Plateforme Hidoop

Projet de données réparties

2020 - 2021

LAPLAGNE Chloé
RAZAFIMANANTSOA Nathan
GUILLAUD Thomas
GRUNIG Axel

Ce document présente le fonctionnement global de la plateforme.

Le répertoire *hidoop* correspond à l'arborescence de fichiers suivante :

- le répertoire **config** contient les fichiers d'initialisation pouvant être utiles lors du lancement de la plateforme
- le répertoire **scripts** contient les scripts internes à la plateforme
- le répertoire **data** accueille les fichiers de données de l'application
- le répertoire **doc** accueille les rapports attendus
- le répertoire **src** contient les codes sources

Environnement d'utilisation

Avant toute chose, sur chaque machine exécutant Hidoop (client ou serveur), il est nécessaire de définir une variable système **HIDOOP_HOME** indiquant la localisation du répertoire *hidoop* local.

La plateforme s'utilise via l'interpréteur de commande bash ouvert par le script **hidoop**.

Commande : `$ ./hidoop <ssh_username>`

Il permet notamment de lancer et d'arrêter automatiquement les serveurs indiqués dans le fichier de configuration *conf.xml* via ssh (machines N7).

Les commandes spécifiques sont les suivantes :

- ***start*** pour lancer le namenode et les serveurs
- ***stop*** pour arrêter le namenode et les serveurs
- ***restart*** pour les redémarrer
- ***hdfs*** raccourci pour java hdfs.HdfsClient, suivi des mêmes arguments
- ***mmr*** raccourci pour java application.MyMapReduce, suivi des mêmes arguments. Hidoop utilisant RMI, il peut être nécessaire de spécifier l'adresse IP à utiliser si la machine possède plusieurs interfaces (valeur de `java.rmi.server.hostname`). Pour cela, lancer ***mmr -ip <ip_address>*** suivi des arguments.
- ***printconf*** pour afficher le fichier *conf.xml*
- ***deploy*** pour ouvrir un shell comportant les commandes de déploiement sur des machines distantes (ssh).
- ***monitoring*** pour ouvrir un shell comportant des commandes de test de performance de l'application.

Sur chaque serveur, les sorties sont redirigées vers le fichier *<ip_serveur>.log* dans $**HIDOOP_HOME.** Sur le client, *stderr* est redirigée vers le fichier *log*.

Configuration de Hidoop

La configuration se fait via un fichier *conf.xml* placé dans le répertoire $**HIDOOP_HOME/config/**. Un fichier d'exemple est donné ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>
<config metadata="meta">
    <default-chunk-size value="64" unit="MB" />
    <servers>
        <node ip="172.211.22.1"/>
        <node ip="chewie"/>
    </servers>
</config>
```

- ***config***: l'attribut obligatoire *metadata* est le nom du fichier de métadonnées qui sera créé.
    - ***default-chunk-size***: élément optionnel précisant la taille de chunk (entier) à utiliser par défaut dans HDFS. Si absent, cette taille est de 64MB.

      Note : les unités de tailles supportées sont bytes, kB, MB, GB (non sensible à la casse). Par simplicité, une unité inconnue a le même effet que bytes.

    - ***servers*** *:* liste des serveurs. L'attribut *ip* d'un *node* correspond en réalité soit à l'adresse ip de la machine soit à son nom (hostname). Cette liste ne peut pas être vide.

Déploiement et compilation

Les fichiers nécessaires sur la machine cliente sont donc ceux des répertoires **src, scripts** et **config**. Le répertoire où les classes java de l'application sont stockées est $***HIDOOP_HOME***/*src* par défaut, mais il est possible de définir une variable système **HIDOOP_CLASSES** indiquant leur localisation.

Première utilisation sur la machine cliente :

```
$ export HIDOOP_HOME=/path/to/hidoop
$ mkdir -p $HIDOOP_HOME && chmod 700 $HIDOOP_HOME
$ export HIDOOP_CLASSES=/path/to/class/files              # Optionnel
$ mkdir -p $HIDOOP_CLASSES && chmod 700 $HIDOOP_CLASSES   # Optionnel
$ cd $HIDOOP_HOME
$ ./hidoop <ssh_username>
hidoop> deploy
hidoop-deploy> ...
```

Les commandes de déploiement accessibles dans le sous shell `hidoop-deploy` sont les suivantes :

- ***compile*** pour compiler en local
- ***tonode*** pour rendre disponible l'application sur une machine serveur distante (le dossier pointé par ***$HIDOOP_HOME*** doit être existant)

- ***mkhome*** pour créer automatiquement les répertoires ***$HIDOOP_HOME*** sur les serveurs définis dans *conf.xml*
- ***rmhome*** pour supprimer le répertoire ***$HIDOOP_HOME*** sur une ou plusieurs machines
- ***rmnodedata*** pour supprimer les données (dossier data) sur une ou plusieurs machines

Commandes Hidoop

La plateforme Hidoop supporte les opérations suivantes :

- Écriture d'un fichier dans HDFS :

  ```
  hidoop> hdfs -w <nom_fichier_local> options
  ```

  Le fichier écrit dans HDFS aura le nom du fichier local. Son chemin peut être absolu (commençant par '/') ou relatif à $**HIDOOP_HOME/data/**.

  - ➢ `-f ln|kv` : format du fichier (ln par défaut)
  - ➢ `--chunks-size=<taille>` : taille des chunks (ex: 100B, 1MB, 1.2MB...). Si aucune unité (B, kB, MB, GB, TB) n'est fournie, la valeur est en bytes. Si la veleur est négative, cet argument est ignoré.
  - ➢ `--rep=<facteur>` : facteur de réplication (entier positif), non supporté et toujours égal à 1 dans cette version

- Lecture d'un fichier dans HDFS

  ```
  hidoop> hdfs -r <nom_fichier> options
  ```

  - ➢ `<fichierLocal>` : fichier local de destination. Son chemin peut être absolu (commençant par '/') ou relatif à $**HIDOOP_HOME/data/**.
    Le fichier est lu dans '`r_<nom_fichier>`' par défaut.

- Liste des fichiers dans HDFS

  ```
  hidoop> hdfs -l options
  ```

  - ➢ `--detail` : informations détaillées sur les chunks

- Suppression d'un fichier de HDFS

  ```
  hidoop> hdfs -d <nom_fichier>
  ```

- Exécution d'une application en Map/Reduce

  ```
  hidoop> mmr <nom_fichier>
  ```

  Le fichier créé contenant les résultats finaux est `<nom_fichier>-tot`

⚠ *Attention : Les noms de fichiers HDFS sont limités à 80 caractères et ne doivent pas contenir d'espace.*

Supervision et évaluation

Les commandes de supervision accessibles dans le sous shell `hidoop-monitoring` sont les suivantes :

- ***cmpref*** pour comparer un fichier de résultat MapReduce avec sa version séquentielle
- ***evalf*** pour évaluer les performances sur un fichier particulier
- ***logtail*** pour affichers les dernières lignes du fichier log d'un serveur donné
- ***logrm*** pour supprimer le fichier log d'un serveur donné
- ***nodels*** pour lister les fichiers de données d'un serveur