

=====

TP1 Le traitement Batch avec Hadoop HDFS

=====

Objectifs du TP

Initiation au framework hadoop et au patron MapReduce, utilisation de docker pour lancer un cluster hadoop de 3 noeuds.

Outils et Versions

- Apache Hadoop Version : 2.7.2.
- Docker Version 17.09.1
- IntelliJ IDEA Version Ultimate 2016.1 (ou tout autre IDE de votre choix)
- Java Version 1.8.
- Unix-like ou Unix-based Systems (Divers Linux et MacOS)

Installation

Nous allons utiliser tout au long de ce TP trois conteneurs représentant respectivement un noeud maître (Namenode) et deux noeuds esclaves (Datanodes).

Vous devez pour cela avoir installé docker sur votre machine, et l'avoir correctement configuré. Ouvrir la ligne de commande, et taper les instructions suivantes :

1. Télécharger l'image docker uploadée sur dockerhub:

```
docker pull hajjitarikensam/hadoop:ensam
```

2. Créer les trois conteneurs à partir de l'image téléchargée. Pour cela : 2.1. Créer un réseau qui permettra de relier les trois conteneurs:

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

3. Créer et lancer les trois conteneurs (les instructions -p permettent de faire un mapping entre les ports de la machine hôte et ceux du conteneur):

```
docker run -itd --net=hadoop -p 50070:50070 -p 8088:8088 -p 7077:7077 -p 16010:16010 --name hadoop-master --hostname hadoop-master hajjitarikensam/hadoop:ensam
```

```
docker run -itd -p 8040:8042 --net=hadoop --name hadoop-slave1 --hostname hadoop-slave1 hajjitarikensam/hadoop:ensam
```

```
docker run -itd -p 8041:8042 --net=hadoop --name hadoop-slave2 --hostname hadoop-slave2 hajjitarikensam/hadoop:ensam
```

4. Entrer dans le conteneur master pour commencer à l'utiliser :

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

Le résultat de cette exécution sera le suivant :

```
root@hadoop-master:~#
```

Vous vous retrouverez dans le shell du namenode, et vous pourrez ainsi manipuler le cluster à votre guise. La première chose à faire, une fois dans le conteneur, est de lancer hadoop et yarn. Un script est fourni pour cela, appelé start-hadoop.sh. Lancer ce script.

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

Le résultat devra ressembler à ce qui suit :

```
root@hadoop-master:~# ./start-hadoop.sh
[
Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.22.0.2' (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-slave1,172.22.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.22.0.4' (ECDSA) to the list of known hosts.
hadoop-slave2: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave2.out
hadoop-slave1: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave1.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
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-slave2,172.22.0.4' (ECDSA) to the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.22.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave2.out
hadoop-slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave1.out
```

Premiers pas avec Hadoop

Toutes les commandes interagissant avec le système Hadoop commencent par hadoop fs. Ensuite, les options rajoutées sont très largement inspirées des commandes Unix standard.

- Créer un répertoire dans HDFS, appelé *input*. Pour cela, taper :

```
hadoop fs -mkdir -p input
```

Si pour une raison ou une autre, vous n'arrivez pas à créer le répertoire input, avec un message ressemblant à ceci: ls: `.`: No such file or directory, veiller à construire l'arborescence de l'utilisateur principal (root), comme suit:

```
hadoop fs -mkdir -p /user/root
```

Nous allons utiliser le fichier purchases.txt comme entrée pour le traitement MapReduce. Ce fichier se trouve déjà sous le répertoire principal de votre machine master.

- Charger le fichier purchases dans le répertoire input que vous avez créé :

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

- Pour afficher le contenu du répertoire input, la commande est:

```
hadoop fs -ls input
```

- Pour afficher les dernières lignes du fichier purchases:

```
hadoop fs -tail input/purchases.txt
```

