Projet Hidoop

Projet de données réparties - Réponses HDFS

LAPLAGNE Chloé
RAZAFIMANANTSOA Nathan

Ce document présente les corrections et réponses à l'évaluation sur notre première version de HDFS, ainsi que les améliorations apportées.

- **Correction de Bugs** :

    - Contexte : *en cas de coupure de connexion avec un des serveurs, la suppression mettait bien les métadonnées à jour, laissant les fragments du fichier sur le serveur*.

      Ce bug a été corrigé, les métadonnées ne sont modifiées que si la connexion avec tous les serveurs requis a réussi (les pannes de serveurs ne sont pas gérées dans cette version).

- **Pistes d'amélioration** :

    - Proposition : *dans HdfsWrite, remplacer le paramètre 'taille d'un chunk' par 'nombre de chunks'*.

      Ici, lorsqu'un utilisateur écrit un fichier il connaît la taille de ce fichier. En revanche, il ne sait pas forcément combien de serveurs sont en ligne à un moment donné et ce nombre peut être variable.
      Indiquer (ici en octets) la taille des chunks permet de mieux se représenter ce qui sera traité par les opérations Map. C'est également le mode de fonctionnement utilisé par Hadoop et cela permet d'adapter par exemple une taille de chunk par défaut pour un format donné.
      Nous avons donc conservé ce choix mais nous avons essayé de le rendre plus clair dans le message d'aide.
      De plus, nous avons enlevé pour simplifier le mode 'distributed' qui permettait de répartir les chunks sur tous les serveurs. Ce mode était en effet surtout utile au début du développement et n'est plus très pertinent sur un fonctionnement classique.

- **Implantation de la fiabilité** dans les échanges client / serveur :

    - Mise en place de codes d'erreur et de retour codés sur un entier de type long. Ces codes sont échangés entre serveur et client et leur permettent de communiquer :
        - -1 **FILE_NOT_FOUND** : fichier introuvable ou vide sur le serveur
        - -2 **FILE_EMPTY** : fichier vide envoyé sur le serveur (n'est pas enregistré)
        - -3 **FILE_TOO_LARGE** : place disponible sur le serveur insuffisante pour le chunk, ou taille du chunk trop importante par rapport à la moyenne annoncée (comme on coupe par ligne, la limite d'un chunk faisant 2 * chunkSize semble raisonnable: on a peu de chances d'avoir une ligne qui fait la taille d'un chunk...)
        - -4 **INCONSISTENT_FILE_SIZE** : fichier reçu de taille différente de celle annoncée par le serveur
        - -5 **IO_ERROR** : erreur de communication
    - Affichage des codes d'erreur sur les communications ayant échoué entre client et serveur.

- **Divers** :

    - Ajout de la possibilité d'utilisation un chemin absolu pour les fichiers locaux :

      Le nom du fichier source pour un HdfsWrite et du fichier de destination pour un HdfsRead peuvent être soit un chemin absolu (commençant par '/'), soit un nom relatif au dossier data.