КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Звіт до лабораторної роботи з курсу «Хмарні Обчислення»

> Студента групи ТТП-41 Маркова Максима Юрійовича

> > викладач: Дерев'янченко Олександр Валерійович

1. Реалізація

Створюємо новий проєкт та робимо його активним Вмикаємо compute APIs та додаємо billing account до проєкту Перевіряємо існуючі регіони: **gcloud compute regions list**

```
texttospeech460@cloudshell:~ (cloudcomputing-456217)$ gcloud compute regions list
NAME: africa-south1
CPUS: 0/24
DISKS_GB: 0/4096
ADDRESSES: 0/8
RESERVED ADDRESSES: 0/8
STATUS: UP
TURNDOWN DATE:
NAME: asia-east1
CPUS: 0/24
DISKS GB: 0/4096
ADDRESSES: 0/8
RESERVED ADDRESSES: 0/8
STATUS: UP
TURNDOWN DATE:
NAME: asia-east2
CPUS: 0/24
DISKS_GB: 0/4096
ADDRESSES: 0/8
RESERVED ADDRESSES: 0/8
STATUS: UP
TURNDOWN DATE:
NAME: asia-northeast1
CPUS: 0/24
DISKS GB: 0/4096
ADDRESSES: 0/8
RESERVED_ADDRESSES: 0/8
STATUS: UP
TURNDOWN DATE:
NAME: asia-northeast2
```

Оберемо europe-north1: gcloud config set compute/region europe-north1

texttospeech460@cloudshell:~ (cloudcomputing-456217)\$ gcloud config set compute/region europe-north1 WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].

Аналогічно виведемо список зон: gcloud compute zones list

```
texttospeech460@cloudshell: (cloudcomputing-456217)$ gcloud compute zones list
NAME: us-east1-b
REGION: us-east1
STATUS: UP
NEXT MAINTENANCE:
TURNDOWN DATE:
NAME: us-east1-c
REGION: us-east1
STATUS: UP
NEXT MAINTENANCE:
TURNDOWN DATE:
NAME: us-east1-d
REGION: us-east1
STATUS: UP
NEXT MAINTENANCE:
TURNDOWN DATE:
NAME: us-east4-c
REGION: us-east4
STATUS: UP
NEXT MAINTENANCE:
TURNDOWN DATE:
NAME: us-east4-b
REGION: us-east4
STATUS: UP
NEXT MAINTENANCE:
TURNDOWN DATE:
NAME: us-east4-a
REGION: us-east4
STATUS: UP
```

Оберемо europe-north1-a: gcloud config set compute/zone europe-north1-a

```
texttospeech460@cloudshell:~ (cloudcomputing-456217)$ gcloud config set compute/zone europe-north1-a
MARNING: Property validation for compute/zone was skipped.
Updated property [compute/zone].
```

Налаштовуємо firewall: gcloud compute firewall-rules create allow-all --direction=INGRESS --priority=1000 --network=default --action=ALLOW

```
texttospesch460@cloudshell: (aloudcomputing-456217)$ gcloud compute firewall-rules create allow-all --direction=INGRESS --priority=100 tr-action=ALLOW --rules=all --source-ranges=0.0.0.0/0 Creating firewall...done.

NAME: allow-all
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 1000
ALLOW: all
DENY:
```

Створюємо master VM: gcloud compute instances create-with-container master

```
--container-image=registry.hub.docker.com/hummer12007/parcs-node
--container-env PARCS ARGS="master"
```

```
Created [https://www.googleapis.com/compute/v1/projects/cloudcomputing-456218/zones/europe-north1-a/instances/master].

NAME: master

ZONE: europe-north1-a

MACHINE TYPE: n1-standard-1

PREEMPTIBLE:
INTERNAL_IP: 10.166.0.2

EXTERNAL_IP: 35.228.239.157

STATUS: RUNNING
```

