Faculté Polydisciplinaire de Safi

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

LabWork 3

Configuration de Hadoop sur plusieurs machines

1. Objectifs

Cet atelier a pour objectif de décrire la mise en œuvre et la configuration complète d'un cluster Hadoop multi-nœuds. Pour tester notre système Hadoop, on a proposé comme solution d'exécuter un traitement Map-Reduce.

2. Architecture d'un cluster Hadoop

Avant de configurer les nœuds maître et le Workers, il est important de comprendre les différents composants d'un cluster Hadoop.

Un **nœud maître** conserve les informations sur le système de fichiers distribué, comme la « inodetable » d'un système de fichiers ext3, et planifie l'allocation des ressources. Le **node-master** assumera ce rôle dans cet atelier et hébergera deux démons :

- Le **NameNode** : gère le système de fichiers distribué et sait où se trouvent les blocs de données stockés à l'intérieur du cluster.
- Le **ResourceManager** : gère les tâches YARN et s'occupe de la planification et de l'exécution des processus sur les nœuds de travail.

Les nœuds Workers stockent les données et fournissent la puissance de traitement nécessaire à l'exécution des tâches. Il s'agit **des nœuds 1** et **2**, qui hébergeront deux démons :

- Le **DataNode** : gère les données physiques stockées sur le nœud ; il est nommé NameNode.
- Le **NodeManager** gère l'exécution des tâches sur le nœud.

3. Installation et Configuration d'un cluster Hadoop

Pour l'installation et les tests, nous aurons besoin de 3 machines virtuelles avec un système Linux Ubuntu. Une machine jouera le rôle du Master Node et les deux autres seront des slaves.

Sur la machine Master, l'installation et la configuration correcte de Java et Hadoop (mêmes versions de préférence) est nécessaire. Il faudra donc se référer aux étapes du **LabWork 1**.

3.1. Configuration du réseau du Cluster

3.1.1. Sur la machine Master

a) Accéder aux paramètres réseau de la machine virtuelle et activez l'adaptateur 2. Ensuite, au lieu de « NAT », choisir « Host Only ».
b) Installer SSH à l'aide de la commande suivante :

```
sudo apt install ssh
```

c) Une fois SSH installé, on crée une nouvelle clé à l'aide de la commande :

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

d) On copie la clé publique dans le fichier authorized_keys avec la commande suivante :

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

e) On vérifie la configuration SSH en se connectant à l'hôte local

```
ssh localhost
```

f) Modifier le fichier de configuration Netplan : les fichiers de configuration Netplan sont stockés dans /etc/netplan/. Vous trouverez ce fichier dans le répertoire /etc/netplan

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
```

```
    addresses:
      - 192.168.1.1/24
```

Remplacez « enp0s3 » par le nom de votre interface. Une fois terminé, appliquez la nouvelle configuration avec la commande :

```
sudo netplan apply
```

Vérifiez l'affectation de la nouvelle configuration réseau avec la commande :

```
ip a
```

3.1.2. Création des Slaves

a) Arrêter la machine virtuelle Master et cloner-la deux fois, en nommant un slave1 et l'autre slave2. Il faut que la copie soit intégrale et que chaque machine disposera d'une adresse MAC dédiée.
b) Refaire l'étape 3.1.1 ➔ a ➔ b ➔ f pour les deux machines slaves et attribuez respectivement les adresses IP 192.168.1.2 et 192.168.1.3 pour les machines slave1 et slave2
c) Sur la machine virtuelle principale, on ouvre le fichier de nom d'hôte avec nano:

```
sudo nano /etc/hostname
```

d) On insère le nom de la machine virtuelle principale : **master**
e) On fait de même sur les esclaves : slave1 et slave2
f) On ouvre le fichier hosts dans les 3 machines et on y insère les configurations réseau :

```
sudo nano /etc/hosts
```

```
192.168.1.1 master
192.168.1.2 slave1
192.168.1.3 slave2
```

```
```

g) Assurez vous que la connexion est établie entre les trois machines avec la commande **ping**.
h) Sur la machine master, on tape les commandes suivantes

