Tao Chen
Project 2

Step 0 setup:

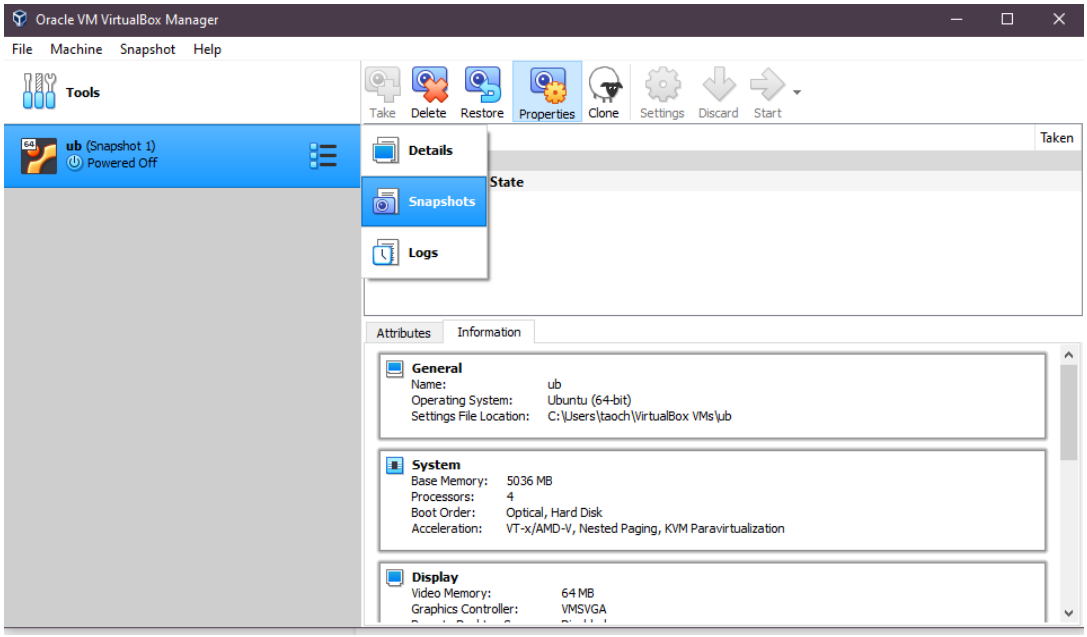
At first I wanted to use AWS EMR however I wanted to test the code because it will most likely have bugs and errors which takes time to fix and that will add to the bill when there is a lot of network usage.
When I tried to run spark on my machine I found a way to run a master and slave which got me into downloading a Linux Virtual Machine and running it from that.
The VM I decided to use is VirtualBox: https://www.virtualbox.org/wiki/Downloads
Creating a linux VM Ubuntu with 2 gb memory

Create a virtual hard disk now → VDI → Dynamically allocated → 10 GB
In the setting need to have the Ubuntu desktop
Setting -> storage - Controller click and import the .iso downloaded from https://ubuntu.com/download/desktop , choose/create a virtual optical disk
Setting → System - uncheck Floppy Disk → Processors 2 core
Setting → Display gave 16 video Memory
When first launch require a username and password the rest is like installing a new OS

After Installing which took a long time (Using Hard Drive)

On the top in device → Insert Guest addition cd image

I decided to take an image of the state so that I can reload it, if I install something that breaks or if I install something different and don't want to reverse it where the reload from the image is faster. Take an image at certain checkpoints/ between tasks.



