Celery on Docker Swarm

Wei Lin

Abstract

In the territory of Python, Celery is a famous distributed task queue framework, its Canvas mechanism is specially powerful in building complex workflow.

Celery can function in distributed environment and go perfectly along with Docker-Swarm: Docker-Swarm provides a cluster environment and the worker containers to sustain Celery; worker containers and processes and be dynamically scaled and expanded to fulfill Celery's need, work together parallelly to accomplish the computation.

With Docker-Swarm , a cluster will be built upon two Raspberry Pi machines. Hadoop entry-level "Word Count" program will be re-writen in Python and executed parallelly via Celery on the cluster.

Steps:

1. Establish a Docker Swarm:

With reference to the article "Let Docker Swarm all over your Raspberry Pi Cluster (http://blog.hypriot.com/post/let-docker-swarm-all-over-your-raspberry-pi-cluster/)", a Docker-Swarm had been built upon two Raspberry Pi.

The swarm is composited fo two Docker machines:

- host rpi202(192.168.0.114) as Swarm Manager，the Docker machine name is "master01".
- host rpi201(192.168.0.109) as Swarm Node，the Docker machine name is "node01".

In [ ]:

```
HypriotOS: pi@rpi202 in ~
$ docker-machine ls
NAME       ACTIVE   DRIVER    STATE     URL                         SWARM
master01            hypriot   Running   tcp://192.168.0.114:2376   master01 (master)
node01              hypriot   Running   tcp://192.168.0.109:2376   master01
HypriotOS: pi@rpi202 in ~
$


# Nodes in the swarm:

HypriotOS: pi@rpi202 in ~
$ docker $(docker-machine config --swarm master01) info
Containers: 4
Images: 51
Role: primary
Strategy: spread
Filters: health, port, dependency, affinity, constraint
Nodes: 2
 master01: 192.168.0.114:2376
  └ Status: Healthy
  └ Containers: 3
  └ Reserved CPUs: 0 / 4
  └ Reserved Memory: 0 B / 972 MiB
  └ Labels: executiondriver=native-0.2, kernelversion=4.1.8-hypriotos-v7+, operatings
 node01: 192.168.0.109:2376
  └ Status: Healthy
  └ Containers: 1
  └ Reserved CPUs: 0 / 4
  └ Reserved Memory: 0 B / 972 MiB
  └ Labels: executiondriver=native-0.2, kernelversion=4.1.8-hypriotos-v7+, operatings
CPUs: 8
Total Memory: 1.899 GiB
Name: b7def5d9af98
HypriotOS: pi@rpi202 in ~
$
```

2. Establish the broker for Celery, with a Docker container running Redis.

In [ ]:

```
HypriotOS: pi@rpi202 in ~
$ docker run -d -p 6379:6379 --net=mynet --name=redis --volume=/data:/data hypriot/rp
a2abf9277b5e4818da89ffa282a706506ef288426486cc25b431208564bf6e0f


HypriotOS: pi@rpi202 in ~
$ docker ps
CONTAINER ID        IMAGE                       COMMAND                  CREATED
a2abf9277b5e        hypriot/rpi-redis           "/entrypoint.sh redis"   13 hours ago
f0ce33ca1152        hypriot/rpi-swarm           "/swarm join --advert"   6 days ago
b7def5d9af98        hypriot/rpi-swarm           "/swarm manage --tlsv"   6 days ago
ad594813f8f0        nimblestratus/rpi-consul    "/bin/start -server -"   6 days ago
HypriotOS: pi@rpi202 in ~
$
```

3. Copy celeryconfig.py、start_workers.sh、and the folder "word_count" to two hosts, under the folder of /data/celery_projects.

In [ ]:

```
HypriotOS: pi@rpi202 in /data/celery_projects
$ ll
total 20
drwxr-xr-x 3  999 root 4096 Jan 25 23:01 ./
drwxr-xr-x 3  999 root 4096 Jan 25 23:01 ../
-rw-r--r-- 1  999 root 1079 Jan 25 21:12 celeryconfig.py
-rwxr-xr-x 1  999 root  732 Jan 25 22:53 start_workers.sh*  <--- script to start up w
drwxr-xr-x 3 root root 4096 Jan 25 23:01 word_count/
HypriotOS: pi@rpi202 in /data/celery_projects
$


HypriotOS: pi@rpi201 in /data/celery_projects
$ ll
total 20
drwxr-xr-x 3 root root 4096 Jan 25 23:03 ./
drwxr-xr-x 3  999 root 4096 Jan 25 22:55 ../
-rw-r--r-- 1 root root 1079 Jan 25 21:12 celeryconfig.py
drwxr-xr-x 3 root root 4096 Jan 25 23:03 word_count/
HypriotOS: pi@rpi201 in /data/celery_projects
$
```

4. Establish Flower (http://docs.celeryproject.org/en/latest/userguide/monitoring.html#flower-real-time-celery-web-monitor) container for monitoring.

