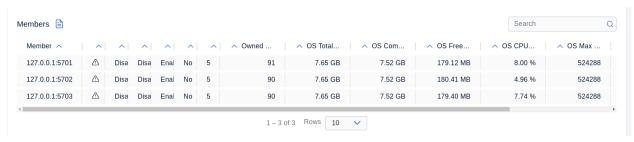
# https://github.com/monberin/microservises\_architecture/tree/hw2



### Продемонструйте роботу Distributed Map

- використовуючи API створіть Distributed Map
- запишіть в неї 1000 значень з ключем від 0 до 1к
- подивитись як зміниться розподіл якщо відключити одну ноду/дві ноди. Чи буде втрата даних?

#### 3 ноди:



### 2 ноди:



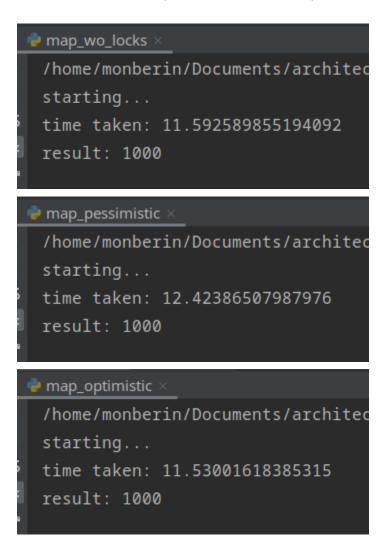
## 1 нода:



Втрати даних немає, дані кожного разу діляться +- симетрично

Продемонструйте роботу Distributed Map with locks

- використовуючи 3 підключення (чи підключення з 3х клієнтів) одночансо запустіть приклади за посиланням з документації з підрахунку значень в циклі: а) без блокування; б) з песимістичним; в) з оптимістичним блокуванням
- порівняйте результати кожного з запусків





Race condition-у не спостерігається навіть при версії без локів, проте необхідно уникати таких ситуацій.

Варіант з оптимістичним блокуванням найшвидший.

# Налаштуйте Bounded queue

- на основі Distributed Queue налаштуйте Bounded queue
- з однієї ноди (клієнта) йде запис, а на двох інших читання
- перевірте яка буде поведінка на запис якщо відсутнє читання, і черга заповнена
- як будуть вичитуватись значення з черги якщо є декілька читачів



```
 > ~/Documents/architecture/hw2_hazelcast python3 ./writer.py
running
writing...
no place left
no place left
ended writing
  > > /Documents/architecture/hw2_hazelcast
 python3 ./reader.py
reading...
taken 13_value
taken 15_value
taken 17_value
taken 19_value
taken 21_value
taken 23_value
taken 25_value
taken 27_value
taken 29_value
taken 31_value
taken 33_value
taken 35_value
taken 37_value
taken 39_value
```

```
 > /Documents/architecture/hw2_hazelcast  python3 ./reader.py
reading...
taken 1_value
taken 2_value
taken 3_value
taken 4_value
taken 5_value
taken 6_value
taken 7_value
taken 8_value
taken 9_value
taken 10_value
taken 11_value
taken 12_value
taken 14_value
taken 16_value
```

Якщо йде запис, але не відбувається читання, то дані записуватись не будуть. Якщо декілька читачів, то вони вичитують різні значення, не повторюючись.