Rapport d’avancement du Projet Hidoop

PARTIE HIDOOP

Sabras Sherwin

Bourguignon Luc

1) Améliorations par rapport à la version précédente

> Mise en place de connexions ssh avec différentes machines de l'N7 pour pouvoir faire les tests ;

> Rédaction des scripts de déploiement et de scripts d'arrêt des démons Hidoop ;

> Ajout d'un sémaphore pour qu'il n'y ait pas plus d'un nombre de map donné (5 dans notre cas) en cours en parallèle ;

> Tests d'intégration en cours avec Hdfs, mais problèmes avec le lancement de Hdfs pour l'instant (en cours de résolution) ;

2) Réflexions par rapport au nouveau Hidoop

> Dans un premier temps, des **tests simples sur des petits fichiers** : intégrer hdfs et découper un fichier en plusieurs morceaux, vérifier que résultat est bon. Nous avons pour l’instant un problème sur la partie HDFS qui nous empêche de créer plus de 4 sockets (nombre de connexions ssh lors du déploiement de Hdfs via un script Bash) ;

> Tester la **persistance du name node** (hashmap enregistrée sur le client), en redémarrant Hidoop suite à un appel à HdfsWrite ;

> **Performance et mise à l'échelle** : découper le fichier en beaucoup de fragments et / ou les répartir sur beaucoup de machines ; comparer les résultats en exécutant le même traitement en local et sur moins de machines. Cela nous permettra de voir si cela prend plus (ou moins) de temps (à cause des latences dûes au réseau, de l'overhead), et nous vérifierons que nous ne perdons pas de fragments ;

2) À faire pour les prochaines itérations

> S’assurer que la gestion plus fine du lancement des maps (1 traitement à la fois sur 1 machine) est effective. Dans le cas où le résultat n’est pas celui attendu, utiliser des primitives wait et notify afin prévenir Job à chaque fois qu'un traitement est terminé (moniteur) ;

> Réaliser une étude de scalabilité (faire varier : nombre de fragments, nombre de nœuds, taille des fragments, nombre de maps en simultané sur une machine) ;

> Ecrire du code pour tracer des courbes de performance (des tests) automatiquement en Python à l’aide de fichiers textes, pour l’étude de scalabilité en faisant varier divers paramètres ;

> Modifier le code Java afin de pouvoir tester d’autres applications (autres que WordCount) ;