In [ ]:

```
HypriotOS: pi@rpi202 in ~
$ docker run -d -p 5555:5555 --net=mynet --name=flower --volume=/data/celery_projects
276f00591fd7042139ddf660730d223bcf19e9f8bd369f075de417140b6dfd4a
HypriotOS: pi@rpi202 in ~
$


HypriotOS: pi@rpi202 in ~
$ docker ps
CONTAINER ID        IMAGE                       COMMAND                  CREATED
276f00591fd7        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   31 seconds ag
```

Connect to container of Flower, view with a web browser, no tasks yet.

5. Deploy Celery worker containers through Swarm-Manager.

deploy worker containers with :

start_workers.sh

In [ ]:

```
# ./start_workers.sh

echo "Starting Celery cluster containers ____________________________________

eval $(docker-machine env --swarm master01)

PROJECT=$1  # project name
WORKER_START_ID=$2  # worker container index, the first one.
WORKER_LAST_ID=$3  # worker container index, the last one.
CONCURRENCY=$4  # number of subprocesses per worker.

for (( i=${WORKER_START_ID}; i<=${WORKER_LAST_ID}; i=i+1 ))
do
  docker run -d --name=${PROJECT}_celery${i} --hostname=${PROJECT}_celery${i} --net=m
done
```

Establish Celery worker containers for the first round.

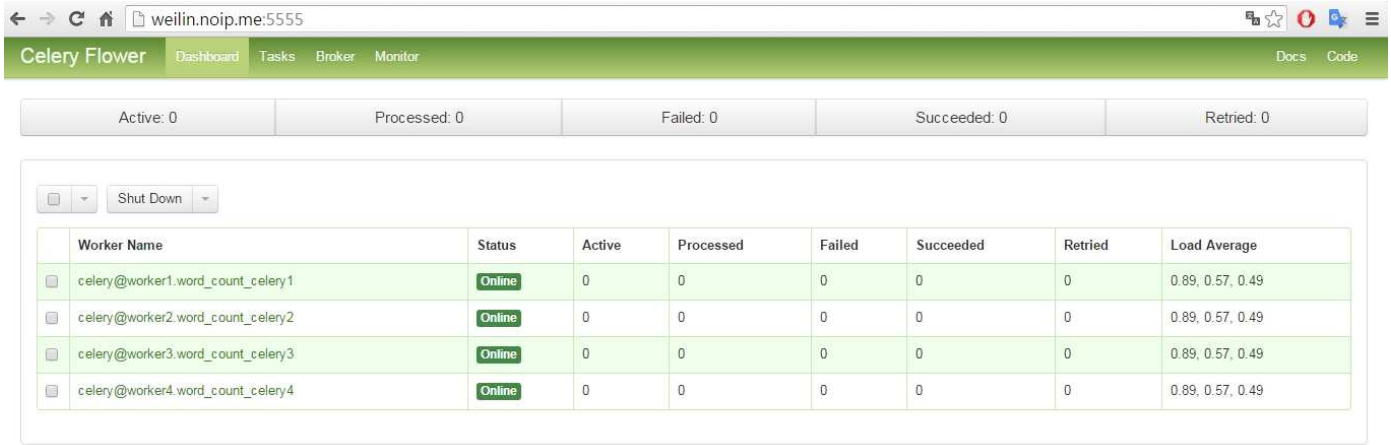
Four containers were established，each container has one Celery worker，each worker can have 5 subprocesses at most。