This is a new installation so it does not have anything, need to be installed

Update Ubuntu > sudo apt update && sudo apt upgrade -y

Need java 8 > sudo apt install openjdk-8-jre-headless

Install SSH > sudo apt install openssh-server openssh-client -y

> sudo apt install ssh

Can > ssh localhostUse scp to transfer to the other servers

Needed for hdfs

> cd /home/tao/Desktop/Project2

Download Hadoop

> wget https://mirror.olnevhost.net/pub/apache/hadoop/common/hadoop-3.1.4/hadoop-3.1.4.tar.gz

Unpackage > tar xvfz hadoop-3.1.4.tar.gz

Download Spark

> wget https://apache.claz.org/spark/spark-3.1.1/spark-3.1.1-bin-hadoop2.7.tgz

Unpackage > tar xvfz spark-3.1.1-bin-hadoop2.7.tgz

Environment

> sudo nano /etc/environment

JAVA_HOME=/lib/jvm/java-8-openjdk-amd64

HADOOP_HOME=/home/tao/Desktop/Project2/hadoop-3.1.4

HADOOP_CONF_DIR=/home/tao/Desktop/Project2/hadoop-3.1.4/etc/hadoop

SPARK_HOME=/home/tao/Desktop/Project2/spark-3.1.1-bin-hadoop2.7

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/tao/Desktop/Project2/hadoop-3.1.4/bin:/home/tao/Desktop/Project2/hadoop-3.1.4/sbin:/lib/jvm/java-8-openjdk-amd64/bin:/home/tao/Desktop/Project2/spark-3.1.1-bin-hadoop2.7/sbin:/home/tao/Desktop/Project2/spark-3.1.1-bin-hadoop2.7/bin”

Step 1 Creating master and workers:

If there are multiple machines then you need to set up the ssh key so the master node can access the worker machine. Then add the ip address of the worker machine to the conf/slave file in spark on the master machine.

The Step I am doing is to set up the cluster on a single Linux machine otherwise you need to set up everything on all machines, installing the java, spark.

Creating master node

> cd $SPARK_HOME/sbin

> ./start-master.sh

```
tao@tao-VirtualBox:/usr/local/spark-3.1.1-bin-hadoop2.7/sbin$ start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark-3.1
.1-bin-hadoop2.7/logs/spark-tao-org.apache.spark.deploy.master.Master-1-tao-Vir
tualBox.out
```

View the log to see the login details

> nano /usr/local/spark-3.1.1-bin-hadoop2.7/logs/spark-tao-org.apache.spark.deploy.master.Master-1-tao-VirtualBox.out

At the end of first line you can see the host and then the port

In the web browser interface

>tao-virtualbox:8080

The url is where you can access the spark(master): spark://tao-VirtualBox:7077

Creating Worker node

Need to set up the spark environment to allow more workers and to prevent all resources used.

>cd /home/tao/Desktop/Project2/spark-3.1.1-bin-hadoop2.7/conf

> nano spark-env.sh

SPARK_WORKER_CORES=1

SPARK_WORKER_INSTANCES=2

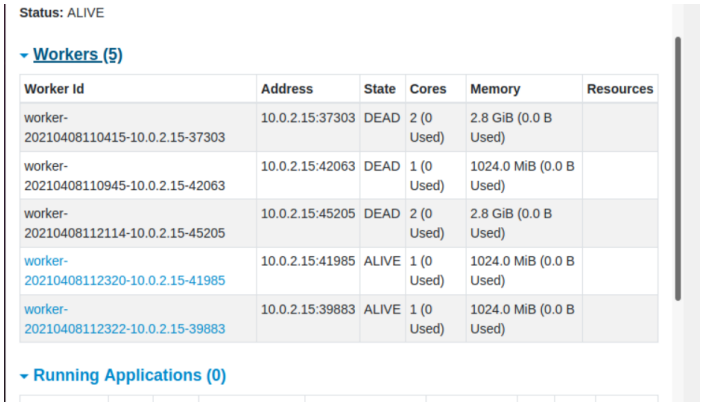
SPARK_WORKER_MEMORY=1G

#This is how many core each worker get and how many worker created

> cd /home/tao/Desktop/Project2/spark-3.1.1-bin-hadoop2.7/sbin

> ./start-worker.sh spark://tao-VirtualBox:7077

Need to add the argument of where the worker node is append to



Status: ALIVE

Workers (5)