Створюємо worker VM: gcloud compute instances create-with-container worker1

--container-image=registry.hub.docker.com/hummer12007/parcs-node --container-env PARCS ARGS="worker 10.166.0.2"

```
Created [https://www.googleapis.com/compute/v1/projects/cloudcomputing-456218/zones/europe-north1-a/instances/worker1].

NAME: worker1

ZONE: europe-north1-a

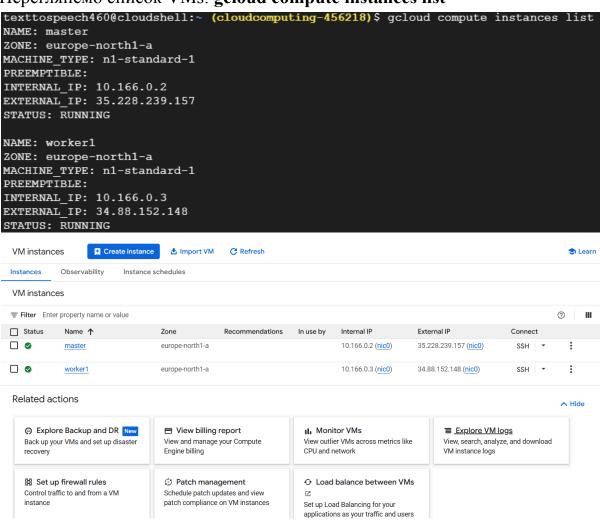
MACHINE TYPE: n1-standard-1

PREEMPTIBLE:
INTERNAL_IP: 10.166.0.3

EXTERNAL_IP: 34.88.152.148

STATUS: RUNNING
```

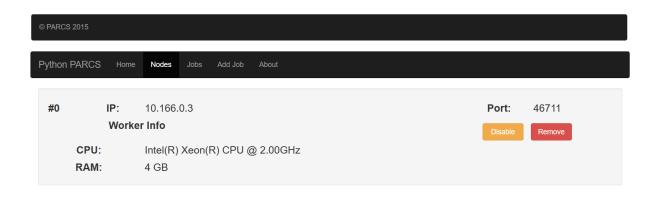
Переглянемо список VMs: gcloud compute instances list



Тепер, перейшовши за http://35.228.239.157:8080 маємо доступ до веб-інтерфейсу:

Python PARCS Home Nodes Jobs Add Job About

PARCS home page.

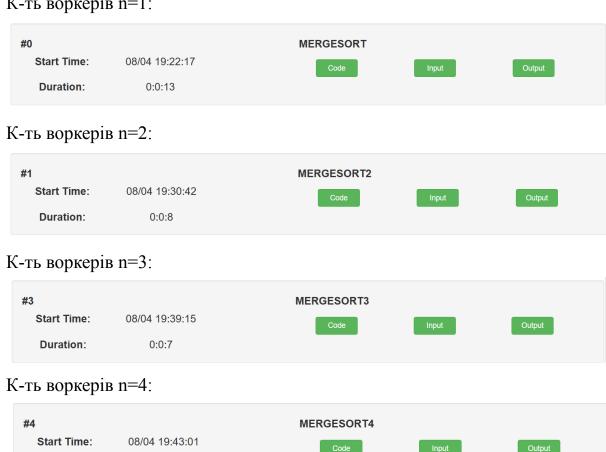


© PARCS 2015

2. Швидкодія

Проведемо заміри швидкодії на різних кількостях воркерів з одним вхідним файлом input.txt (n=1000000)

К-ть воркерів n=1:

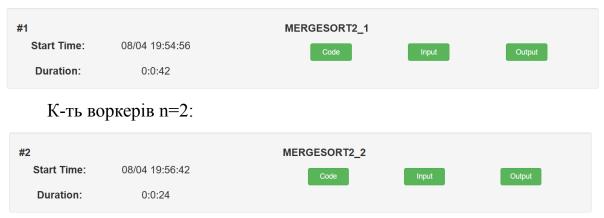


Для файлу n=3000000:

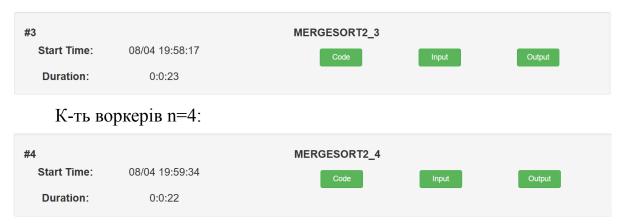
Duration:

К-ть воркерів n=1:

0:0:8



К-ть воркерів n=3:



Бачимо пришвидшення продуктивності при збільшенні кількості воркерів. При чому, "критичне число" (число, після якого швидкодія тільки спадає; витрати на комунікацію між воркерами та мастером більші, ніж виграши у швидкодії внаслідок більшого числа воркерів) воркерів залежить від розмірів вхідних даних.

Висновки

В результаті виконання лабораторної роботи було розроблено функціональний розподілений застосунок для сортування масивів з використанням алгоритму MergeSort у середовищі Руго4. Реалізація використовує принцип MapReduce, де сортування окремих частин масиву виконується паралельно на декількох воркерах, після чого результати зливаються в єдиний відсортований масив.

Такий підхід дозволив значно скоротити час обробки великих обсягів даних за рахунок розпаралелювання процесу сортування.

У ході реалізації було виконано наступні етапи:

- Реалізовано зчитування вхідних даних з файлу та збереження результату у вихідний файл.
- Масив розбито на частини відповідно до кількості воркерів.
- Запроваджено виклики для розподіленого сортування частин масиву (тар-фаза).
- Реалізовано поетапне злиття відсортованих частин у єдиний масив (reduce-фаза).
- Проведено аналіз продуктивності.

Перспективами розвитку проєкту ϵ :

• Розширення підтримки різних алгоритмів сортування (наприклад, QuickSort, HeapSort).

Розроблений додаток може бути використаний як приклад застосування технологій розподілених обчислень для обробки великих даних, зокрема для задач попереднього аналізу, обробки логів або підготовки даних у високонавантажених системах.

Додатки

```
from Pyro4 import expose
      class Solver:
        def init (self, workers=None, input file name=None,
output file name=None):
           self.input file name = input file name
           self.output file name = output file name
           self.workers = workers
        def solve(self):
           arr = self.read input()
           chunk size = len(arr) / len(self.workers)
           mapped = []
           for i in xrange(len(self.workers)):
             start = int(i * chunk size)
             end = int((i + 1) * chunk size) if i != len(self.workers) - 1 else
len(arr)
             chunk = arr[start:end]
             mapped.append(self.workers[i].mymap(chunk))
           result = self.myreduce(mapped)
           self.write output(result)
        @staticmethod
        @expose
        def mymap(subarray):
```

```
return sorted(subarray)
@staticmethod
@expose
def myreduce(mapped):
  sorted_lists = [m.value for m in mapped]
  def merge(left, right):
    merged = []
    i = j = 0
     while i < len(left) and j < len(right):
       if left[i] < right[j]:</pre>
          merged.append(left[i])
          i += 1
       else:
          merged.append(right[j])
         i += 1
     merged.extend(left[i:])
     merged.extend(right[j:])
     return merged
  while len(sorted lists) > 1:
     temp = []
     for i in xrange(0, len(sorted_lists), 2):
       if i + 1 < len(sorted lists):
          merged = merge(sorted lists[i], sorted lists[i+1])
```

```
else:
    merged = sorted_lists[i]
    temp.append(merged)
    sorted_lists = temp
    return sorted_lists[0]

def read_input(self):
    f = open(self.input_file_name, 'r')
    arr = map(int, f.readline().split())
    f.close()
    return arr

def write_output(self, output):
    f = open(self.output_file_name, 'w')
    f.write(' '.join(map(str, output)) + '\n')
    f.close()
```