Projet

Daniel Hagimont - hagimont@enseeiht.fr
Ehouarn SIMON - ehouarn.simon@enseeiht.fr

Certificat Big Data
Novembre 2018

L'objectif de ce projet est de concevoir une application Big Data et de la déployer dans une infrastructure de calcul. Ce projet utilise les notions vues en cours et mises en pratique en TP. Le projet se compose de deux parties, une relative à la conception d'une application de traitement de données, l'autre à la mise en place d'une infrastructure de calcul.

Conception d'une application Big Data

Le but de cette partie est l'implantation et l'évaluation d'une méthode de descente de gradient pour la résolution d'un problème de filtrage collaboratif. Nous avons à notre disposition un fichier contenant les évaluations d'un certain nombre de films par différents utilisateurs d'une plateforme. Ces utilisateurs n'ayant pas noté, ni même vu, l'ensemble des films à disposition, l'objectif est de pouvoir estimer ces notes « manquantes » depuis l'ensemble de notes, films et utilisateurs à notre disposition. L'objectif est de pouvoir proposer de futures recommandations de films pertinentes pour chacun des utilisateurs.

Ayant stocké ces données sous forme d'un matrice R=[r_{ij}] (ligne → utilisateur, colonne → film, entrée de la matrice → note), ce problème peut se modéliser comme la recherche d'une factorisation de rang faible de R. Ceci conduit au problème d'optimisation suivant :

$$\min_{P,Q} \sum_{(i,j) \text{ t.q. } \exists R_{ij}} \left(r_{ij} - q_j^\top p_i\right)^2 + \lambda(||p_i||^2 + ||q_j||^2)$$

avec (p_i) et (q_j) les lignes de P et Q.

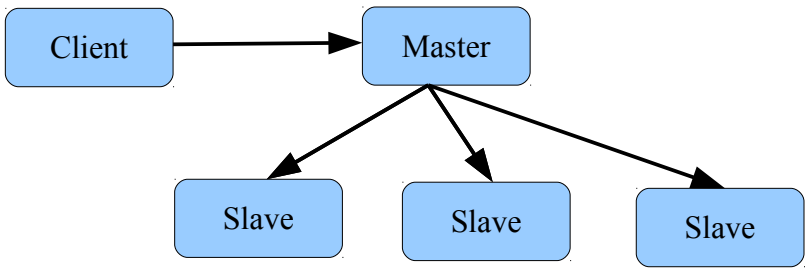
On cherche ainsi le couple de matrices P et Q, de rang fixé, tels que le produit entre P et la transposée de Q minimise l'écart aux données R.

Pour ce faire, vous avez à votre disposition un notebook jupyter à compléter, ainsi que le fichier de données *ratings.dat*. Ce notebook est en python et fait intervenir la librairie pyspark, vous permettant de manipuler le modèle de programmation Spark. Le fichier de données *ratings.dat* doit se trouver dans un répertoire *data/* par défaut. Il vous est demandé de répondre aux différentes questions du notebook, afin de mettre en place la résolution du problème de recommandation dans un environnement Spark.

Une fois la version notebook du TP réalisé, il vous est demandé d'en extraire les sources python (copier/coller dans un fichier par exemple) afin de pouvoir exécuter votre programme de filtrage collaboratif dans l'infrastructure de calcul, que vous mettrez en place dans la seconde partie du projet.

Mise en place d'une infrastructure de calcul

Le but de cette partie est de concevoir une infrastructure Spark de calcul distribuée, simulée sur une ordinateur personnel à l'aide de containers Docker. Un container est utilisé pour exécuter le Master de Spark et un ensemble de containers exécutent les Slaves de Spark. Cette infrastructure est représentée sur le schéma ci-dessous.



Les Slaves sont enregistrés auprès du Master. Un client (un shell) peut soumettre une application Spark pour exécution auprès du Master (avec spark-submit) et le Master distribue cette exécution auprès des Slaves.
Comme cette infrastructure s'exécute en local, vous avez deux possibilités du point de vue du stockage :

- soit les containers partagent tous le même espace disque. Il suffit de partager un répertoire entre ces containers et votre ordinateur, et les fichiers source et destination des traitements seront gérés dans ce répertoire.
- soit on démarre HDFS dans tous les containers, le Master exécutant le NameNode et les Slave exécutant des DataNode. Une application Spark fait alors référence à un fichier avec une URL HDFS. Cette solution est plus proche de la réalité.

Vous pouvez commencer avec la première option et aborder la seconde si vous en avez le temps.

Pour construire cette infrastructure, vous devez donc implanter un script qui va créer les images des containers Docker (ces images contiennent toutes les installations nécessaires à l'exécution de Spark et de votre application), démarrer les instances de container de l'infrastructure (par exemple un Master et trois Slaves) et lancer l'exécution de votre application. L'application utilisée est celle de la section précédente.

Pour réaliser ce projet, vous devrez vous former (ou perfectionner) aux techniques suivantes :

- Docker
  - Vous pouvez trouver de nombreux tutoriaux Docker sur Internet
  - Vous trouverez sur la page ci-dessous un support de cours et un TP permettant de vous entraîner
    http://hagimont.perso.enseeiht.fr/resources-N7/cloud/3IN-cloud.html
- Spark
  - Vous avez fait un TP sur Spark, mais il vous faudra vous former à l'utilisation de Spark en mode distribué.

Condition de travail

- Le projet est à faire en binôme. Vous utilisez vos ordinateurs personnels.

- Vous devez notifier par email la composition de votre binôme avant le 21 novembre 2018 midi.
- Vos réalisations (code source incluant le notebook jupyter, script et un rapport) devront être envoyées par email avant le 21 décembre midi.
- Les emails sont à envoyer aux deux auteurs de ce sujet.
- Le suivi du projet se fera via le forum créé pour le Certificat.