Worker Id	Address	State	Cores	Memory	Resources
worker-20210408110415-10.0.2.15-37303	10.0.2.15:37303	DEAD	2 (0 Used)	2.8 GiB (0.0 B Used)	
worker-20210408110945-10.0.2.15-42063	10.0.2.15:42063	DEAD	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20210408112114-10.0.2.15-45205	10.0.2.15:45205	DEAD	2 (0 Used)	2.8 GiB (0.0 B Used)	
worker-20210408112320-10.0.2.15-41985	10.0.2.15:41985	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20210408112322-10.0.2.15-39883	10.0.2.15:39883	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

Running Applications (0)

Step 2 deploy HDFS:

This requires Hadoop, everything is already set up the environment and the unpackage in step 0. We need to edit some files

Setup

> cd /home/tao/Desktop/Project2/hadoop-3.1.4/etc/hadoop

> nano core-site.xml

Add property between configuration

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://tao-VirtualBox:9000</value>
    </property>
</configuration>
```

This is the port that hdfs reside in. the hdfs://have to match /etc/hosts

> nano hdfs-site.xml

```
  GNU nano 4.8                    hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

hdfs replicates data so that if one node crashes there is a backup. The replication tells the hdfs how many times to replicate then put it into different nodes. I only need 1, it also saves space.

There are two users: the localhost and the tao-virtualbox so to edit file need to create ssh keys else it will encounter permissions issues.

Create access

> ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

```
The key fingerprint is:
SHA256:fTwlcJQfFklrFBxj45ze/SzHdfr2xcxtBrlU56lLoWk tao@tao-VirtualBox
The key's randomart image is:
+---[RSA 3072]----+
|          ..o+X= |
|           o.=== |
|            .oB.o|
|         . . =.=+|
|        S . +.=.*|
|           .oooX=|
|           E o+ &|
|          . . .*o|
|             . .+|
+----[SHA256]-----+
```

> cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

>chmod 0600 ~/.ssh/authorized_keys

Creating hdfs

> cd /home/tao/Desktop/Project2/hadoop-3.1.4

> bin/hdfs namenode -format

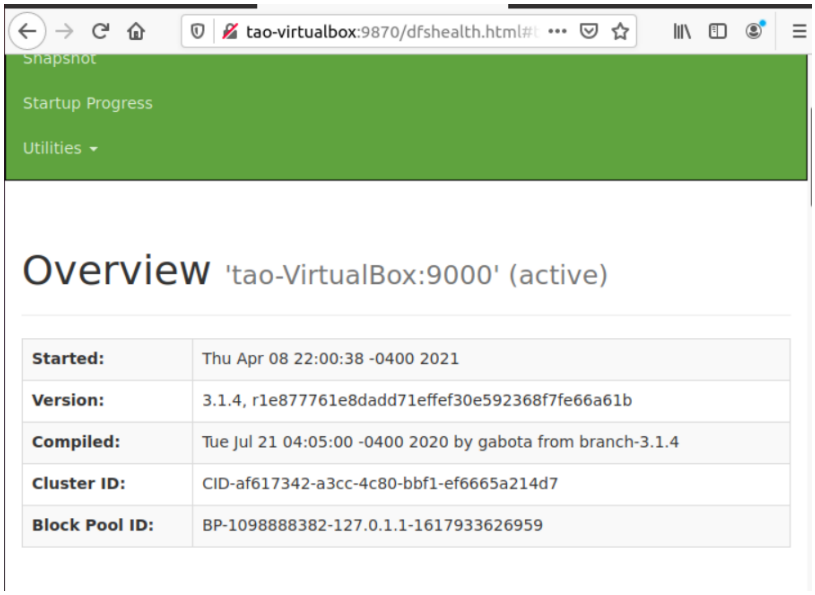
```
2021-04-08 21:31:18,826 INFO namenode.FSNamesystem: Retry cache will use 0.03 o
f total heap and retry cache entry expiry time is 600000 millis
2021-04-08 21:31:18,827 INFO util.GSet: Computing capacity for map NameNodeRetr
yCache
2021-04-08 21:31:18,827 INFO util.GSet: VM type       = 64-bit
2021-04-08 21:31:18,827 INFO util.GSet: 0.029999999329447746% max memory 875 MB
 = 268.8 KB
2021-04-08 21:31:18,827 INFO util.GSet: capacity      = 2^15 = 32768 entries
2021-04-08 21:31:18,980 INFO namenode.FSImage: Allocated new BlockPoolId: BP-19
34590963-127.0.1.1-1617931878975
2021-04-08 21:31:19,059 INFO common.Storage: Storage directory /tmp/hadoop-tao/
dfs/name has been successfully formatted.
2021-04-08 21:31:19,094 INFO namenode.FSImageFormatProtobuf: Saving image file
/tmp/hadoop-tao/dfs/name/current/fsimage.ckpt_0000000000000000000 using no comp
ression
2021-04-08 21:31:19,167 INFO namenode.FSImageFormatProtobuf: Image file /tmp/ha
doop-tao/dfs/name/current/fsimage.ckpt_0000000000000000000 of size 390 bytes sa
ved in 0 seconds .
2021-04-08 21:31:19,177 INFO namenode.NNStorageRetentionManager: Going to retai
n 1 images with txid >= 0
2021-04-08 21:31:19,190 INFO namenode.FSImage: FSImageSaver clean checkpoint: t
xid = 0 when meet shutdown.
2021-04-08 21:31:19,190 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at tao-VirtualBox/127.0.1.1
************************************************************/
tao@tao-VirtualBox:~/Desktop/Project2/hadoop-3.1.4/bin$
```

