BiG Data

Atelier 2 : Hadoop - MapReduce (WordCount)

Annassiri Fatima Zahra _ MBisd2

Objectif :

L'objectif de ce TP est de faire une Initiation au Framework Hadoop et au patron MapReduce, utilisation de docker pour lancer un cluster Hadoop de 3 nœuds.
L'intérêt de l'utilisation des centenaires Docker et de garantir la consistance entre les environnements de développement et permettra de réduire considérablement la complexité de configuration des machines (dans le cas d'un accès natif) ainsi que la lourdeur d'exécution (si on opte pour l'utilisation d'une machine virtuelle).

Outils et version:

On utilise pour ce tp :

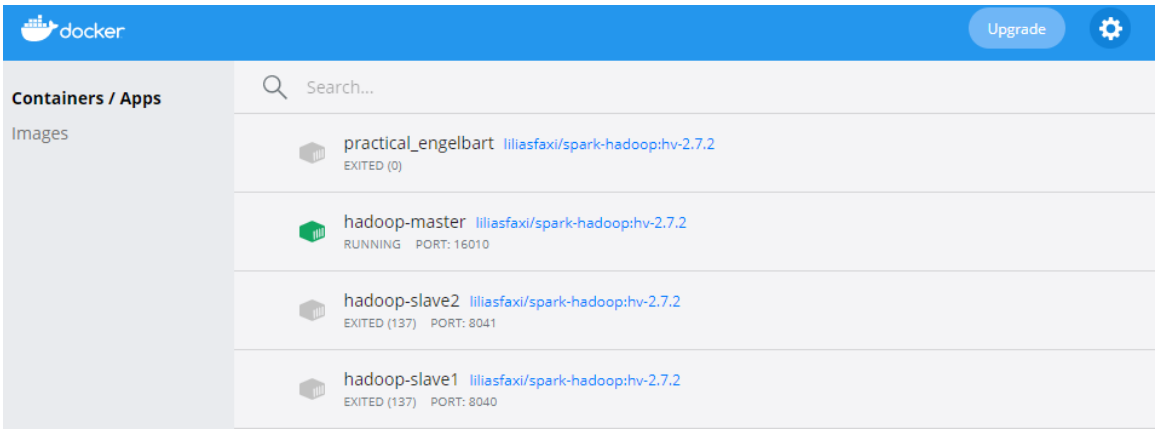
- Apache Hadoop Version: 2.7.2.
- Docker Version 17.09.1
- Java Version 1.8.

Installation :

Télécharger l'image docker uploadée sur dockerhub, ainsi la création et le lancement des trois centenaires ; un master et deux exclave :

```
docker network create --driver=bridge hadoop
```

```
PS C:\> docker pull liliasfaxi/spark-hadoop:hv-2.7.2
hv-2.7.2: Pulling from liliasfaxi/spark-hadoop
1be7f2b886e8: Pull complete
6fbc4a21b806: Pull complete
c71a6f8e1378: Pull complete
4be3072e5a37: Pull complete
06c6d2f59700: Pull complete
b8606274051a: Pull complete
8176485c06ce: Pull complete
f3a132dac987: Pull complete
a3c7183d2677: Pull complete
d010f061a722: Pull complete
d81c164d96f9: Pull complete
d8d441090d24: Pull complete
7c12d721deef: Pull complete
091d1ad175e0: Pull complete
793a639c13bb: Pull complete
040b0d6351fa: Pull complete
262437b95da7: Pull complete
Digest: sha256:56f4243e1b22684301e611df6e724605846f4ddbaf8d8884ef841fc5f2e48a70
Status: Downloaded newer image for liliasfaxi/spark-hadoop:hv-2.7.2
docker.io/liliasfaxi/spark-hadoop:hv-2.7.2
```




```
docker run -itd --net=hadoop -p 50070:50070 -p 8088:8088 -p 7077:7077 -p
16010:16010 \
            --name hadoop-master --hostname hadoop-master \
            liliasfaxi/spark-hadoop:hv-2.7.2

docker run -itd -p 8040:8042 --net=hadoop \
        --name hadoop-slave1 --hostname hadoop-slave1 \
              liliasfaxi/spark-hadoop:hv-2.7.2

docker run -itd -p 8041:8042 --net=hadoop \
        --name hadoop-slave2 --hostname hadoop-slave2 \
              liliasfaxi/spark-hadoop:hv-2.7.2
```

```
PS C:\> docker run -itd -p 8040:8042 --net=hadoop --name hadoop-slave1 --hostname hadoop-slave1  liliasfaxi/spark-hadoop:hv-2.7.2
docker: Error response from daemon: Conflict. The container name "/hadoop-slave1" is already in use by container "bc2140ab6a3f451211b0b91ae750df323b986e1feeab750b215f2ef4764eee37"
```

```
   docker exec -it hadoop-master bash
```

```
PS C:\> docker run -itd -p 8041:8042 --net=hadoop  --name hadoop-slave2 --hostname hadoop-slave2  liliasfaxi/spark-hadoop:hv-2.7.2
37f02b4248cdea3b4f21d8a7430db8a32784ca36d5461929ba6abcf40682c74e
PS C:\> docker run -itd --net=hadoop -p 50070:50070 -p 8088:8088 -p 7077:7077 -p 16010:16010 --name hadoop-master --hostname hadoop-master liliasfaxi/spark-hadoop:hv-2.7.2
fe7aa474433b57a2b2e09209d98379a209e17c3d3d8d464ad07350085571ab39
PS C:\> docker exec -it hadoop-master bash
root@hadoop-master:~#   ./start-hadoop.sh
```

