Travaux pratiques préparatoires : HBase

Jonathan Lejeune

Objectifs

Ce sujet de travaux pratiques vous indique comment installer HBase et vous initiera à son utilisation dans un environnement UNIX.

Prérequis

Vous devez être familier avec l'utilisation de la plate-forme Hadoop. Votre machine devra avoir une installation de Hadoop version 3 ou supérieure afin d'assurer le bon fonctionnement de Hbase. Vous devez également pouvoir vous connecter par clé ssh aux différentes machines que vous utiliserez. Si tel n'est pas le cas, reportez-vous au TP prépatoire de SSH et Hadoop.

Introduction

Hbase est un SGBD NoSQL orienté colonne et fait partie de l'écosystème Hadoop. Son système de stockage repose dans la plupart des cas sur le HDFS et il est possible d'interroger les données via le map-reduce.

Hbase repose sur une architecture maître esclave. Le serveur maître (**Hmaster**) gère la distribution de la responsabilité des données des différentes tables sur les serveurs esclaves (les **RegionServer**). Les machines esclaves stockent des fragments de table et interagissent directement avec les clients lors des requêtes de lecture ou d'écriture. On pourra noter également que Hbase repose sur le service ZooKeeper pour stocker des métadonnées et pour détecter les défaillances des différents serveurs.

Exercice 1 – Installation et démarrage

Question 1

Tapez la commande :

```
cat /etc/hosts
```

Vérifiez que la sortie de cette commande comporte bien les lignes suivantes

```
127.0.0.1 localhost
127.0.1.1 <nom de votre machine locale>
```

Question 2

Téléchargez la dernière version stable de Hbase à l'URL suivante.

`http://archive.apache.org/dist/hbase/stable/`

.

Extrayez le contenu de l'archive dans votre home (attention aux quota). Au même titre qu'Hadoop, Hbase a besoin que la variable d'environnement JAVA_HOME soit définie. **Si ce n'est pas déjà fait** ajoutez la ligne suivante dans votre *.bashrc* :

```
export JAVA_HOME=<chemin vers la JVM courante de votre machine>
```

Ajouter ensuite les lignes suivantes :

```
export MY_HBASE_HOME=<votre_repertoire_installation_hbase>
export PATH=$PATH:$MY_HBASE_HOME/bin
```

Pour prendre en compte ces modifications dans votre terminal ouvert, tapez :

```
source ~/.bashrc
```

Vérifiez que Hbase est opérationnel en tapant :

```
hbase version
```

La sortie terminale de cette commande doit contenir le pattern suivant :

```
HBase x.y.z
Source code repository .....................
Compiled by .... on ....
From source with checksum .....
```

N.B. : il est possible que le message d'erreur suivant (qui est sans importance) apparaisse :

```
HADOOP_ORG.APACHE.HADOOP.HBASE.UTIL.GETJAVAPROPERTY_USER: nom de variable invalide
HADOOP_ORG.APACHE.HADOOP.HBASE.UTIL.GETJAVAPROPERTY_USER: nom de variable invalide
```

Question 3

À l'instar de Hadoop, Hbase possède trois modes d'exécution : local, pseudo-distribué et distribué. Nous allons configurer Hbase en mode pseudo-distribué (tous les démons Hbase en une seule machine) sur le HDFS. Pour cela, éditez le fichier *conf/hbase-site.xml* en ajoutant les properties suivantes :

—
```
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
```
Ceci indique que Hbase sera en mode distribué (une JVM par démon).

—
```
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://localhost:9000/hbase</value>
</property>
```
Ceci indique le chemin HDFS à utiliser pour stocker les données. Ici ça sera le dossier */hbase*.

—
```
<property>
  <name>hbase.zookeeper.property.dataDir</name>
  <value>/tmp/zookeeper</value>
</property>
```
Ceci indique où Zookeeper stockera ses informations sur votre système de fichiers local.

—
```
<property>
  <name>hbase.wal.provider</name>
  <value>filesystem</value>
</property>
```
Ceci permet de palier un bug sur l'interfaçage entre Hbase 2.4.7 et Hadoop 3.3.1.

Question 4

Démarrez les démons du HDFS (pensez éventuellement à refaire au préalable un formatage de celui-ci si nécessaire) et de Hbase. Il est inutile démarrer Yarn ici.

```
hdfs namenode -format #seulement si necesaire
start-dfs.sh
start-hbase.sh
```

La commande `start-hbase.sh` démarre trois démons : Zookeeper, hbase master et regionserver. Vérifiez que tout fonctionne grâce au script check_start.sh fourni en ressources des TP.