Starting the hdfs

>sbin/start-dfs.sh

```
**********************************************************/
tao@tao-VirtualBox:~/Desktop/Project2/hadoop-3.1.4/bin$ cd ..
tao@tao-VirtualBox:~/Desktop/Project2/hadoop-3.1.4$ sbin/start-dfs.sh
Starting namenodes on [tao-VirtualBox]
Starting datanodes
Starting secondary namenodes [tao-VirtualBox]
```

http://tao-virtualbox:9870 to view on the web



Task 3 Download and save to HDFS:

First need the file on the linux machine

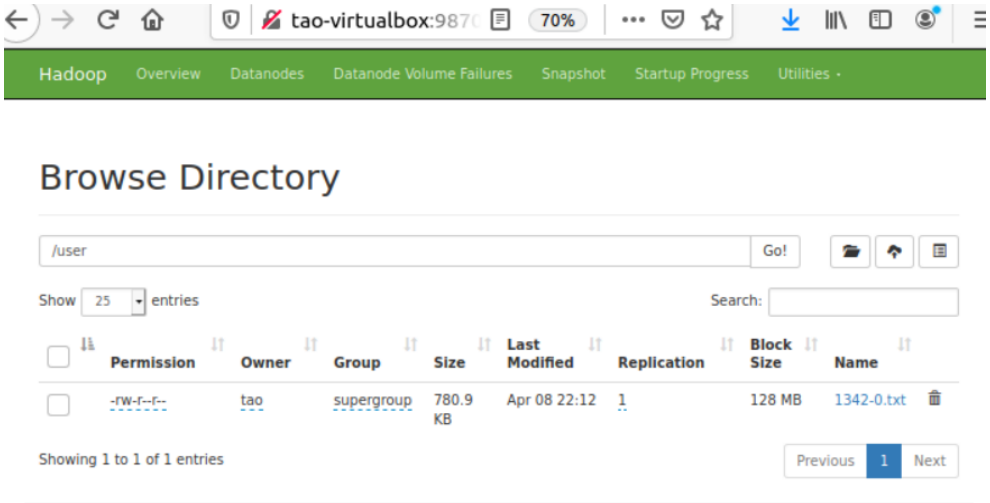
https://www.gutenberg.org/files/1342/1342-0.txt