Le résultat suivant va donc s'afficher :

```
[root@hadoop-master:~# hadoop fs -tail input/purchases.txt
31      17:59  Norfolk Toys      164.34  MasterCard
2012-12-31    17:59  Chula Vista      Music   380.67  Visa
2012-12-31    17:59  Hialeah Toys    115.21  MasterCard
2012-12-31    17:59  Indianapolis    Men's Clothing 158.28  MasterCard
2012-12-31    17:59  Norfolk Garden  414.09  MasterCard
2012-12-31    17:59  Baltimore      DVDs    467.3   Visa
2012-12-31    17:59  Santa Ana      Video Games   144.73  Visa
2012-12-31    17:59  Gilbert Consumer Electronics  354.66  Discover
2012-12-31    17:59  Memphis Sporting Goods 124.79  Amex
2012-12-31    17:59  Chicago Men's Clothing 386.54  MasterCard
2012-12-31    17:59  Birmingham     CDs     118.04  Cash
2012-12-31    17:59  Las Vegas      Health and Beauty 420.46  Amex
2012-12-31    17:59  Wichita Toys   383.9   Cash
2012-12-31    17:59  Tucson Pet Supplies 268.39  MasterCard
2012-12-31    17:59  Glendale      Women's Clothing 68.05   Amex
2012-12-31    17:59  Albuquerque    Toys    345.7   MasterCard
2012-12-31    17:59  Rochester     DVDs    399.57  Amex
2012-12-31    17:59  Greensboro     Baby    277.27  Discover
2012-12-31    17:59  Arlington     Women's Clothing 134.95  MasterCard
2012-12-31    17:59  Corpus Christi DVDs    441.61  Discover
root@hadoop-master:~#
```

Nous présentons dans le tableau suivant les commandes les plus utilisées pour manipuler les fichiers dans HDFS :

Instruction	Fonctionnalité
<code>hadoop fs -ls</code>	Afficher le contenu du répertoire racine
<code>hadoop fs -put file.txt</code>	Upload un fichier dans hadoop (à partir du répertoire courant linux)
<code>hadoop fs -get file.txt</code>	Download un fichier à partir de hadoop sur votre disque local
<code>hadoop fs -tail file.txt</code>	Lire les dernières lignes du fichier
<code>hadoop fs -cat file.txt</code>	Affiche tout le contenu du fichier
<code>hadoop fs -mv file.txt newfile.txt</code>	Renommer le fichier
<code>hadoop fs -rm newfile.txt</code>	Supprimer le fichier
<code>hadoop fs -mkdir myinput</code>	Créer un répertoire
<code>hadoop fs -cat file.txt \ less</code>	Lire le fichier page par page

Interfaces web pour Hadoop

Hadoop offre plusieurs interfaces web pour pouvoir observer le comportement de ses différentes composantes. Vous pouvez afficher ces pages en local sur votre machine grâce à l'option `-p` de la commande `docker run`. En effet, cette option permet de publier un port du conteneur sur la machine hôte. Pour pouvoir publier tous les ports exposés, vous pouvez lancer votre conteneur en utilisant l'option `-P`.

En regardant le contenu du fichier `start-container.sh` fourni dans le projet, vous verrez que deux ports de la machine maître ont été exposés :

Le port 50070 : qui permet d'afficher les informations de votre namenode.

Le port 8088 : qui permet d'afficher les informations du resource manager de Yarn et visualiser le comportement des différents jobs.

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 :

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Overview 'hadoop-master:9000' (active)


Started:	Fri Jan 26 11:47:09 UTC 2018
Version:	2.7.2, rUnknown
Compiled:	2016-05-27T18:05Z by root from Unknown
Cluster ID:	CID-3c662456-d44e-4301-bc39-28e479c4dc88
Block Pool ID:	BP-431089505-172.17.0.2-1465730089024

Summary

Security is off.
Safemode is off.
20 files and directories, 8 blocks = 28 total filesystem object(s).
Heap Memory used 85.33 MB of 165.5 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 37.42 MB of 38.44 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	125.49 GB
DFS Used:	406.85 MB (0.32%)
Non DFS Used:	34.1 GB

Vous pouvez également visualiser l'avancement et les résultats de vos Jobs (Map Reduce ou autre) en allant à l'adresse : <http://localhost:8088>

 **All Applications** Logged in as: dr.who

Cluster

About Nodes Node Labels Applications NEW NEW SAVING SUBMITTED ACCEPTED RUNNING FINISHED FAILED KILLED Scheduler Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	16 GB	0 B	0	16	0	2	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:32>

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
No data available in table											

Showing 0 to 0 of 0 entries

First Previous Next Last