TP Spark Streaming

Jonathan Lejeune

Prérequis

Vous devez avoir installé la plate-forme Spark et avoir configuré Eclipse correctement pour coder et compiler vos programmes scala avec l'API de Spark.

NB : Dans l'ensemble des exercices de ce TP, pour éviter d'afficher l'énorme quantité de LOG produits par spark, ajoutez dans vos programmes la ligne suivante

```
Logger.getLogger("org.apache.spark").setLevel(Level.OFF)
```

Attention de bien importer le `Logger` et le `Level` du package `org.apache.log4j`

Exercice 1 – Calcul de Pi

En vous inspirant du programme de SparkPi qui permet de calculer une valeur approchante de Pi, nous allons coder un programme streaming qui à partir d'un flux de points, calcule en sortie la valeur de Pi. Cette valeur devra normalement converger vers Pi au fur et à mesure de l'avancement du Job. La source sera une socket de type Text.

Question 1

Le programme source est un programme scala qui envoie indéfiniment les coordonnées de points compris dans l'ensemble $[-1.0, 1.0[^2$. Le code de ce programme est le suivant :

```
import scala.math.random;
object Random extends App{
        while(true) println((random*2-1)+" "+(random*2-1));
}
```

Compilez ce programme avec la commande scalac.

Question 2

Tapez la commande :

```
scala Random | nc -l -p 4242
```

La commande `nc` ouvre une socket d'écoute sur la machine local sur le port 4242. Pendant le codage de votre programme il vous est conseillé d'arrêter cette commande pour éviter de consommer inutilement des ressources

Question 3

Dans le package `datacloud.spark.streaming.pi`, codez un programme streaming `PiAtTime` qui :

- prend comme source une socket de type texte
- convertit les String reçu en Double
- teste si le point est dans le cercle ou pas
- affiche le ratio $4 * NbIn/NbTotal$

Dans le cadre de cette question, nous souhaitons calculer uniquement ce ratio sur l'instant t.

Question 4

Dans le même package, codez un programme streaming `TowardsPi` qui s'inspire du programme de la question précédente mais qui cette fois calcule le ratio en prenant en compte les valeurs calculées dans le passé du stream. Pour cela vous utiliserez la méthode `updateStateByKey`. N'oubliez donc pas d'activer le checkpointing.

Exercice 2 – Top ten de Twit

Un twit est un message posté sur le réseau social Twitter. Concrètement un twit est un ensemble de mots séparé par des espaces. Certains mots commencent par un *"#"*. Ils sont désignés sous le nom de *hashtag* et permettent de marquer certains mots du message. Une application courante du streaming est de connaître en temps-réel la liste des hashtags les plus référencés dans le flux de twits postés sur Twitter.

Dans cet exercice vous utiliserez le fichier *generateTwit.sh* fourni dans les ressources globales de l'UE. Ce programme émule de manière probabiliste la production de twits depuis Twitter. Le programme choisit aléatoirement N (par défaut 10) hashtags qui seront produits en priorité. En faisant varier la probabilité de production de N hashtags, il est possible d'avoir une estimation d'un classement. Une fois que le programme est lancé, il est possible de changer ce classement potentiel à chaque fois que vous appuierez sur la touche "entrée". Ainsi vous pouvez simuler le changement des tendances des hashtags en temps réel. Par la suite vous redirigerez la sortie standard de ce programme vers la commande `nc` en suivant le même principe que l'exercice précédent :

```
generateTwit.sh | nc -l -p 4242
```

Question 1

Dans un package `datacloud.spark.streaming.twit`, codez un programme spark streaming `TopTwitAtTime` pour connaître le classement des 10 hashtags les plus utilisés à l'instant courant. Vous prendrez 1 seconde comme période de base.

Question 2

Dans le même package, toujours, en gardant la même période de base et en vous inspirant du programme précédent, coder le programme `TopTwitAtTime` afin d'afficher toutes les x secondes un classement des 10 hashtags les plus utilisés en prenant en compte les y dernières secondes.

Question 3

Tester plusieurs fois votre programme en faisant varier y. Pendant chaque test, changer le classement attendu et remarquer le temps nécessaire pour que le nouveau classement prenne place sur l'ancien dans l'affichage du programme de spark streaming. Que remarquez-vous ?