Le génie pour l'industrie

RAPPORT TECHNIQUE
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
DANS LE CADRE DU COURS GTI792 PROJET DE FIN D'ÉTUDES EN GÉNIE DES TI

BASE DE DONNÉES DISTRIBUÉE APPLIQUÉE À LA GÉNÉTIQUE DANS LE CADRE DE L'ANALYSE DU SÉQUENÇAGE GÉNOMIQUE
RAPPORT D'ÉQUIPE

JEAN-FRANÇOIS HAMELIN
HAMJ12068802
ET
JEAN-PHILIPPE BOND
BONJ06048709

DÉPARTEMENT DE GÉNIE LOGICIEL ET DES TI

Professeur-superviseur

Alain April

MONTRÉAL, 9 AOÛT 2012
ÉTÉ 2012

REMERCIEMENTS

Alain April : Professeur de génie logiciel.

Patrice Dion : Analyste des systèmes et réseaux informatiques, département de systèmes éducationnels et de recherche de l'ÉTS.

Anna Klos : Diplômée de l'ÉTS en génie logiciel.

Ousmane Diallo, B.Sc. : programmeur pour le projet S2D, laboratoire Guy Rouleau, CRCHUM.

OPTIMISATION DE RECHERCHE GRÂCE À HBASE SOUS HADOOP
RAPPORT D'ÉQUIPE

JEAN-FRANÇOIS HAMELIN
HAMJ12068802

JEAN-PHILIPPE BOND
BONJ06048709

RÉSUMÉ

Ce projet s'insère dans un contexte d'affaires où le Centre de Recherche du Centre Hospitalier de l'Université de Montréal (CRCHUM) est aux prises avec des problèmes avec un système d'identification de gènes et où l'ÉTS est désireuse d'amasser du matériel en vue d'un cours sur le « big data ».

Le CRCHUM, possédant bien au-delà de 150 millions d'enregistrements de données génomiques, utilise à l'heure actuelle un système permettant d'effectuer des recherches sur des gênes afin de, par exemple, trouver certaines variantes de gênes partageant des similarités. Selon les informations publiées sur le site Web du laboratoire Roueleau, « l'objectif principal du projet Synapse to Disease (de la synapse à la maladie ou S2D) est d'identifier des gènes causant ou prédisposant à des maladies du développement et du fonctionnement neuronal » (http://www.laboguyrouleau.ca/S2D.html). S'appuyant sur une base de données relationnelle conventionnelle, le CRCHUM voit rapidement sa solution atteindre un plateau. En effet, plusieurs de leurs requêtes sont longues à effectuer et ont déjà demandé un remaniement de la base de données important. Leurs responsables voient donc, à l'horizon, un problème dans leur capacité de stocker et effectuer des requêtes efficaces sur les données.

De son côté, le professeur Alain April réfléchit depuis un bon moment à créer de toute pièce un cours portant sur le nouveau phénomène du « big data ». Cependant, soucieux deLorsqu'il rencontra le Dr Rouleau, en charge du laboratoire Rouleau, lors d'une conférence, le professeur April vu immédiatement un potentiel dans le problème technologique de ce dernier.

En conséquence de ce qui précède, le projet aura des retombées sur les deux parties prenantes distinctes. D'un côté, le CRCHUM bénéficiera d'une preuve de concept qui peut servir comme prototype de solution à leurs problèmes de performance lors de requêtes sur plusieurs millions d'enregistrements. De l'autre côté, le professeur Alain April et son entourage pourront se servir des extrants du projet comme ressources pour monter un futur cours portant sur la technologie "big data".

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	Application Programming Interface
ADN	Acide désoxyribonucléique
CRCHUM	Centre de Recherche du CHUM
CPU	Central Processing Unit
GIG	GigaByte
HDFS	Hadoop Distributed File system
JPS	Java Process Status tool
MB	MegaByte
NoSQL	Not only SQL
RAM	Random Access Memory
SGBD	Système de gestion de base de données
SSH	Secure SHell