> ctrl + s

Make a folder easier to see

>bin/hdfs dfs -mkdir /user

upload>bin/hdfs dfs -put /home/tao/Desktop/1342-0.txt /user



Testing for accessing the hdfs:

```
>cd /home/tao/Desktop/Project2/spark-3.1.1-bin-hadoop2.7
>bin/spark-shell
download>val text = sc.textFile("hdfs://tao-virtualbox:9000/user/1342-0.txt")
upload>text.saveAsTextFile("hdfs://tao-virtualbox:9000/user/count_output.txt")
```

Task 4 Deploy Spark Service:

Follow Step 0 to install Spark.

I have created a Spark cluster with master nodes and worker nodes in Step 1. To submit work to the cluster need the master url which is not the url to the web interface.

Task 5 HDFS as input run a wordcount program:

The code to submit the job starts and end with ====

```
==========================================================
import sys
from pyspark import SparkConf
from pyspark import SparkContext

sc = SparkContext("spark://tao-VirtualBox:7077","Word Count")
text_file = sc.textFile("hdfs://tao-virtualbox:9000/user/1342-0.txt")
line = text_file.flatMap(lambda line: line.lower().split(" "))
maps = line.map(lambda word: (word, 1))
reducer = maps.reduceByKey(lambda a, b: a + b)
most_use =reducer.takeOrdered(20,lambda x:-x[1])
list_to_rdd = sc.parallelize(most_use)
list_to_rdd.coalesce(1).saveAsTextFile("hdfs://tao-virtualbox:9000/user/coutputv6.txt")
==========================================================
```

Explanation

- Need to set the SparkContext on where to run it, and name the execution
  - If running locally it be local[*]. Running on spark url automatically split the work base on the number of worker node and local[*] split the work based on resources. * can be changed.
- Everything up to reducer is standard word count program
- most_use is a list of 20 objects ordered in descending order from reducer. (action)
  - It collects the parts from the worker node into one, reduce it, and takeOrdered
  - First arg is take(amount) and second argument is how it is ordered.
  - The passing in lambda is element of the map where x[1] represent the value and -x[1] represent reverse order by value
- Spark code have to run from transformation than action and cannot be action to action
  - Since takeOrdered is an action which creates a list and saveAsTextFile is also an action it will produce an error, so I have to transform most_use to an rdd.

When saveAsTextFile is called it will have two parts 1 per worker node so .coalesce(1) have them all combine into 1 part.

> cd /Desktop/Project2/spark-3.1.1-bin-hadoop2.7
> bin/spark-submit /home/tao/Desktop/wordc.py --master spark://tao-VirtualBox:7077

```
2021-04-11 11:58:48,081 INFO spark.SparkContext: Invoking stop() from shutdown
hook
2021-04-11 11:58:48,086 INFO server.AbstractConnector: Stopped Spark@492d70fa{H
TTP/1.1, (http/1.1)}{0.0.0.0:4040}
2021-04-11 11:58:48,088 INFO ui.SparkUI: Stopped Spark web UI at http://10.0.2.
15:4040
2021-04-11 11:58:48,090 INFO cluster.StandaloneSchedulerBackend: Shutting down
all executors
2021-04-11 11:58:48,090 INFO cluster.CoarseGrainedSchedulerBackend$DriverEndpoi
nt: Asking each executor to shut down
2021-04-11 11:58:48,124 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTra
ckerMasterEndpoint stopped!
2021-04-11 11:58:48,156 INFO memory.MemoryStore: MemoryStore cleared
2021-04-11 11:58:48,156 INFO storage.BlockManager: BlockManager stopped
2021-04-11 11:58:48,162 INFO storage.BlockManagerMaster: BlockManagerMaster sto
pped
2021-04-11 11:58:48,168 INFO scheduler.OutputCommitCoordinator$OutputCommitCoor
dinatorEndpoint: OutputCommitCoordinator stopped!
2021-04-11 11:58:48,188 INFO spark.SparkContext: Successfully stopped SparkCont
ext
2021-04-11 11:58:48,192 INFO util.ShutdownHookManager: Shutdown hook called
2021-04-11 11:58:48,192 INFO util.ShutdownHookManager: Deleting directory /tmp/
spark-907992cd-c720-488a-b5af-80a6ee6b9c1b
2021-04-11 11:58:48,193 INFO util.ShutdownHookManager: Deleting directory /tmp/
spark-907992cd-c720-488a-b5af-80a6ee6b9c1b/pyspark-e17b54e8-3364-4951-8022-461c
5202f198
2021-04-11 11:58:48,194 INFO util.ShutdownHookManager: Deleting directory /tmp/
spark-691bd746-bc4c-420b-b6e2-77aa73a5b77f
tao@tao-VirtualBox:~/Desktop/Project2/spark-3.1.1-bin-hadoop2.7$
```