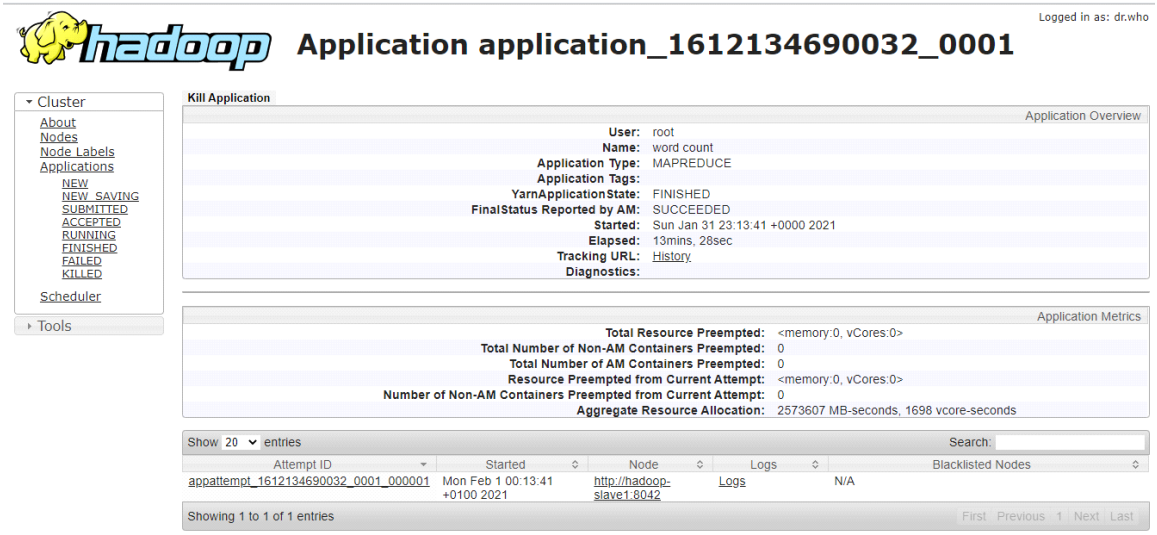
```
  ./start-hadoop.sh
```

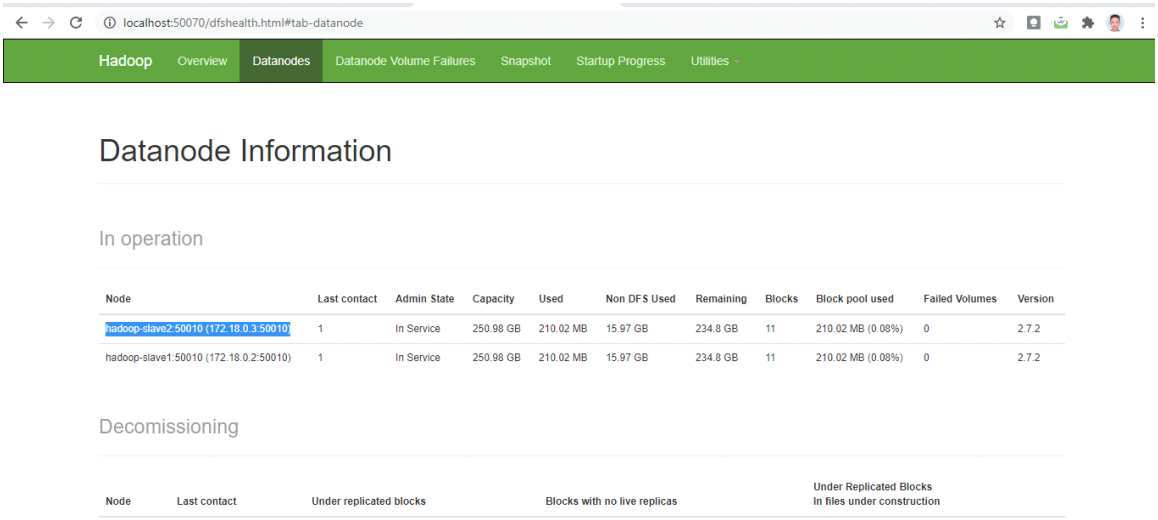
```
PS C:\Users\soufi>  docker exec -it hadoop-master bash
root@hadoop-master:~# ./start-hadoop.sh


Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.4' (ECDSA) to the list of known hosts.
hadoop-master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-hadoop-master.out
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.3' (ECDSA) to the list of known hosts.
hadoop-slave1: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave1.out
hadoop-slave2: datanode running as process 63. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-hadoop-master.out


starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn--resourcemanager-hadoop-master.out
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.3' (ECDSA) to the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.2' (ECDSA) to the list of known hosts.
hadoop-slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave1.out
hadoop-slave2: nodemanager running as process 169. Stop it first.
```