```
ssh-copy-id user@master
ssh-copy-id user@slave1
ssh-copy-id user@slave2
```

Changez user avec le nom d'utilisateur que vous avez créé.

4. Configuration du nœud Master

4.1. Définir l'emplacement du NameNode

Mettez à jour le fichier core-site.xml pour définir l'emplacement NameNode sur la machine master sur le port9000

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
    <configuration>
        <property>
            <name>fs.default.name</name>
            <value>hdfs://master:9000</value>
        </property>
    </configuration>
```

4.2. Définir le chemain pour HDFS

Modifiez hdfs-site.conf pour ressembler à la configuration suivante :

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

```
        <value>/usr/local/hadoop/data/namenode</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>/usr/local/hadoop/data/datanode</value>
    </property>
</configuration>
```

La propriété, dfs.replication, indique le nombre de réplications de données dans le cluster. Vous pouvez configurer la duplication de toutes les données sur les deux nœuds. Ne saisissez pas une valeur supérieure au nombre réel de nœuds worker.

4.3. Définir YARN comme planificateur de tâches

Modifiez le fichier mapred-site.xml en définissant YARN comme framework par défaut pour les opérations MapReduce

```
<configuration>
    <property>
            <name>mapreduce.framework.name</name>
            <value>yarn</value>
    </property>
    <property>
            <name>yarn.app.mapreduce.am.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
            <name>mapreduce.map.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
            <name>mapreduce.reduce.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
</configuration>
```

4.4. Configurer YARN

Modifier yarn-site.xml, qui contient les options de configuration de YARN. Dans la valeur du champ correspondant « yarn.resourcemanager.hostname », mettrez par l'adresse IP du nœud maître (qui est 192.168.1.1)

```
<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>
    <!-- Site specific YARN configuration properties -->
    <property>
        <name>yarn.application.classpath</name>
        <value>
$HADOOP_CONF_DIR,$HADOOP_COMMON_HOME/share/hadoop/common/*,$HADOOP_COMMON_HO
ME/share/hadoop/common/lib/*,$HADOOP_HDFS_HOME/share/hadoop/hdfs/*,$HADOOP_H
DFS_HOME/share/hadoop/hdfs/>

$HADOOP_YARN_HOME/share/hadoop/yarn/*,$HADOOP_YARN_HOME/share/hadoop/yarn/li
b/*
       </value>
    </property>
    <property>
       <name>yarn.acl.enable</name>
       <value>0</value>
    </property>
    <property>
       <name>yarn.resourcemanager.hostname</name>
       <value>192.168.1.1</value>
    </property>
</configuration>
```

4.5. Configurer les Workers

Le fichier « workers » est utilisé par les scripts de démarrage pour démarrer les démons requis sur tous les nœuds. Modifiez-le pour inclure les deux nœuds slave1 et slave2

4.6. Dupliquez les fichiers de configuration sur chaque nœud du cluster

Copiez les fichiers de configuration Hadoop sur les nœuds workers

```
scp $HADOOP_HOME /etc/hadoop/* slave1: $HADOOP_HOME /etc/hadoop/
scp $HADOOP_HOME /etc/hadoop/* slave2: $HADOOP_HOME /etc/hadoop/
```

5. Test sur l'environnement Hadoop

a) On commence par formater le système de fichier HDFS avec la commande (une opération à lancer sur le Master): hdfs namenode –format
b) On démarre HDFS avec la commande : **start-dfs.sh**
c) Sur les 3 machines, on teste avec la commande **jps** que le namenode et secondarynamenode ont bien démarrés sur Master et que le processus DataNode a démarré sur les esclaves.
d) Sur le navigateur, on tape : **master :9870**
e) Dans les 2 esclaves, on ouvre le fichier yarn-site.xml, on modifie la configuration yarn.resourcemanager.hostname avec la valeur **master**
f) Sur la machine Master, on exécute la commande : **start-yarn.sh**
g) Sur le navigateur, on tape l'adresse : **master :8088/cluster**