In [ ]:

```
# CLI parameters:
# $1 # project name
# $2 # worker container index, the first one.
# $3 # worker container index, the last one.
# $4 # number of subprocesses per worker.

HypriotOS: pi@rpi202 in /data/celery_projects
$ ./start_workers.sh word_count 1 4 5
Starting Celery cluster containers ____________________________________________
a22b08a0818b3246f90511ad21cb2a0ab37a4e72661bf559ade7e320db030505
77eabded27e4ea3aaa640480c088fa7b4b9818fc3e40fb66636cc9abe8a78e69
df05a7204f40470cfd8eee21a06be45f5a306ea32df0196f3d004beac5d2f82d
e67d39740ace5c2a5b9a05e6ca1adc73c5e5944e62302d02391d37f7ee6aa479



# Four containers were established，each container has one Celery worker，each worker

HypriotOS: pi@rpi202 in /data/celery_projects
$ docker ps
CONTAINER ID        IMAGE                       COMMAND                  CREATED
e67d39740ace        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   About a minut
df05a7204f40        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   About a minut
77eabded27e4        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   About a minut
a22b08a0818b        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   About a minut
276f00591fd7        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   37 minutes ag
a2abf9277b5e        hypriot/rpi-redis           "/entrypoint.sh redis"   13 hours ago
980161d10fc4        hypriot/rpi-swarm           "/swarm join --advert"   6 days ago
f0ce33ca1152        hypriot/rpi-swarm           "/swarm join --advert"   6 days ago
b7def5d9af98        hypriot/rpi-swarm           "/swarm manage --tlsv"   6 days ago
ad594813f8f0        nimblestratus/rpi-consul    "/bin/start -server -"   6 days ago
HypriotOS: pi@rpi202 in /data/celery_projects
$
```

Flower shows the status. There are four workers.

- celery@worker1.word_count_celery1
- celery@worker2.word_count_celery2
- celery@worker3.word_count_celery3
- celery@worker4.word_count_celery4



Expand Celery worker containers for the second round.

The Celery cluster can be scaled up dynamically, in response to the need of performance.
Adding 4 containers, each container has one Celery worker, each worker has 5 subprocesses at most.

In [ ]:

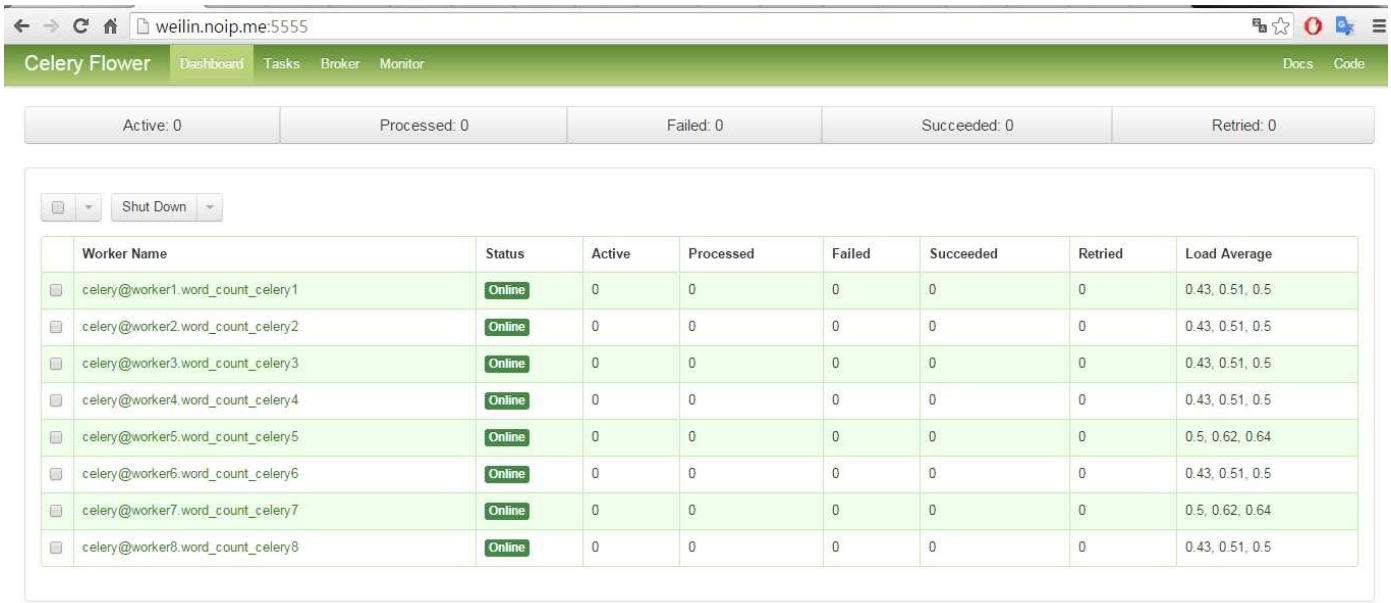
```
# CLI parameters:
# $1 # project name
# $2 # worker container index, the first one.
# $3 # worker container index, the last one.
# $4 # number of subprocesses per worker.

HypriotOS: pi@rpi202 in /data/celery_projects
$ ./start_workers.sh word_count 5 8 5
Starting Celery cluster containers ____________________________________________
a4de4967fd6211266cbad04fecfc357aa81789063cca3042388019adab2a6c71
7066ba6021de870f1332858c6f96673a159d7e5031a5682d3853fa6bd8fe2252
79792c823fbf2769e4983c525598c30ba3758c23697ef66a78a54163374d3233
25c02d07ed6f1217ee68dc486a6586262ca2e3ed01a2a8690eaa2a783ad07d73


HypriotOS: pi@rpi202 in /data/celery_projects
$ docker ps
CONTAINER ID        IMAGE                       COMMAND                  CREATED
25c02d07ed6f        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   18 seconds ag
79792c823fbf        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   39 seconds ag
7066ba6021de        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   About a minut
a4de4967fd62        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   About a minut
e67d39740ace        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   15 minutes ag
df05a7204f40        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   15 minutes ag
77eabded27e4        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   15 minutes ag
a22b08a0818b        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   16 minutes ag
276f00591fd7        wei1234c/celery_armv7       "/bin/sh -c 'cd /cele"   51 minutes ag
a2abf9277b5e        hypriot/rpi-redis           "/entrypoint.sh redis"   14 hours ago
980161d10fc4        hypriot/rpi-swarm           "/swarm join --advert"   6 days ago
f0ce33ca1152        hypriot/rpi-swarm           "/swarm join --advert"   6 days ago
b7def5d9af98        hypriot/rpi-swarm           "/swarm manage --tlsv"   6 days ago
ad594813f8f0        nimblestratus/rpi-consul    "/bin/start -server -"   6 days ago
HypriotOS: pi@rpi202 in /data/celery_projects
$
```