Apres que l'installation est done ! , Nous allons utiliser le fichier PG100.TXT comme entrée pour le traitement MapReduce. Ce fichier se trouve déjà sous le répertoire principal de la machine master.

Hadoop web interfaces

Une fois votre cluster lancé et prêt à l'emploi, vous pouvez, sur votre navigateur préféré de votre machine hôte, aller à : http://localhost:50070. Vous obtiendrez le résultat suivant :





Manipulation de fichier PG100 .txt

Premier pas avec Haddop debuer avec la creation d'un repertoire HDFS nommée Input :

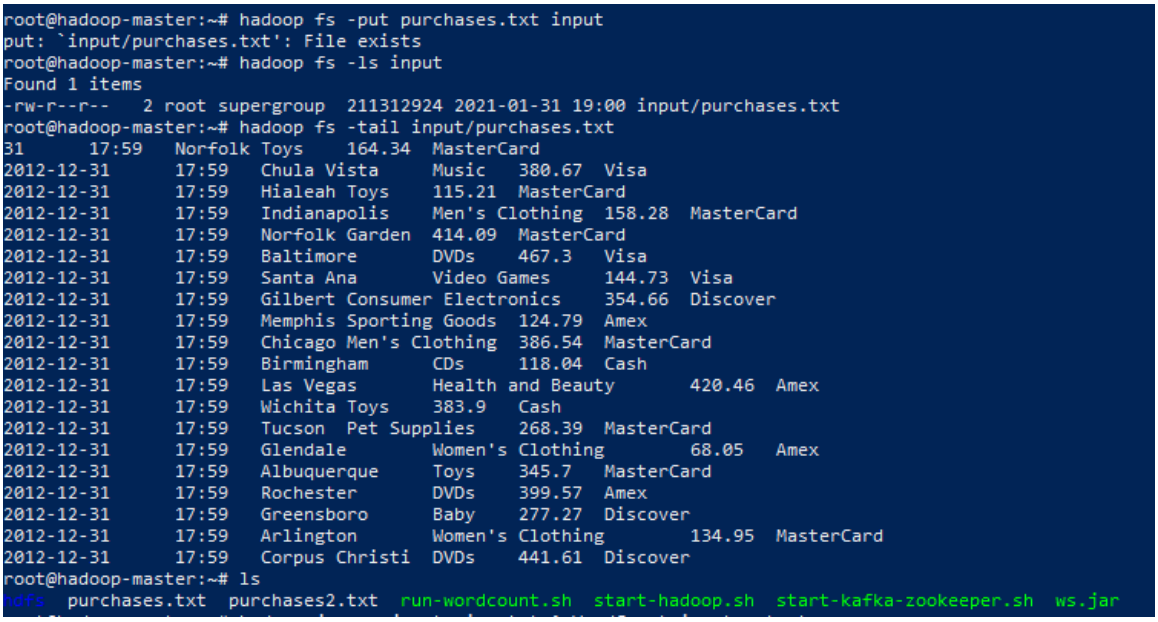
```
hadoop fs –mkdir -p input
```

Ensuite , charger le fichier texte dans le répertoire créer :

```
hadoop fs –put purchases.txt input
```

