安裝 Scala、Spark

下載安裝 Scala

```
wget http://www.scala-lang.org/files/archive/scala-2.11.6.tgz
tar xvf scala-2.11.6.tgz
sudo mv scala-2.11.6 /usr/local/scala
```

Scala 使用者環境變數設定

```
sudo gedit ~/.bashrc
```

輸入下列內容

```
export SCALA_HOME=/usr/local/scala
export PATH=$PATH:$SCALA_HOME/bin
```

使~/.bashrc 修改生效

```
source ~/.bashrc
```

下載安裝 Spark

```
wget https://archive.apache.org/dist/spark/spark-2.0.0/spark-2.0.0-bin-hadoop2.6.tgz
tar zxf spark-2.0.0-bin-hadoop2.6.tgz
sudo mv spark-2.0.0-bin-hadoop2.6 /usr/local/spark/
sudo chown hduser:hduser -R /usr/local/spark
```

Spark 使用者環境變數設定

```
sudo gedit ~/.bashrc
```

輸入下列內容

```
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SPARK_HOME/bin
```

使~/.bashrc 修改生效

```
source ~/.bashrc
```

啟動 python spark 互動介面

```
pyspark
```

離開 python spark

```
exit()
```

設定 pyspark 顯示訊息

```
cd /usr/local/spark/conf
cp log4j.properties.template log4j.properties
```

修改 log4j.properties

```
sudo gedit log4j.properties
```

將 log4j.rootCategory=INFO, console 的 INFO 改為 WARN

複製 LICENSE.txt

先建立工作目錄

```
mkdir -p ~/wordcount/input
```

複製檔案

```
cp /usr/local/hadoop/LICENSE.txt ~/wordcount/input
ll ~/wordcount/input
```

進入 master 虛擬機器，啟動 Hadoop Multi-Node Cluster

```
start-all.sh
```

上傳測試檔案至 HDFS 目錄

```
hadoop fs -mkdir -p /user/hduser/wordcount/input
cd ~/wordcount/input
hadoop fs -copyFromLocal LICENSE.txt /user/hduser/wordcount/input
hadoop fs -ls /user/hduser/wordcount/input
```

本機執行 pyspark 程式

Step1 進入 pyspark

```
pyspark --master local[*]
```

Step2. 查看目前的執行模式

```
sc.master
```

Step3 讀取本機檔案

```
textFile=sc.textFile("file:/usr/local/spark/README.md")
```

顯示筆數

```
textFile.count()
```

Step4 讀取 HDFS 檔案

```
textFile=sc.textFile("hdfs://master:9000/user/hduser/wordcount/input/LICENSE.txt")
```

顯示筆數

```
textFile.count()
```

Step5. 離開 pyspark

```
exit()
```

Spark 版 WordCount

```
spark-shell
val textFile = sc.textFile("hdfs://master:9000/user/bdauser/wordcount/input/all-shakespeare.txt")
val stringRDD=textFile.flatMap(line => line.split(" "))
val countsRDD = stringRDD.map(word => (word, 1)).reduceByKey(_ + _)
countsRDD.sortByKey().collect.foreach(println)
```

開啟網址：http://192.168.56.100:4040查看 Spark 工作狀況

```
:q
```

探勘頻繁項目集(Mining Frequent Itemsets)

```
val trans=sc.makeRDD(
      Array(
            Array("牛奶","香蕉","可樂","麵包"),
            Array("麵包","啤酒","尿布"),
            Array("香蕉","牛奶","尿布","餅乾"),
            Array("可樂","尿布","啤酒"),
            Array("啤酒","小蘋果","尿布"),
            Array("尿布","奇異果","啤酒"),
            Array("可樂果","啤酒","冰淇淋","布丁","尿布")
      )
)

val allComb=trans.map{t=>
      for(i<-1 to t.length) yield {
          val eleCom=t.combinations(i)
          val kv=eleCom.map(ele=>"("+ele.sorted.mkString(",")+")")
          kv
      }
}.flatMap(x=>x).flatMap(x=>x)

allComb.collect

allComb.map(x=>(x,1)).reduceByKey(_+_).sortBy(x=>x._2,false).take(10).foreach(println)
```