- If you scroll up a bit and there is a line said traceback means there is an error in code.
- One problem with saving the file to hdfs was the quotes. It has to be the straight double quote and not the opening double quote.

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	tao	supergroup	780.9 KB	Apr 11 10:12	1	128 MB	1342-0.txt
drwxr-xr-x	tao	supergroup	0 B	Apr 11 10:19	0	0 B	count_output.txt
drwxr-xr-x	tao	supergroup	0 B	Apr 11 11:11	0	0 B	coutput.txt
drwxr-xr-x	tao	supergroup	0 B	Apr 11 11:16	0	0 B	coutputv2.txt
drwxr-xr-x	tao	supergroup	0 B	Apr 11 11:40	0	0 B	coutputv3.txt
drwxr-xr-x	tao	supergroup	0 B	Apr 11 11:45	0	0 B	coutputv4.txt
drwxr-xr-x	tao	supergroup	0 B	Apr 11 11:52	0	0 B	coutputv5.txt
drwxr-xr-x	tao	supergroup	0 B	Apr 11 11:58	0	0 B	coutputv6.txt

Showing 1 to 8 of 8 entries Previous 1 Next

```
('', 73700)
('the', 4493)
('to', 4171)
('of', 3686)
('and', 3397)
('a', 1981)
('her', 1939)
('in', 1894)
('was', 1798)
('i', 1725)
('she', 1607)
('that', 1442)
('not', 1382)
('he', 1249)
('his', 1239)
('be', 1213)
('as', 1171)
('it', 1152)
('had', 1149)
('you', 1123)
```

Spark Master at spark://tao-VirtualBox:7077

3.1.1

URL: spark://tao-VirtualBox: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, 24 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20210411104444-10.0.2.15-35309	10.0.2.15:35309	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20210411104447-10.0.2.15-43345	10.0.2.15:43345	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 (24)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20210411115842-0023	Word Count	2	1024.0 MiB		2021/04/11 11:58:42	tao	FINISHED	5 s
app-20210411115803-0022	Word Count	2	1024.0 MiB		2021/04/11 11:58:03	tao	FINISHED	13 s
app-20210411115229-0021	Word Count	2	1024.0 MiB		2021/04/11 11:52:29	tao	FINISHED	7 s
app-20210411115141-0020	Word Count	2	1024.0 MiB		2021/04/11 11:51:41	tao	FINISHED	3 s
app-20210411114455-0019	Word Count	2	1024.0 MiB		2021/04/11 11:44:55	tao	FINISHED	14 s

Step 6 Estimate pi With Monte Carlo:

The code to submit the job starts and end with ====

==========

```
import sys
from random import random
from operator import add
from pyspark import SparkConf
from pyspark import SparkContext

sc = SparkContext("spark://tao-VirtualBox:7077","esti_pi")

n = 10000000
def sample_points(p):
    x = random() * 2 - 1
    y = random() * 2 - 1
    return 1 if x*x + y*y <= 1 else 0
count = sc.parallelize(range(1, n + 1)).map(sample_points).reduce(add)
string = ("Pi is %f" % (4.0 * count / n))
rdd = sc.parallelize(string.split(" "))
rdd.coalesce(1).saveAsTextFile("hdfs://tao-virtualbox:9000/user/piv4.txt")
```

==========

Explanation:

random() return a number 0 to 1. To get the negative we have to subtract 1 however That will produce -1 to 0. So random()*2 will return 0 to 2 then subtract 1 will Get -1 to 1

At the count variable parallelize a range of number so represent how many task to run In this case running n task (10000000) .map is in each of the task run the Function. . reduce(add) sums all the return value from the function.