We can also view the list of Swarm nodes.

In [ ]:

```
HypriotOS: pi@rpi201 in /data/celery_projects
$ docker ps
CONTAINER ID        IMAGE                    COMMAND                  CREATED
25c02d07ed6f        wei1234c/celery_armv7    "/bin/sh -c 'cd /cele"   16 minutes ago
7066ba6021de        wei1234c/celery_armv7    "/bin/sh -c 'cd /cele"   17 minutes ago
e67d39740ace        wei1234c/celery_armv7    "/bin/sh -c 'cd /cele"   32 minutes ago
df05a7204f40        wei1234c/celery_armv7    "/bin/sh -c 'cd /cele"   32 minutes ago
77eabded27e4        wei1234c/celery_armv7    "/bin/sh -c 'cd /cele"   32 minutes ago
a22b08a0818b        wei1234c/celery_armv7    "/bin/sh -c 'cd /cele"   32 minutes ago
980161d10fc4        hypriot/rpi-swarm        "/swarm join --advert"   6 days ago
HypriotOS: pi@rpi201 in /data/celery_projects
$
```

Flower shows the workers. We have 8 of them now.



"Word Count" program re-writen in Python.

In [18]:

```
from word_count.tasks import *

# split text file into a list of words
def getWordsFromText(file = '.\\text\\test.txt'):
    with open(file) as f:
        lines = f.readlines()
    return ' '.join(lines).replace(',', '').replace('.', '').split()


def reduce(word_counts):
    wordCounts = {}

    for word_count in word_counts:
        if word_count is not None:
            wordCounts[word_count[0]] = wordCounts.get(word_count[0], 0) + word_count

    result = sorted(list(wordCounts.items()),
                    key = lambda x: (x[1], x[0]),
                    reverse = True)
    return result
```

In [19]:

```
# list of words
words = getWordsFromText()
words[:3]
```

Out[19]:

```
["Aesop's", 'Fables', 'Translated']
```

In [20]:

```
# how many words in the article.
len(words)
```

Out[20]:

```
2190
```

./word_count/tasks.py , which contains mapper function:

```
from celery import group
from word_count.celery import app

@app.task
def mapper(word):
    return (word, 1) if len(word) >= 5 else None    # discard words which ar
e too short.
```

Send messages to Celery workers, computing in parallel.

