

Performance of the program

1. Submit a spark job for only one vm (hadoop1)

I started the master VM using command- `/opt/spark/sbin/start-master.sh`

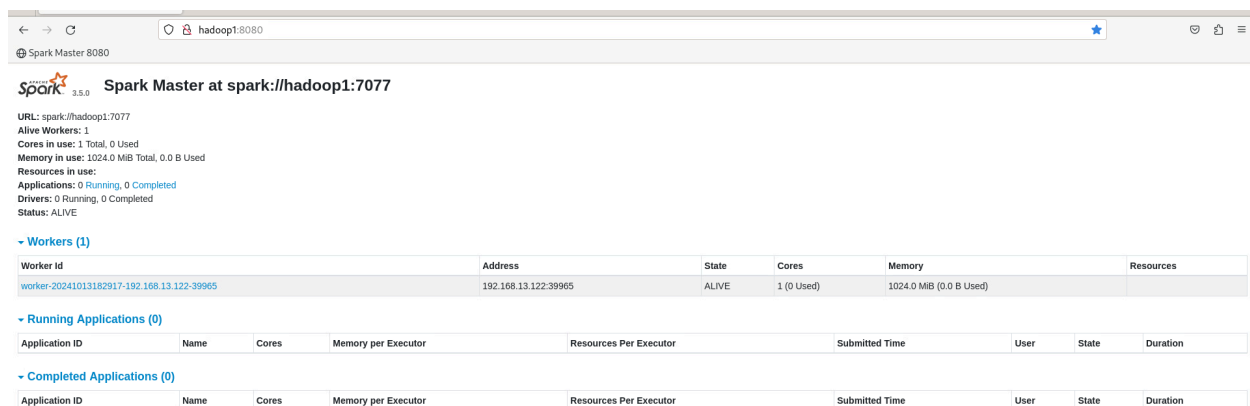
Then I started the worker VM (only hadoop 1) using this command

`/opt/spark/sbin/start-worker.sh spark://hadoop1:7077`

```
[root@hadoop1 bin]# /opt/spark/sbin/start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spark
-sat3812-org.apache.spark.deploy.master.Master-1-hadoop1.out
```

```
[root@hadoop1 spark]# sbin/start-worker.sh spark://hadoop1:7077
starting org.apache.spark.deploy.worker.Worker, logging to /opt/spark/logs/spark
-sat3812-org.apache.spark.deploy.worker.Worker-1-hadoop1.out
[root@hadoop1 spark]# jps
5058 SparkSubmit
6004 Jps
5944 Worker
5835 Master
[root@hadoop1 spark]#
```

Here, I am only considering one worker that is (hadoop1 VM).



The screenshot shows the Spark Master web interface at `spark://hadoop1:7077`. The interface displays the following information:


- URL:** `spark://hadoop1:7077`
- Alive Workers:** 1
- Cores in use:** 1 Total, 0 Used
- Memory in use:** 1024.0 MiB Total, 0.0 B Used
- Resources in use:**
- Applications:** 0 Running, 0 Completed
- Drivers:** 0 Running, 0 Completed
- Status:** ALIVE

Below this information, there are three expandable sections:

- Workers (1):** A table showing one worker with ID `worker-20241013182917-192.168.13.122-39965`, address `192.168.13.122:39965`, state `ALIVE`, 1 core used, and 1024.0 MiB memory used.
- Running Applications (0):** A table with columns: Application ID, Name, Cores, Memory per Executor, Resources Per Executor, Submitted Time, User, State, and Duration.
- Completed Applications (0):** A table with columns: Application ID, Name, Cores, Memory per Executor, Resources Per Executor, Submitted Time, User, State, and Duration.

Now I am running my python code using only on this vm(hadoop1) using command

`opt/spark/bin/spark-submit --master spark://hadoop1:7077 /opt/preprocessing.py`


Spark Master at spark://hadoop1:7077

URL: spark://hadoop1:7077
 Alive Workers: 1
 Cores in use: 1 Total, 0 Used
 Memory in use: 1024.0 MiB Total, 0.0 B Used
 Resources in use:
 Applications: 0 Running, 2 Completed
 Drivers: 0 Running, 0 Completed
 Status: ALIVE

▼ Workers (1)

Worker Id	Address	State	Cores	Memory	Resources
worker-20241013184718-192.168.13.122-39759	192.168.13.122:39759	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

▼ Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

▼ Completed Applications (2)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20241013185149-0001	StudentPerformanceFactors	1	1024.0 MiB		2024/10/13 18:51:49	root	FINISHED	2.4 min
app-20241013184829-0000	StudentPerformanceFactors	1	1024.0 MiB		2024/10/13 18:48:29	root	FINISHED	2.3 min

I observed that the duration for completing the action using only 1 Vm was approximately 2.4 minutes. When I ran the command again to check for consistency, the time remained almost the same at 2.3 minutes. This indicates that running the Python code on a single VM (using Hadoop 1 only) consistently takes about 2.4 minutes.

Now Let's check how this duration will be changed if we use 2 workers (VMs).

2. Submit a spark job for two Vms (hadoop1 and hadoop2)

I started all (master and worker) Vms using command - /opt/spark/sbin/start-all.sh

```

[root@hadoop1 ~]# /opt/spark/sbin/start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spark-
-sat3812-org.apache.spark.deploy.master.Master-1-hadoop1.out
192.168.13.123: starting org.apache.spark.deploy.worker.Worker, logging to /opt/
spark/logs/spark-root-org.apache.spark.deploy.worker.Worker-1-hadoop2.out
192.168.13.122: starting org.apache.spark.deploy.worker.Worker, logging to /opt/
spark/logs/spark-root-org.apache.spark.deploy.worker.Worker-1-hadoop1.out
[root@hadoop1 ~]# jps
4096 Master
4325 Jps
4234 Worker
[root@hadoop1 ~]#
  
```

I also checked for my second vm (hadoop2) using jps, and it shows that it started as a worker.

```
[root@hadoop2 sat3812]# jps
2932 Worker
3006 Jps
[root@hadoop2 sat3812]#
```

Here, I am considering 2 Vms (2 workers - hadoop1 and hadoop2)

Spark Master at spark://hadoop1:8080

Spark Master at spark://hadoop1:7077

URL: spark://hadoop1:7077
Alive Workers: 2
Cores in use: 2 Total, 0 Used
Memory in use: 2.0 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 1 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20241013174017-192.168.13.123-41531	192.168.13.123:41531	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20241013174020-192.168.13.122-45811	192.168.13.122:45811	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

The duration for completing a specific action using two VMs was about 27 seconds. This is approximately five times faster than the time taken when using only one VM, which was around 2.4 minutes. Running the Python code using both VMs (Hadoop 1 and Hadoop 2) demonstrates a significant improvement in processing speed.

Spark Master at spark://hadoop1:8080

Spark Master at spark://hadoop1:7077

URL: spark://hadoop1:7077
Alive Workers: 2
Cores in use: 2 Total, 0 Used
Memory in use: 2.0 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 1 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20241013175159-192.168.13.123-34981	192.168.13.123:34981	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20241013175203-192.168.13.122-41309	192.168.13.122:41309	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20241013175502-0000	StudentPerformanceFactors	2	1024.0 MiB		2024/10/13 17:55:02	root	FINISHED	27 s