In the string %f represent that f will be replaced by a value.

Since it is a string not an rdd have to convert to an rdd to save to a file.

```
> cd /Desktop/Project2/spark-3.1.1-bin-hadoop2.7
> bin/spark-submit /home/tao/Desktop/esti_pi.py --master spark://tao-VirtualBox:7077
```

```
2021-04-11 13:33:15,866 INFO cluster.StandaloneSchedulerBackend: Shutting down
all executors
2021-04-11 13:33:15,867 INFO cluster.CoarseGrainedSchedulerBackend$DriverEndpoi
nt: Asking each executor to shut down
2021-04-11 13:33:15,909 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTra
ckerMasterEndpoint stopped!
2021-04-11 13:33:16,080 INFO memory.MemoryStore: MemoryStore cleared
2021-04-11 13:33:16,080 INFO storage.BlockManager: BlockManager stopped
2021-04-11 13:33:16,083 INFO storage.BlockManagerMaster: BlockManagerMaster sto
pped
2021-04-11 13:33:16,085 INFO scheduler.OutputCommitCoordinator$OutputCommitCoor
dinatorEndpoint: OutputCommitCoordinator stopped!
2021-04-11 13:33:16,097 INFO spark.SparkContext: Successfully stopped SparkCont
ext
2021-04-11 13:33:16,097 INFO util.ShutdownHookManager: Shutdown hook called
2021-04-11 13:33:16,098 INFO util.ShutdownHookManager: Deleting directory /tmp/
spark-70533b91-eeed-4337-bf7a-0a90bbbd2a68
2021-04-11 13:33:16,113 INFO util.ShutdownHookManager: Deleting directory /tmp/
spark-467b48da-5cd2-40ea-95ec-483b697005ac
2021-04-11 13:33:16,119 INFO util.ShutdownHookManager: Deleting directory /tmp/
spark-70533b91-eeed-4337-bf7a-0a90bbbd2a68/pyspark-e519aa26-9b5d-44e0-92b1-00db
7ad09f8e
tao@tao-VirtualBox:~/Desktop/Project2/spark-3.1.1-bin-hadoop2.7$
```

```
1 Pi
2 is
3 3.142093
```

Completed Applications (42)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20210411133308-0041	esti_pi	2	1024.0 MiB		2021/04/11 13:33:08	tao	FINISHED	7 s
app-20210411133244-0040	esti_pi	2	1024.0 MiB		2021/04/11 13:32:44	tao	FINISHED	9 s
app-20210411133112-0039	esti_pi	2	1024.0 MiB		2021/04/11 13:31:12	tao	FINISHED	10 s