LISTE DES SYMBOLES ET UNITÉS DE MESURE

INTRODUCTION

Comme le quotidien La Presse le soulignait dans son article du 22 mai 2012, "l'heure est au big data" (McKenna, 2012). En effet, un nouveau phénomène dû en partie à la popularité d'Internet a pour conséquence de produire des quantités titanesques de données chaque jour. Cependant, le phénomène est bien plus large que la consommation de données sur l'Internet. En effet, les domaines comme la génomique, l'épidémiologie et ou la sécurité nationale produisent et consomment énormément de données. Bien que certaines entreprises comme Google et Facebook puissent engranger des profits monstrueux avec ces données, encore doivent-elles bien les gérer et les supporter; bienvenue à l'ère du "big data". Comme le décrit Wikipédia, « Big data (« grosse donnée » ou données massives) est une expression anglophone utilisée pour désigner des ensembles de données qui deviennent tellement volumineux qu'ils en deviennent difficiles à travailler avec des outils classiques de gestion de base de données. Dans ces nouveaux ordres de grandeur, la capture, le stockage, la recherche, le partage, l'analyse et la visualisation des données doivent être redéfinis » (http://fr.wikipedia.org/wiki/Big_data).

Le besoin en stockage de données à large échelle et l'engouement entourant les technologies émergentes comblant ce besoin sont tellement récents qu'aucune école spécialisée à Montréal, y compris l'ÉTS, ne dispense des cours sur la matière. Dans une toile de fond où l'ÉTS se veut chef de file en matière d'école de technologie, il est impératif de bâtir un cours ayant pour thématique le "big data". C'est donc dans cet esprit que ce projet veut accomplir la création d'un environnement HBase distribué, bien documenté et s'appuyant sur un cadre d'utilisation concret. En effet, afin de rester le plus près possible d'une utilisation industrielle de la base de données HBase, un échantillon de données portant sur la génomique humaine, du Centre Hospitalier de l'Université de Montréal (CRCHUM), sera utilisé.

Faisant suite à un projet de fin d'études antérieurement réalisé par Mme Anna Klos, ce projet ira au-delà de ce que Mme Klos a accompli en utilisant plusieurs renseignements de

son rapport. Sommairement, le but ultime du projet est de fournir du matériel de calibre professionnel, qu'il s'agisse de la preuve de concept ou de la documentation y étant rattachée, pouvant être utilisé dans le cadre d'un cours à venir traitant du phénomène "big data". L'environnement distribué créé doit donc pouvoir être reproduit sans heurts à la fin du projet grâce aux instructions fournies.

CHAPITRE 1

ACCOMPLISSEMENTS DE L'ÉQUIPE

1.1 Compréhension des requêtes problématiques

Au tout début du projet, notre équipe reprenait les travaux ainsi que le rapport d'Anna Klos, une étudiante de l'ÉTS qui avait fait son projet de fin d'études avec le professeur Alain April. La première étape du projet fut de prendre possession des artéfacts produits par Mme Klos et de comprendre la problématique technique du laboratoire Guy Rouleau du CRCHUM. Ainsi, nous avons rencontré Ousmane Diallo au laboratoire Guy Rouleau afin de bien saisir les besoins d'affaires ainsi que de voir le système utilisé actuellement en action. Nous avons par la suite compris que les trois requêtes effectuées actuellement par le logiciel peuvent en fait se résumer à une seule. En effet, les deux premières requêtes effectuées ont pour seul but d'inclure ou d'exclure certaines données de la requête finale. Avec ceci en tête, nous avons trouvé une stratégie de recherche sur HBase qui permet de reproduire le même comportement que le système fonctionnant avec une base de données relationnelle conventionnelle. La solution en est une hybride, c'est-à-dire qu'elle utilise HBase pour aller chercher les informations à afficher à l'utilisateur selon plusieurs critères d'inclusion et utilise PostgreSQL pour trouver les identifiants de variants à exclure de la requête en fonction des échantillons génomiques exclus de la recherche qui ont été sélectionnés par l'utilisateur dans l'interface graphique de l'application.

1.2 Création d'un environnement Hadoop/HBase pseudo-distribué sous Mac OS X Lion et Ubuntu (Linux)

Afin de pouvoir travailler à distance sans devoir continuellement être connecté au VPN et subir les délais d'attente du réseau, notre équipe s'est installée et configuré deux postes de développement sur les machines des développeurs. Ces postes fonctionnent sous des systèmes d'exploitation différents, soit Mac OS X Lion 10.7.2 et Ubuntu (Linux) 12.04 LTS. Sur chacune de ces machines, Hadoop et HBase ont été installés en mode pseudo-

distribué. En mode pseudo-distribué, tous les processus (« daemons ») s'exécutent sur un seul nœud alors qu'en mode pleinement distribué, ils sont répartis sur plusieurs machines physiques de la grappe. Cependant, le mode pseudo-distribué tire profit du système de fichiers distribué HDFS, comme le mode pleinement distribué. L'objectif de ces environnements de développement est de reproduire en tout point la grappe de serveurs mise en place à l'ÉTS, mais à plus petite échelle. Ceci a pour résultat que les configurations et le code produit sur les postes de développement sont directement exportables sur le serveur.

