Ouassaidi Sami

Liu Xiaoyi

2015/2016

Rapport Projet Hadoop

Ouassaidi Sami
Liu Xiaoyi

2015/2016

Introduction

Dans le cadre de notre Projet Hadoop nous avons choisis d'implémenter l'algorithme du pagerank en utilisant mapreduce, en effet il nous a semblé être le sujet ideal afin de tester la puissance de la parallelisation avec hadoop, de plus le lien évident avec google entre le principe du mapreduce et l'algorithme y a également contribué.

Fonctionnement

Le programme se découpe en trois jobs, le premier établit la liste de toutes les URLs présentes dans le fichier warc et le second la liste des pages et les liens href vers d'autres pages.
Ces deux premiers jobs sont très similaires et servent à la construction de la matrice d'adjacence utilisée plus tard pour le calcul du pagerank.
La matrice est ensuite construite directement dans un tableau à deux dimensions de taille nxn (n : nombre total d'url trouvées dans le fichier), cette matrice représente un lien pointant de la page i vers la page j sachant que le lien (i,i) est mis à 1 pour simuler le rafraîchissement d'une page la valeur 0 indique qu'aucun lien n'existe entre les deux pages, le premier fichier d'output est utilisé pour numéroter toutes ces pages le second permet donc de savoir si l'on doit mettre à 1 la valeur de la case ou non.
Le dernier job quand à lui permet de calculer le pagerank de toutes les pages par mapreduce, le fichier ainsi obtenu contient la liste des urls(indices dans la matrice) le pagerank associé ainsi que les pages vers lesquelles elles pointent.

Problèmes rencontrés

Le problème majeur était venu du fait de créer la matrice d'adjacence, après des tentatives infructueuses nous n'avons pas trouvé d'autres solutions que de passer par un tableau intermédiaire et de modifier les paramètres de nos fonctions map et reduce, le reste du programme a été assez facile à coder au vu des nombreuses ressources disponibles en ligne .
Un autre point sur lequel nous avions bloqué était l'utilisation du makefile pour l'automatisation des exécutions, en effet dans un soucis de lisibilité et d'organisation nous avons préféré en faire usage et de structurer nos dossier en bin et src, nous avons mis du temps à nous rendre compte que GNU make n'intègre pas l'export de variables d'environnement pour le classpath et que nous devions pour cela passer par le teminal à nouveau, nous avons au final ajouté une ligne dans la bashrc afin que java puisse trouver les classes directement dans le répertoire bin.

Ouassaidi Sami
Liu Xiaoyi

2015/2016

Améliorations possibles

Un des points essentiels à améliorer serait donc de ne plus passer par un tableau intermédiaire mais que la matrice d'adjacence soit construite directement au deuxième job, ceci permettrait également de ne pas dépasser le max autorisé pour le heap java, en effet pour un gros fichier warc c'est ce qui arrive et le programme s'arrête à ce niveau là, le fichier utilisé est juste un extrait et prouve uniquement le bon fonctionnement de l'algorithme de pagerank pour des fichiers ne générant pas un tableau dont la taille totale ne dépasse pas 2Go. C'est en ecrivant ces lignes que l'on s'est rendu compte que Hadoop autorise de multiples fichiers en input comme en output pour un même map, ceci facilite grandement la chose et la matrice pourrait être créée directement au 2ème job en prenant comme fichier d'input le fichier warc ainsi que le fichier contenant la liste des toutes les URLs obtenu par le premier job. En allant plus loin les deux jobs pourraient être confondus en un seul ce qui diminuerais drastiquement le temps d'exécution. À l'heure actuelle les deux premiers jobs terminent leur exécution au bout d'une vingtaine de minutes pour un fichier warc assez large (4 .5 go) on pourrait imaginer que cela prendrait une douzaine si cela était achevé.
Autre point à améliorer serait l'algorithme de pagerank en introduisant des heuristiques pour que le pagerank soit plus précis, nous avons içi uniquement mis en place une boucle sur 10 itération pour que le calcul du pagerank.

Utilisation

Pour exécuter le programme il suffit lancer le make depuis le terminal en veillant bien à indiquer les variables d'environnement :

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64/
export HADOOP_HOME=/usr/local/hadoop-2.7.1
export PATH=$HADOOP_HOME/bin:$PATH
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar:${PWD}/bin
```

puis :
make all
make run
make clean

Des commentaires (en anglais) se trouvent au niveau des sources et du makefile pour davantage de précisions.