Département de Mathématiques et Informatique
Master Sciences des Données et Analytiques – S2
Module Traitement Distribué
Pr. Mourdi Youssef

LabWork 1

Installation et Configuration de Hadoop 2

1. Objectifs

L'objectif de ce premier atelier est d'installer et de configurer Hadoop 2 sur une machine virtuelle Unix. L'installation et la configuration concerneront dans un premier lieu un simple cluster. Les autres ateliers permettront la configuration et la liaison de plusieurs clusters en implémentant un exemple Java.

2. Environnement

Puisque l'installation se fera sur une machine virtualisée Linux, on aura besoin de réaliser les téléchargements suivants :

- VMware Workstation 10 ou VM Virtual Box
- Ubuntu version 16+ (ici la version 22 est utilisée)

Hadoop est écrit en Java, il faut donc l'installer sur la machine virtuelle (version 7 ou ultérieure). Pour ce LabWork, la version 11 de Java sera utilisée via la distribution OpenJDK.

```
sudo apt-get install openjdk-11-jdk
```

Afin de vérifier le succès de l'installation : **java -version**

3. Installation

Dans cet atelier, on installe la dernière version stable de Hadoop. La plus stable à la rédaction de cet atelier est la **version 3.4.0.** Ici la version 3.3.6 est installée

Dans le site officiel d'Apache Hadoop, on télécharge le fichier compressé **« hadoop-3.3.6.tar.gz »**

On décompresse l'archive de Hadoop via la commande suivante :

```
tar xzf hadoop-3.3.6.tar.gz
```

On déplace le répertoire hadoop-3.3.4 vers /usr/local sous le nom hadoop :

```
sudo mv hadoop-3.3.6 /usr/local/hadoop
```

Maintenant, il faut signaler à Hadoop l'emplacement de Java sur le système. Seul le fichier de configuration utilisateur .bashrc a besoin d'être modifié afin qu'il propage les nouvelles valeurs des variables d'environnement.

On ouvre le fichier /home/.bashrc (avec la commande « nano » ou « gedit »)

```
nano /home/.bashrc
```

Puis on ajoute à la fin du fichier les lignes suivantes : (PS : Vérifiez le chemin de Java dans votre machine)

```
# Java Environment Variable
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
# Hadoop Environment Variables
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

Le fichier **$HADOOP_HOME/etc/hadoop/hadoop-env.sh** contient des variables d'environnement utilisées par Hadoop. On décommente (ou on modifie) celle de la variable JAVA_HOME.

On redémarre la machine pour que toutes les modifications soient prises en compte. Après relancement, on vérifie que Hadoop est bien installé: **hadoop version**

4. Configuration

Tous les fichiers de configuration de Hadoop sont disponibles dans le répertoire /etc/hadoop.
Les fichiers de configuration de Hadoop fonctionnent sur le principe de clé/valeur : la clé correspondant au nom du paramètre et valeur à la valeur assignée à ce paramètre. Ces

fichiers de configuration utilisent le format XML. Les nouveaux paramètres sont à ajouter entre la balise **<configuration>…</configuration>.**

Chaque propriété de ce fichier est de la forme :

```
<property>
   <name>nom de la propriété</name>
   <value>valeur de la propriété</value>
</property>
```

1. *core-site.xml*

Le fichier **$HADOOP_HOME/etc/hadoop/core-site.xml** est le fichier général de la configuration de la plate-forme. Dans ce fichier, on modifie le contenu afin d'obtenir le résultat ci-dessous :

```
<configuration>
  <property>
     <name>fs.defaultFS</name>
     <value>hdfs://localhost:9000</value>
     <description>The name of the default file system.</description>
   </property>
</configuration>
```

- La propriété **fs.defaultFS** permet de spécifier quant à elle le nom du système de fichier.
- **Hdfs://localhost :9000** est utilisé pour définir un système de fichiers par défaut pour Hadoop. Les démons HDFS vont utiliser cette propriété pour déterminer l'hôte et le port du NameNode HDFS.

2. *hdfs-site.xml*

Le fichier **$HADOOP_HOME/etc/hadoop/hdfs-site.xml** contient les paramètres spécifiques au système de fichiers HDFS. Il désigne l'endroit où vous souhaitez stocker l'infrastructure Hadoop.

```
<configuration>

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

  <property>
     <name>dfs.namenode.name.dir</name>
     <value>file:/usr/local/hadoop/hadoopdata/hdfs/namenode</value>
  </property>

  <property>
     <name>dfs.datanode.data.dir</name>
     <value>file:/usr/local/hadoop/hadoopdata/hdfs/datanode</value>
  </property>

</configuration>
```

Créez les répertoires de NameNode et DataNode et changez le propriétaire du dossier «hadoop» et les dossiers et fichiers contenus à l'intérieur de ce dossier.

```
mkdir -p /usr/local/hadoop/hadoopdata/hdfs/namenode
mkdir -p /usr/local/hadoop/hadoopdata/hdfs/datanode
```

3. *mapred-site.xml*

Le fichier **$HADOOP_HOME/etc/hadoop/mapred-site.xml** contient les paramètres spécifiques à MapReduce. Depuis la version 2.x de Hadoop avec l'arrivée de Yarn, ce fichier de configuration est épaulé par **yarn-site.xml**.

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

4. *yarn-site.xml*

On doit configurer le fichier **$HADOOP_HOME/etc/hadoop/yarn-site.xml.**

```
<configuration>
   <property>
       <name>yarn.nodemanager.aux-services</name>
       <value>mapreduce_shuffle</value>
   </property>
   <property>
```

```
   <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
   <value>org.apache.hadoop.mapred.ShuffleHandler</value>
 </property>
</configuration>
```

5. *Configuration SSH*

Hadoop nécessite un accès SSH pour gérer les différents nœuds.

On s'assure d'abord que SSH est bien installé.

```
sudo apt-get install ssh
```

Par la suite, on crée une paire de clés RSA (publique/privée) avec un mot de passe vide. Même si cet atelier se focalise sur l'utilisation de Hadoop sur un seul nœud et en localhost, il faudra prendre cette habitude de créer les clés RSA avec mot de passe dans le cas où le contexte changera.

```
ssh-keygen -t rsa -P ""
```

On doit autoriser l'accès au SSH de la machine avec cette nouvelle clé fraîchement créée.

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

La dernière chose à réaliser est de tester la connexion SSH.

```
ssh localhost
```

5. Initialiser et tester le système Hadoop

Avant de démarrer le serveur Hadoop, on doit formater le système de fichiers HDFS.

```
hdfs namenode -format
```

Si la commande « hdfs namenode –format » n'a pas fonctionné convenablement, essayez la commande suivante : **hadoop namenode -format**

Pour démarrer Hadoop, on doit démarrer le système de fichiers HDFS et le serveur MapReduce dans le cas où on souhaite utiliser des jobs MapReduce.

```
start-dfs.sh

start-yarn.sh
```

Pour s'assurer que tout fonctionne, on utilise l'outil jps pour lister les processus Java en cours d'exécution : **jps**

Depuis la version 2.3.x de Hadoop, ResourceManager remplace JobTracker.

On peut vérifier si les démons ont commencé avec succès à l'adresse http://localhost:8088/ pour le ResourceManager et à l'adresse http://localhost:9870/ pour la NameNode.