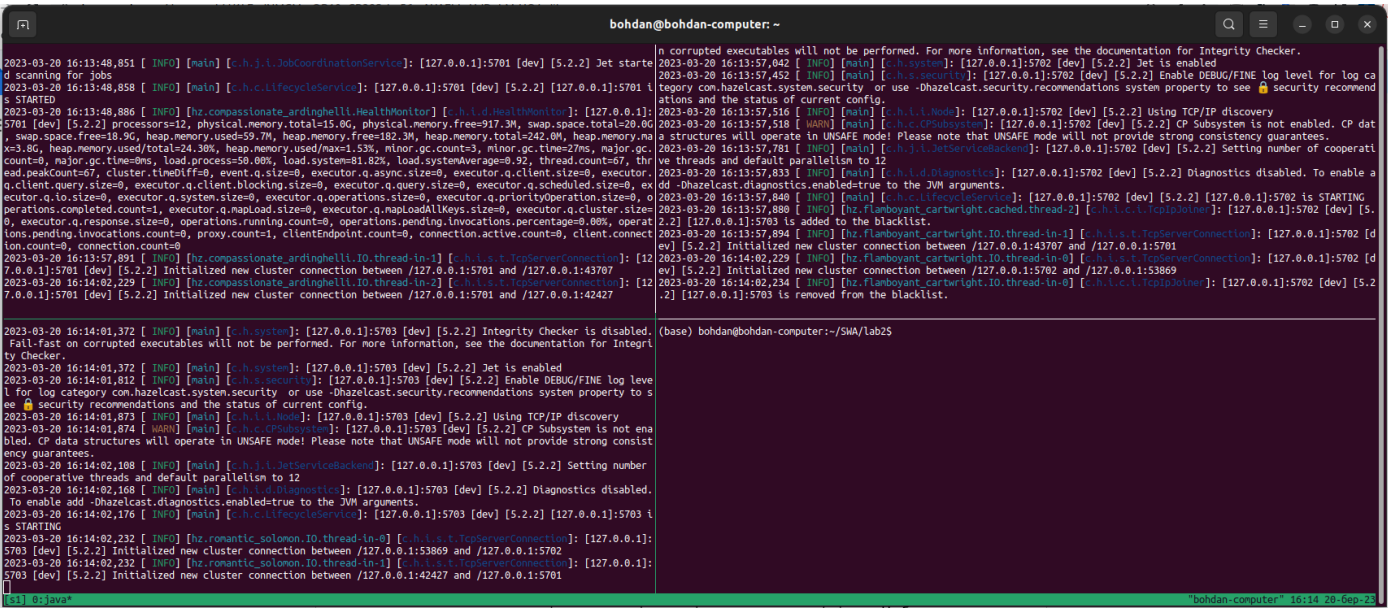


Protocol: lab2

Author: Bohdan Mahometa
Github branch: <https://github.com/bogdanmagometa/SWAms/tree/hazelcast>

2. Configure 3 instances



3. Demonstrate usage of Distributed Map

- write 1000 values into a distributed map:

Output of command:



Hazelcast logs:

```
bohdan@bohdan-computer: ~
Member [127.0.0.1]:5701 - 3279487c-4f41-48dd-8c8c-e568b961f377 this
Member [127.0.0.1]:5702 - 95291bbc-3cf2-47eb-bfdf-a305f7ee53ce
Member [127.0.0.1]:5703 - 0832c2c1-fc8b-45a0-95ba-05cf3e743afb

2023-03-20 16:25:58,955 [ INFO] [hz.mystifying.ganguly.priority-generic-operation.thread-0] [c.h.i.p.t.PartitionSt
orJobs]: [127.0.0.1]:5701 [dev] [5.2.2] Initializing cluster partition table arrangement...
2023-03-20 16:26:00,513 [ INFO] [hz.mystifying.ganguly.priority-generic-operation.thread-0] [c.h.c.i.p.t.Authentic
ationMessageTask]: [127.0.0.1]:5701 [dev] [5.2.2] Received auth from Connection[id=3, /127.0.0.1:5701->/127.0.0.1:
45174, qualifier=null, endpoint=[127.0.0.1]:45174, remoteUid=76c16738-a0a9-44b6-9b6a-ac833e1a17c6, alive=true, co
nnectionType=PVH, planeIndex=-1], successfully authenticated, clientUid: 76c16738-a0a9-44b6-9b6a-ac833e1a17c6, cl
ient name: hz.client_0, client version: 5.1
2023-03-20 16:26:02,013 [ INFO] [hz.mystifying.ganguly.io.thread-in-0] [c.h.l.s.t.TcpServerConnection]: [127.0.0.1
]:5701 [dev] [5.2.2] Connection[id=3, /127.0.0.1:5701->/127.0.0.1:45174, qualifier=null, endpoint=[127.0.0.1]:4517
4, remoteUid=76c16738-a0a9-44b6-9b6a-ac833e1a17c6, alive=false, connectionType=PVH, planeIndex=-1] closed. Reason:
Connection closed by the other side
2023-03-20 16:26:02,018 [ INFO] [hz.mystifying.ganguly.event-5] [c.h.c.i.ClientEndpointManager]: [127.0.0.1]:5701
[dev] [5.2.2] Destroying ClientEndpoint[connection=Connection[id=3, /127.0.0.1:5701->/127.0.0.1:45174, qualifie
r=null, endpoint=[127.0.0.1]:45174, remoteUid=76c16738-a0a9-44b6-9b6a-ac833e1a17c6, alive=false, connectionType=PVH,
planeIndex=-1], clientUid=76c16738-a0a9-44b6-9b6a-ac833e1a17c6, clientName=hz.client_0, authenticated=true, cli
entVersion=5.1, creationTime=1679322360511, latest clientAttributes=null, labels=[]]
2023-03-20 16:25:58,998 [ INFO] [hz.xenodochial.payne.generic-operation.thread-0] [c.h.c.i.ClientEndpointManager]:
[127.0.0.1]:5703 [dev] [5.2.2] The address ([127.0.0.1]:5701) will be added as a temporary member address to the
CP-IP joiner configuration.
2023-03-20 16:25:58,996 [ INFO] [main] [c.h.j.l.JobCoordinationService]: [127.0.0.1]:5703 [dev] [5.2.2] Jet starte
d scanning for jobs
2023-03-20 16:25:59,910 [ INFO] [main] [c.h.c.l.LifecycleService]: [127.0.0.1]:5703 [dev] [5.2.2] [127.0.0.1]:5703 i
s STARTED
2023-03-20 16:26:00,532 [ INFO] [hz.xenodochial.payne.priority-generic-operation.thread-0] [c.h.c.i.p.t.Authentic
ationMessageTask]: [127.0.0.1]:5703 [dev] [5.2.2] Received auth from Connection[id=3, /127.0.0.1:5703->/127.0.0.1:4
5454, qualifier=null, endpoint=[127.0.0.1]:45454, remoteUid=76c16738-a0a9-44b6-9b6a-ac833e1a17c6, alive=true, con
nectionType=PVH, planeIndex=-1], successfully authenticated, clientUid: 76c16738-a0a9-44b6-9b6a-ac833e1a17c6, cli
ent name: hz.client_0, client version: 5.1
2023-03-20 16:26:02,013 [ INFO] [hz.xenodochial.payne.io.thread-in-2] [c.h.l.s.t.TcpServerConnection]: [127.0.0.1
]:5703 [dev] [5.2.2] Connection[id=3, /127.0.0.1:5703->/127.0.0.1:45454, qualifier=null, endpoint=[127.0.0.1]:45454
, remoteUid=76c16738-a0a9-44b6-9b6a-ac833e1a17c6, alive=false, connectionType=PVH, planeIndex=-1] closed. Reason:
Connection closed by the other side
2023-03-20 16:26:02,019 [ INFO] [hz.xenodochial.payne.event-5] [c.h.c.i.ClientEndpointManager]: [127.0.0.1]:5703
[dev] [5.2.2] Destroying ClientEndpoint[connection=Connection[id=3, /127.0.0.1:5703->/127.0.0.1:45454, qualifie
r=null, endpoint=[127.0.0.1]:45454, remoteUid=76c16738-a0a9-44b6-9b6a-ac833e1a17c6, alive=false, connectionType=PVH,
planeIndex=-1], clientUid=76c16738-a0a9-44b6-9b6a-ac833e1a17c6, clientName=hz.client_0, authenticated=true, cli
entVersion=5.1, creationTime=1679322360529, latest clientAttributes=null, labels=[]]
(base) bohdan@bohdan-computer: /usr/lib/hazelcast-management-center$
```