pour afficher le contenue de fichier texte en utilise :

```
hadoop fs –ls input
```



```
root@hadoop-master:~# hadoop fs -mkdir -p input2
root@hadoop-master:~# hadoop fs -put pg100.txt input2
root@hadoop-master:~# hadoop fs -ls input2
Found 1 items
-rw-r--r--   2 root supergroup    5582655 2021-01-31 23:39 input2/pg100.txt
root@hadoop-master:~# hadoop fs -tail input2/pg100.txt
o lovered?
  Ay me, I fell, and yet do question make
  What I should do again for such a sake.

  'O, that infected moisture of his eye,
  O, that false fire which in his cheek so glowed,
  O, that forced thunder from his heart did fly,
  O, that sad breath his spongy lungs bestowed,
  O, all that borrowed motion, seeming owed,
  Would yet again betray the fore-betrayed,
  And new pervert a reconciled maid.'

THE END



<<THIS ELECTRONIC VERSION OF THE COMPLETE WORKS OF WILLIAM
SHAKESPEARE IS COPYRIGHT 1990-1993 BY WORLD LIBRARY, INC., AND IS
PROVIDED BY PROJECT GUTENBERG ETEXT OF ILLINOIS BENEDICTINE COLLEGE
WITH PERMISSION.  ELECTRONIC AND MACHINE READABLE COPIES MAY BE
DISTRIBUTED SO LONG AS SUCH COPIES (1) ARE FOR YOUR OR OTHERS
PERSONAL USE ONLY, AND (2) ARE NOT DISTRIBUTED OR USED
COMMERCIALLY.  PROHIBITED COMMERCIAL DISTRIBUTION INCLUDES BY ANY
SERVICE THAT CHARGES FOR DOWNLOAD TIME OR FOR MEMBERSHIP.>>



End of this Etext of The Complete Works of William Shakespeare



root@hadoop-master:~#  hadoop jar wordcount-1.jar tn.insat.tp1.WordCount input2 output2
21/01/31 23:41:33 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.18.0.4:8032
21/01/31 23:41:33 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute
.
21/01/31 23:41:33 INFO input.FileInputFormat: Total input paths to process : 1
21/01/31 23:41:33 INFO mapreduce.JobSubmitter: number of splits:1
21/01/31 23:41:34 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1612134690032_0002
21/01/31 23:41:34 INFO impl.YarnClientImpl: Submitted application application_1612134690032_0002
21/01/31 23:41:34 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1612134690032_0002/
21/01/31 23:41:34 INFO mapreduce.Job: Running job: job_1612134690032_0002
21/01/31 23:41:43 INFO mapreduce.Job: Job job_1612134690032_0002 running in uber mode : false
```

Map Reduce¶¶

Un Job Map-Reduce se compose principalement de deux types de programmes:

Mappers : permettent d'extraire les données nécessaires sous forme de clef/valeur, pour pouvoir ensuite les trier selon la clef

```java
package tn.insat.tp1;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

import java.io.IOException;
import java.util.StringTokenizer;

public class TokenizerMapper
      extends Mapper<Object, Text, Text, IntWritable>{

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(Object key, Text value, Mapper.Context context
  ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
          word.set(itr.nextToken());
          context.write(word, one);
      }
  }
}
```

Reducers : prennent un ensemble de données triées selon leur clef, et effectuent le traitement nécessaire sur ces données (somme, moyenne, total...)

```java
package tn.insat.tp1;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class IntSumReducer
      extends Reducer<Text,IntWritable,Text,IntWritable> {

   private IntWritable result = new IntWritable();

   public void reduce(Text key, Iterable<IntWritable> values,
                      Context context
   ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
          System.out.println("value: "+val.get());
          sum += val.get();
      }
      System.out.println("--> Sum = "+sum);
      result.set(sum);
      context.write(key, result);
   }
}
```

Drivers :

```java
package tn.insat.tp1;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {
    Run | Debug
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Job Result :

Pour lancer le job en utilise la commande suivante en mode shell :

```
hadoop jar wordcount-1.jar tn.insat.tp1.WordCount input output
```

on obtient :

```
root@hadoop-master:~# hadoop fs -tail output2/part-r-00000
you'st  1
you,    1428
you-    45
you--   1
you--you        1
you-I   1
you-he  1
you-often       1
you-pray        1
you-that        1
you-well,       1
you-wondrous    1
you.    811
you.'   4
you.-   5
you:    29
you;    261
you?    259
you?'   3
young   345
young'  1
young's 1
young'st        1
young,  36
young-ey'd      1
young.  9
young;  8
young?  2
younger 26
younger,        2
younger.        2
youngest        21
youngest,       1
youngest;       1
younglings,     1
youngly 2
younker 3
your    6009
your-   1
your@login      1
yours   77
yours!  3
yours,  60
```

- The end