Question 5

Vérifiez que le dossier *hbase* a bien été créé sur le HDFS et que son contenu soit (pour la version 2.4.7) :

```
Found 12 items
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/.hbck
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/.tmp
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/MasterData
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/WALs
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/archive
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/corrupt
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/data
-rw-r--r--   1 jlejeune supergroup         42 2021-11-03 16:35 /hbase/hbase.id
-rw-r--r--   1 jlejeune supergroup          7 2021-11-03 16:35 /hbase/hbase.versio
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/mobdir
drwxr-xr-x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/oldWALs
drwx--x--x   - jlejeune supergroup          0 2021-11-03 16:35 /hbase/staging
```

Question 6

Contrairement à Hadoop, il est possible de lancer plusieurs démons du même type sur la même machine. Nous allons démarrer 3 regionServer supplémentaires. Bien que ceci n'ait pas beaucoup de sens en réalité, ce mécanisme nous permettra cependant d'émuler un cluster en considérant plusieurs nœuds regionServer. Le regionServer écoute sur 2 ports : 16020 et 16030. Pour éviter le conflit de port d'écoute, il existe une commande qui permet de démarrer des démons supplémentaires en décalant le numéro de port d'écoute de ces derniers. Pour cela, tapez :

```
local-regionservers.sh start 2 3 4
```

Ceci démarrera trois regionServer supplémentaires, écoutant respectivement sur 16022/16032, 16023/16033 et 16024/16034

N.B. : Sur ce même principe, il est également possible de lancer des masters de backup en local mais nous nous limiterons à un seul master ici. Vous pouvez à nouveau vérifier le bon démarrage de ces démons via le script `check_start.sh`

Question 7

Une fois que la plate-forme est démarrée, chaque démon démarre un serveur web sur la machine hôte (localhost) afin de pouvoir superviser son état à l'aide d'un navigateur. Vous pouvez voir l'interface Web du master au port 16010 et les interfaces des différents RegionServer au port 1603X où X représente l'offset.

Question 8

Pour taper des commandes shell Hbase, il vous est possible de lancer la commande :

```
hbase shell
```

qui vous ouvrira un nouveaux shell spécifique à Hbase et vous permettra de lancer des commandes Hbase en mode interactif (pour revenir au shell bash, taper *exit*). Il est également possible d'utiliser le shell hbase en mode non interactif, c'est à dire en lançant les commandes Hbase directement depuis le shell bash. Pour cela il suffit d'envoyer la commande à exécuter sur l'entrée standard de *hbase shell*. Affichez le contenu de la table *hbase* : *meta* avec le mode non interactif.

```
echo "scan 'hbase:meta' " | hbase shell -n
```

Question 9

Pour arrêter les regionServers additionnels, taper

```
local-regionservers.sh stop 2 3 4
```

Pour arréter l'ensemble des démons de Hbase, taper

```
stop-hbase.sh
```

Exercice 2 – Configuration d'Eclipse

Afin d'utiliser *Eclipse* pour éditer et compiler des programmes clients Hbase, il faut déclarer une nouvelle librairie permettant de déclarer dans le build path d'un projet Eclipse (= le classpath du point de vue de la JVM) les différents fichiers jars de Hbase.

Pour ce faire :

- Dans la barre de menu de Eclipse cliquez sur *Window*
- Cliquez sur *Preferences*
- Déroulez l'onglet *Java* sur le panneau de gauche
- Déroulez l'onglet *Build Path*
- Cliquez sur le menu *User Libraries*
- Cliquez sur le bouton *New...*
- Nommez la librairie `hbase-x.y.z` (où `x`, `y` et `z` sont à remplacer par la version que vous avez téléchargée)
- Sélectionnez la nouvelle librairie créée et cliquez sur le bouton *Add External JARs...*
- Ajoutez l'ensemble des fichiers *Jar* présents dans l'arborescence du dossier *lib* du dossier d'installation.

— cliquez sur *Apply and Close* pour valider.

Vous pouvez ainsi ajouter la librairie créée précédemment afin de programmer avec l'API de Hbase dans un projet Eclipse. Pour ce faire :

— Sélectionnez *Build Path* dans le menu contextuel (clic droit sur le dossier) du projet puis *Add Libraries...*
— Sélectionnez *User Library* puis *Next >*
— Cochez la librairie puis cliquez sur *Finish.*
— Vérifiez que la librairie a bien été ajoutée au projet dans le Package Explorer d'Eclipse.

Attention, il peut y avoir des conflits de nom de classe lors de l'ajout automatique des import. Le cas échéant, choisissez les classes issues des packages :

— org.apache.hadoop.mapreduce
— org.apache.hadoop.fs
— org.apache.hadoop.conf
— org.apache.hadoop.fs
— org.apache.hadoop.io
— org.apache.hadoop.hbase.client
— org.apache.hadoop.hbase.mapreduce
— org.apache.hadoop.hbase.util