- look into distribution of values in Management Center:

as we can see, the entries are distributed almost equally:

Map Statistics (In-Memory Format: BINARY)										
RESET TIME 1 minute ago → now Default View										
Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events	Hits		
127.0.0.1:5701	326	0	326	0	0	41.99 kB	0	0		
127.0.0.1:5702	328	0	328	0	0	42.24 kB	0	0		
127.0.0.1:5703	346	0	346	0	0	44.57 kB	0	0		
TOTAL	1000	0	1000	0	0	128.80 kB	0	0		

- remove nodes

After removing one node, the entries were not lost:

Map Statistics (In-Memory Format: BINARY)										
RESET TIME 1 minute ago → now Default View										
Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events	Hits		
127.0.0.1:5701	510	0	326	0	0	65.70 kB	0	0		
127.0.0.1:5702	490	0	328	0	0	63.10 kB	0	0		
TOTAL	1000	0	654	0	0	128.80 kB	0	0		

After removing two nodes, the entries were not lost:

Map Statistics (In-Memory Format: BINARY)										
RESET TIME 1 minute ago → now Default View										
Member	Entries	Gets	Puts	Removals	Sets	Entry Memory	Events	Hits		
127.0.0.1:5701	1000	0	326	0	0	128.80 kB	0	0		
TOTAL	1000	0	326	0	0	128.80 kB	0	0		

4. Demonstrate working with Distributed Map with locks

```

• (base) bohdan@bohdan-computer:~/SWA/lab2$ python3 locking_strategies.py
Without locking:
  counter value: 1000
  time taken: 11.75 seconds

With pessimistic lock:
  counter value: 3000
  time taken: 37.65 seconds

With optimistic lock:
  counter value: 3000
  time taken: 35.06 seconds

○ (base) bohdan@bohdan-computer:~/SWA/lab2$ 

```

In the first case, we have race condition and loss of data, so we do not get 3000.

In case of pessimistic and optimistic locks, there are no race conditions. Optimistic lock seems to be a little bit faster, but we haven't proved it statistically.

5. Configure Bounded Queue

- When there are no reads (0 consumers), the put operation blocks:

```

(base) bohdan@bohdan-computer:~/SWA/lab2$ python3 bqueue.py
Putting 0
Putting 1
Putting 2
Putting 3
Putting 4
Putting 5
Putting 6
Putting 7
Putting 8
Putting 9
Putting 10
□

```

- When there is at least one consumer, the bounded queue might block the provider until reads occur:

```

• (base) bohdan@bohdan-computer:~/SWA/Lab2$ python3 bqueue.py
Putting 0
Putting 1
Putting 2
Putting 3
Putting 4
Putting 5
Putting 6
Putting 7
Putting 8
      Reading 9
      Reading 0

Putting 9
Putting 10
Putting 11
Putting 12
Putting 13
      Reading 1
      Reading 2

Putting 14
Putting 15
      Reading 3
      Reading 4

Putting 16
Putting 17
      Reading 5
      Reading 6

Putting 18
Putting 19
      Reading 7
      Reading 8
      Reading 9
      Reading 10
      Reading 11
      Reading 12
      Reading 13
      Reading 14
      Reading 15
      Reading 16
      Reading 18
      Reading 17
      Reading 19

○ (base) bohdan@bohdan-computer:~/SWA/Lab2$ █

```

or the consumers might be fast enough to keep the queue not full:

```

• (base) bohdan@bohdan-computer:~/SWA/Lab2$ python3 bqueue.py
Putting 0
      Reading 0

Putting 1
      Reading 1

Putting 2
      Reading 2

Putting 3
      Reading 3

Putting 4
      Reading 4

Putting 5
      Reading 5

Putting 6
      Reading 6

Putting 7
      Reading 7

Putting 8
      Reading 8

Putting 9
      Reading 9

Putting 10
      Reading 10

Putting 11
      Reading 11

Putting 12
      Reading 12

Putting 13
      Reading 13

Putting 14
      Reading 14

Putting 15
      Reading 15

Putting 16
      Reading 16

Putting 17
      Reading 17

Putting 18
      Reading 18

Putting 19
      Reading 19

○ (base) bohdan@bohdan-computer:~/SWA/Lab2$ █

```