1.3 Création d'un environnement Hadoop/HBase pleinement distribué sous Ubuntu Server 12.04 LTS (Linux) composé de trois serveurs

Au début du projet, l'ÉTS a mis à notre disposition trois machines virtuelles puissantes afin de mettre en place une grappe Hadoop pleinement distribuée. La grappe est composée d'un serveur maître et de deux serveurs de régions (« region server »). L'environnement est fonctionnel et a été utilisé pour comparer les performances entre un environnement pseudo-distribué et un environnement pleinement distribué. Cependant, la grappe ne fut pas utilisée lors de la démonstration ayant eu lieu pendant la présentation orale puisque la latence entre le serveur Web installé sur l'ordinateur portable et la grappe située à l'ÉTS a pour conséquence de ralentir l'arrivée des résultats de recherche. Pour avoir un environnement optimal de démonstration, il aurait fallu installer le serveur Web au même endroit que la grappe, afin que les requêtes entre la couche applicative (Tomcat) et la couche de données (HBase) ne subissent pas de multiples délais. C'est donc dire que l'environnement pleinement distribué n'a pas été utilisé à sa juste valeur, mais nous ne voulions pas donner l'impression que nos requêtes étaient lentes simplement à cause de la latence causée par le réseau.

1.4 Mise en place d'un schéma HBase sur mesure pour la requête

HBase est une base de données qui requiert de repenser complètement l'organisation des données dans un schéma donné. En effet, HBase ne supporte pas le langage SQL, ni les fonctionnalités comme les JOIN. Les associations un à un, un à plusieurs et plusieurs à plusieurs doivent donc être converties vers une approche « dénormalisée ». Comme Lars

George le mentionne, « le support pour une conception soutenant de grandes tables clairsemées et orientées colonne élimine souvent la nécessité de normaliser les données et, dans le processus, les coûteuses opérations de jointure nécessaires pour agréger les données au moment de la requête. » (George, 2011) Ainsi, la table a été taillée sur mesure pour les besoins du projet de fin d'études avec, pour colonnes, toutes les informations à retourner dans la grille de résultats et avec, pour clé de rangée, une concaténation de tous les paramètres possibles de sélectionner dans le formulaire de l'application. La clé intelligente permet d'effectuer des recherches performantes à l'aide de filtres à base d'expressions régulières.

1.5 Migration des données vers HBase

Les données fournies par l'ÉTS au début du projet étaient stockées dans une base de données PostgreSQL. Les données ont été fournies sous forme de fichiers SQL « dump ». Il s'agissait de plusieurs fichiers qui créaient le schéma, les tables et qui ensuite effectuaient tous les INSERT nécessaires. Afin de copier ces millions de données vers HBase, l'utilitaire Sqoop (SQL to Hadoop) développé par The Apache Software Foundation a été utilisé. Comme le décrit Apache, Sqoop « est un outil conçu pour transférer efficacement les données en vrac entre Apache Hadoop et des bases de données plus conventionnelles telles que les bases de données relationnelles » (The Apache Software Foundation, 2012). Ainsi, les 8 millions de données ont pu être efficacement et (presque) automatique transférées entre PostgreSQL et HBase. Notre équipe a créé une vue SQL qui modélisait la structure de la table HBase désirée et qui construisait aussi la clé selon le format désiré dans HBase. Sqoop effectua une requête « SELECT * FROM » sur la vue et transféra le tout à HBase en utilisant les paramètres de configuration entrés lors de la commande en ligne de console.

1.6 Transposition des requêtes SQL sur HBase à l'aide de chaînes de filtres à expressions régulières

Sachant que la requête du CRCHUM exprimée ci-haut sert essentiellement à inclure et exclure des variants, que ce soit directement ou via des échantillons, voici la stratégie de

requête utilisée avec HBase comme base de données. Tout d'abord, qu'il soit d'ore et déjà mentionné que plus d'informations à ce propos se retrouveront dans le rapport de Jean-Philippe Bond, puisque sa manière de procéder fut retenue parmi toutes celles explorées.

