File "map_locks/pessimistic_lock.py", line 17, in main
    distributed_map_lock(key)
  File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast
line 325, in f
    return result.result()
  File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast
line 58, in result
    self._event.wait()
  File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast
line 191, in wait
    self.condition.wait()
  File "/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.8/lib/python3.8/threading.py", line 302, in wait
    waiter.acquire()
KeyboardInterrupt
mkuzyshyn@F7R90L60KY microservice_architecture % python3 map_locks/pessimistic_lock.py
at: 0
at: 100
at: 200
at: 300
at: 400
at: 500
at: 600
at: 700
at: 800
at: 900
Finished! Result = 3000
mkuzyshyn@F7R90L60KY microservice_architecture %
```

3. Optimistic locking

```
time.sleep(0.01) # 10 milliseconds
KeyboardInterrupt
mkuzyshyn@F7R90L60KY microservice_architecture % python3 map_locks/optimistic_lock.py
at: 0
at: 100
at: 200
at: 300
at: 400
at: 500
at: 600
at: 700
at: 800
at: 900
Finished! Result = 4226
mkuzyshyn@F7R90L60KY microservice_architecture %
mkuzyshyn@F7R90L60KY microservice_architecture %
mkuzyshyn@F7R90L60KY microservice_architecture %
at: 0
at: 100
at: 200
at: 300
at: 400
at: 500
at: 600
at: 700
at: 800
at: 900
Finished! Result = 1899
mkuzyshyn@F7R90L60KY microservice_architecture %
mkuzyshyn@F7R90L60KY microservice_architecture %
mkuzyshyn@F7R90L60KY microservice_architecture %
at: 0
at: 100
at: 200
at: 300
at: 400
at: 500
at: 600
at: 700
at: 800
at: 900
Finished! Result = 2513
mkuzyshyn@F7R90L60KY microservice_architecture %
```

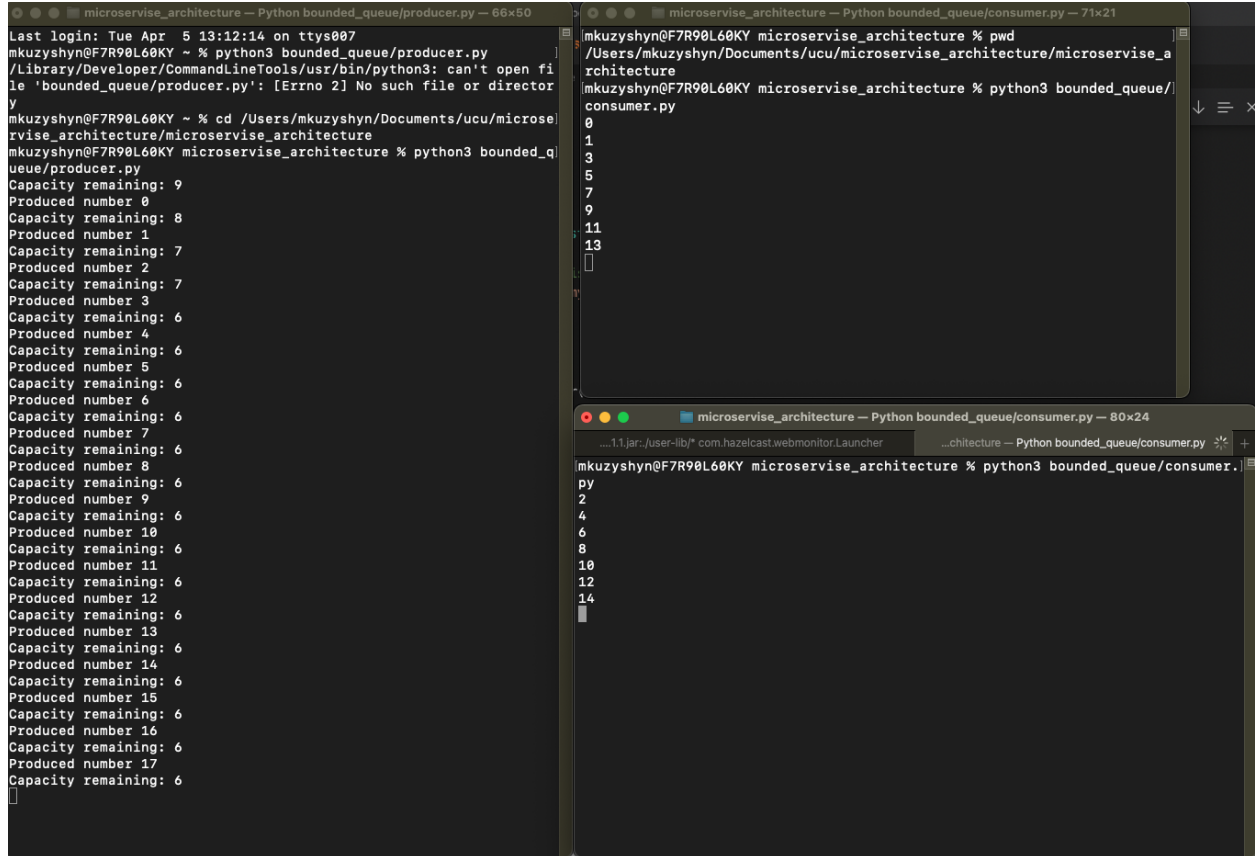
```
File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast/future.py",
line 325, in f
    return result.result()
  File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast/future.py",
line 58, in result
    self._event.wait()
  File "/Users/mkuzyshyn/Library/Python/3.8/lib/python/site-packages/hazelcast/future.py",
line 191, in wait
    self.condition.wait()
  File "/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3
/lib/python3.8/threading.py", line 302, in wait
    waiter.acquire()
KeyboardInterrupt
mkuzyshyn@F7R90L60KY microservice_architecture % python3 map_locks/optimistic_lock.py
at: 0
at: 100
at: 200
at: 300
at: 400
at: 500
at: 600
at: 700
at: 800
at: 900
Finished! Result = 4844
mkuzyshyn@F7R90L60KY microservice_architecture % python3 map_locks/optimistic_lock.py
at: 0
at: 100
at: 200
at: 300
at: 400
at: 500
at: 600
at: 700
at: 800
at: 900
Finished! Result = 2075
mkuzyshyn@F7R90L60KY microservice_architecture %
```

Бачимо що Optimistic lock та Pessimistic lock справилися з задачею без гонитв даних (data race)

TASK 5, Bounded queue

Capacity of the bounded queue is set to 10 in xml config file

1. One write client, two read clients