In [21]:

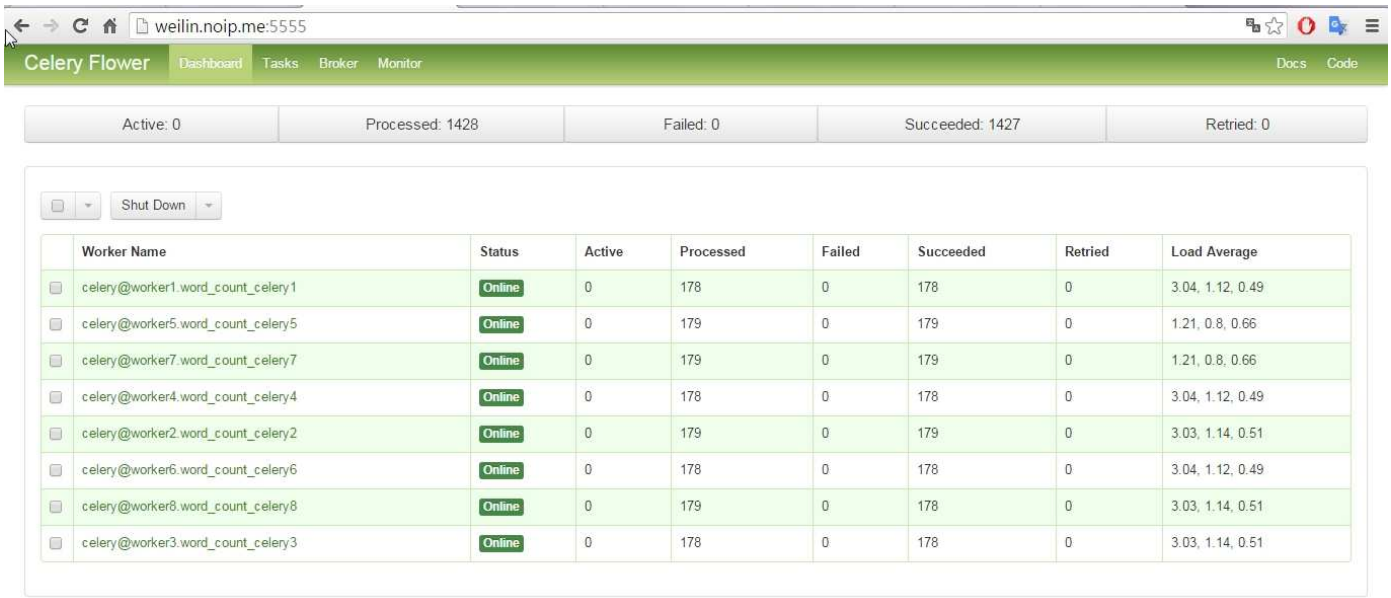
```
def count_celery(words):

    # send messages to Celery workers.
    asyncResults = [mapper.s(word).delay() for word in words]  # mapper function is a
    results = [asyncResult.get() for asyncResult in asyncResults if asyncResult.get()

    return reduce(results)

%time counts = count_celery(words)
counts[:5]
```

```
Wall time: 3min 23s
```

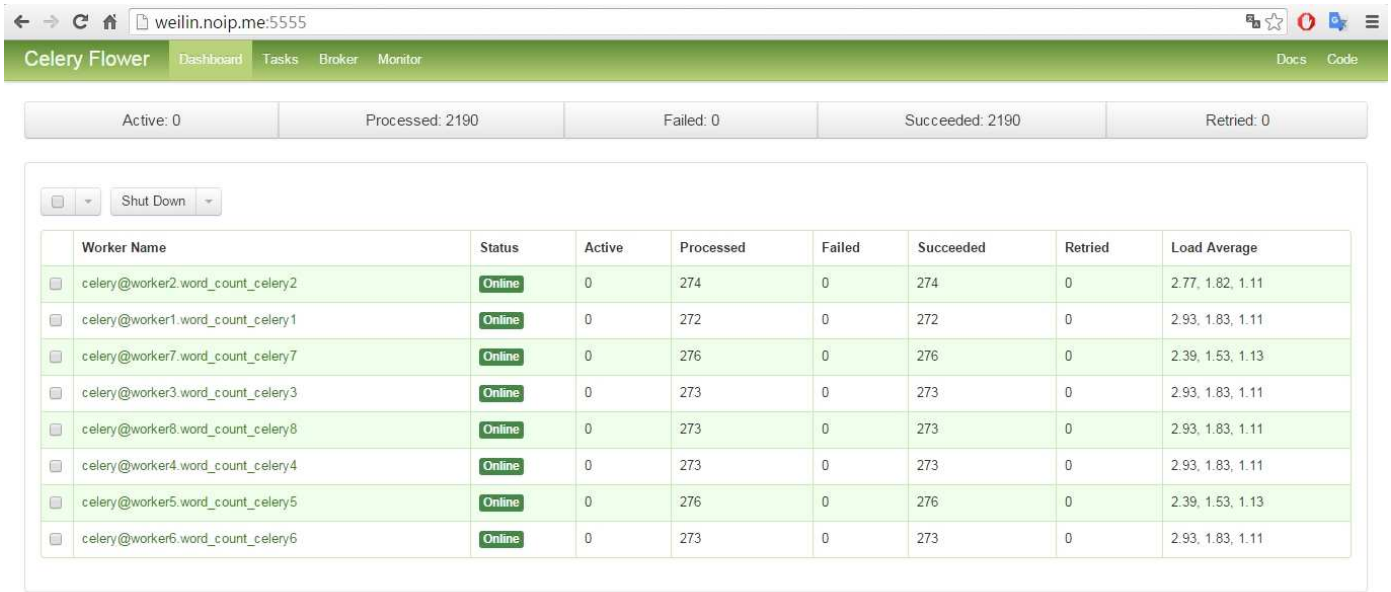
Out[21]:

```
[('would', 12), ('which', 8), ('their', 8), ('caught', 6), ('Farmer',
6)]
```