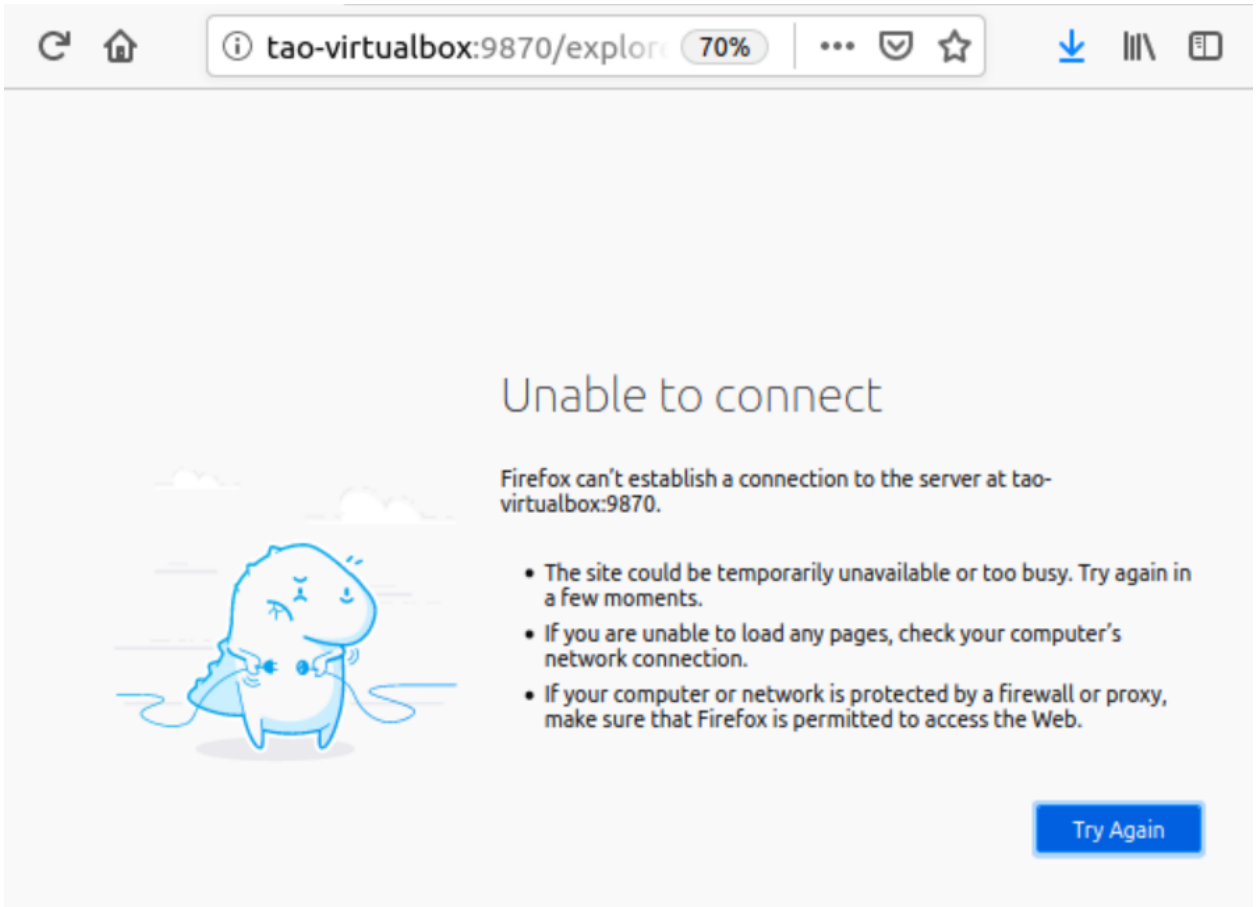
Step 7 Closing hdfs and spark cluster:

To Stop the hdfs:

> cd /home/tao/Desktop/Project2/hadoop-3.1.4

> sbin/stop-dfs.sh

```
tao@tao-VirtualBox:~/Desktop/Project2/hadoop-3.1.4$ sbin/stop-dfs.sh
Stopping namenodes on [tao-VirtualBox]
Stopping datanodes
Stopping secondary namenodes [tao-VirtualBox]
```



To stop the spark cluster:

> cd /Desktop/Project2/spark-3.1.1-bin-hadoop2.7

> sbin/stop-all.sh

```
tao@tao-VirtualBox:~/Desktop/Project2/spark-3.1.1-bin-hadoop2.7$ sbin/stop-all.
sh
localhost: stopping org.apache.spark.deploy.worker.Worker
localhost: stopping org.apache.spark.deploy.worker.Worker
stopping org.apache.spark.deploy.master.Master
```