```
microservice_architecture — Python bounded_queue/producer.py — 66x50
Last login: Tue Apr 5 13:12:14 on ttys007
mkuzyshyn@F7R90L60KY ~ % python3 bounded_queue/producer.py
/Library/Developer/CommandLineTools/usr/bin/python3: can't open file
'bounded_queue/producer.py': [Errno 2] No such file or director
y
mkuzyshyn@F7R90L60KY ~ % cd /Users/mkuzyshyn/Documents/ucu/microse
rvice_architecture/microservice_architecture
mkuzyshyn@F7R90L60KY microservice_architecture % python3 bounded_q
ueue/producer.py
Capacity remaining: 9
Produced number 0
Capacity remaining: 8
Produced number 1
Capacity remaining: 7
Produced number 2
Capacity remaining: 7
Produced number 3
Capacity remaining: 6
Produced number 4
Capacity remaining: 6
Produced number 5
Capacity remaining: 6
Produced number 6
Capacity remaining: 6
Produced number 7
Capacity remaining: 6
Produced number 8
Capacity remaining: 6
Produced number 9
Capacity remaining: 6
Produced number 10
Capacity remaining: 6
Produced number 11
Capacity remaining: 6
Produced number 12
Capacity remaining: 6
Produced number 13
Capacity remaining: 6
Produced number 14
Capacity remaining: 6
Produced number 15
Capacity remaining: 6
Produced number 16
Capacity remaining: 6
Produced number 17
Capacity remaining: 6
[]

microservice_architecture — Python bounded_queue/consumer.py — 71x21
mkuzyshyn@F7R90L60KY microservice_architecture % pwd
/Users/mkuzyshyn/Documents/ucu/microservice_architecture
mkuzyshyn@F7R90L60KY microservice_architecture % python3 bounded_queue/
consumer.py
0
1
3
5
7
9
11
13
[]

microservice_architecture — Python bounded_queue/consumer.py — 80x24
...1.1.jar:/user-lib/* com.hazelcast.webmonitor.Launcher
...chitecture — Python bounded_queue/consumer.py
mkuzyshyn@F7R90L60KY microservice_architecture % python3 bounded_queue/consumer.py
py
2
4
6
8
10
12
14
[]
```

We can see that the capacity of the queue is constant. Read is every 2 seconds, and write is every second. We have 2 consumers, so equilibrium in capacity of the queue is remained.

One consumer reads only odd numbers, while the second- only even numbers

2. No read, and queue is full:

```
mkuzyshyn@F7R90L60KY microservice_architecture % python3 bounded_queue/producer.py
Capacity left: 9
Capacity left: 8
Capacity left: 7
Capacity left: 6
Capacity left: 5
Capacity left: 4
Capacity left: 3
Capacity left: 2
Capacity left: 1
Queue is full, waiting for customers to read
Queue is full, waiting for customers to read
Queue is full, waiting for customers to read
Queue is full, waiting for customers to read
Queue is full, waiting for customers to read
```

Бачимо що коли черга заповнюється повністю, то нові значення не записуються, та продюсер чекає щоб звільнилось місце для запису