As messages sent，Flower shows the status.

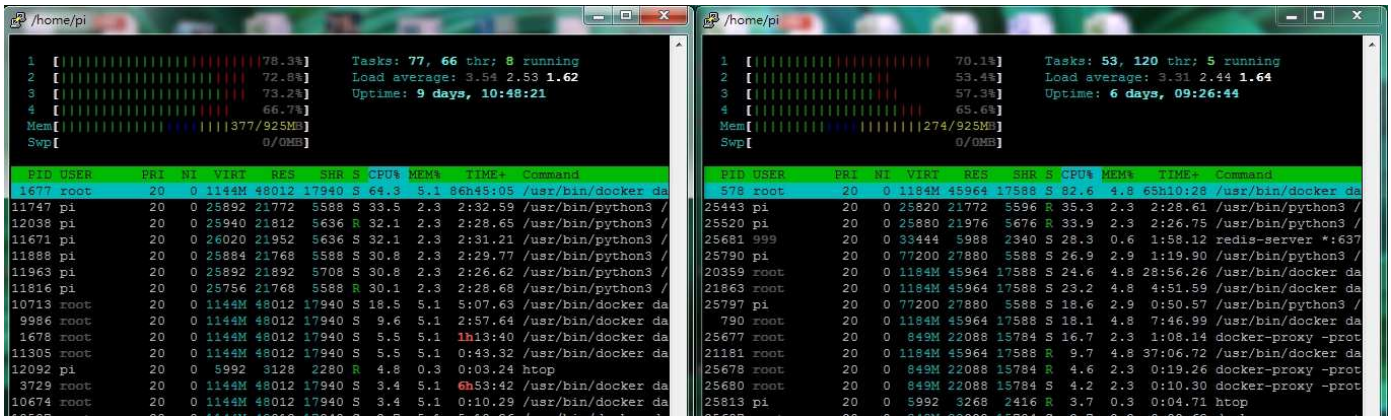


Finished.



Worker Name	Status	Active	Processed	Failed	Succeeded	Retried	Load Average
celery@worker2.word_count_celery2	Online	0	274	0	274	0	2.77, 1.82, 1.11
celery@worker1.word_count_celery1	Online	0	272	0	272	0	2.93, 1.83, 1.11
celery@worker7.word_count_celery7	Online	0	276	0	276	0	2.39, 1.53, 1.13
celery@worker3.word_count_celery3	Online	0	273	0	273	0	2.93, 1.83, 1.11
celery@worker8.word_count_celery8	Online	0	273	0	273	0	2.93, 1.83, 1.11
celery@worker4.word_count_celery4	Online	0	273	0	273	0	2.93, 1.83, 1.11
celery@worker5.word_count_celery5	Online	0	276	0	276	0	2.39, 1.53, 1.13
celery@worker6.word_count_celery6	Online	0	273	0	273	0	2.93, 1.83, 1.11

Showing CPUs' load during the processing.



Summary:

With respect to Hadoop ecosystem, we can actually build a parallel-computing cluster in minutes with Celery and Docker-Swarm. This practice could be used as a handy tool in some circumstances.