Lors de l'inclusion de types de variants, d'échantillons, de « pipelines » de dépistage ou de variants précis, l'application créer un objet Scan et parcourra la table de la base de données créée pour l'occasion à la recherche d'enregistrements qui, dans leur clé de rangée, contient toutes les informations demandées. En guise de rappel, voici la composition de la clé de rangée de chaque enregistrement de la table HBase :

```
# Format de la clé de chaque rangée
variant_type_id | sample_id | pipeline_id | variant_id
```

L'application confirme qu'un enregistrement de données est conforme à la recherche effectuée en créant un objet Filter pour chaque condition (échantillon, « pipeline » et/ou variant) ajoutée à la requête. C'est objets Filter créerons un modèle d'expression régulière chacun qui sera comparé à chaque enregistrement de la table. Si un enregistrement parvient à traverser tous les filtres avec succès, alors ce dernier est considéré comme un résultat valide et est ajouté à la liste des résultats à retourner au client. Par exemple, considérons le scénario suivant :

> L'utilisateur désire avoir tous les variants dépistés par le « pipeline » de dépistage 24, qui font partie des échantillons 2001 à 3000 et qui ne font pas partie des échantillons 1001 à 2000.

Cet exemple créerait un filtre avec une expression régulière qui validerait le « pipeline » 24 dans le troisième tronçon de clé. De plus, mille filtres à expression régulière devraient être créés pour l'inclusion des échantillons 2001 à 3000 et mille autres pour l'exclusion des échantillons 1001 à 2000.

Cependant, cet exemple ne s'arrête pas simplement ici. Tel que spécifié lors de l'explication des requêtes effectuées sur le système actuel du laboratoire Rouleau, les échantillons exclus doivent faire en sorte que tous leurs variants soient aussi exclus de la recherche, même dans le cas où ces variants sont présents dans d'autres échantillons. Dans la solution développée, ceci se concrétise en une approche hybride. En effet, comme les relations échantillon-variant sont documentées de façon concise dans la base de données PostgreSQL (table « ngs_sample_variant »), mais pas dans la table HBase, une requête est envoyée à la base de données PostgreSQL pour aller chercher tous les identifiants de variants à exclure en fonction des identifiants d'échantillons sélectionnés. Une fois que la réponse de PostgreSQL est disponible, tous les identifiants de variants reçus formeront chacun un nouveau filtre à expression régulière auquel seront confrontés les enregistrements. Ce mécanisme permet d'exclure des résultats tous les variants d'un ou plusieurs échantillons exclus dans le formulaire, sans pour autant transporter le schéma SQL en entier vers HBase.

Enfin, pour améliorer les performances, la table n'est pas parcourue séquentiellement, mais pas intervalle. Effectivement, considérons l'exemple suivant :

> L'utilisateur sélectionne deux échantillons à inclure, par exemple 001056 et 406751. Ces échantillons produiront des clés comme suit (les autres sections de la clé sont purement inventées) :
>
> - `000004 | 001056 | 000024 | 000000000000050`
> - `000004 | 406751 | 000024 | 000000000000050`

S'il fallait parcourir la table séquentiellement, en gardant toujours en tête que HBase ordonne les rangées de manière alphanumérique en se basant sur la clé, il est facile de constater que ces deux clés sont à des milliers d'enregistrements l'une de l'autre. Pour éviter de balayer inutilement des dizaines de milliers de rangées, la recherche se fait par intervalle. Dès que les valeurs de deux échantillons (« samples ») sont supérieures à une limite, par exemple 100, l'application instanciera deux objets Scan qui effectueront chacun une recherche ultrarapide, au lieu d'effectuer une seule recherche qui serait beaucoup plus longue. Bref, l'exemple

mentionné ci-haut produirait deux balayages très courts de la table, puisque les index de début et de fin de la recherche (« startRow » et « stopRow ») sont ajustés chacun à leur balayage respectif.

1.7 Implémentation d'une application Web permettant de tester les requêtes et de modifier

Afin de démontrer la solution développée lors de la présentation orale, notre équipe a développé une application Web qui tente tant bien que mal d'imiter le formulaire de l'application S2D utilisée au laboratoire Guy Rouleau. L'application permet d'inclure des types de variant, des échantillons, des « pipelines » de séquençage, mais aussi d'exclure des échantillons. L'exclusion des échantillons a pour conséquence d'exclure tous les variants étant associés à ces échantillons de la recherche. L'application permet aussi de modifier en temps réel les paramètres de HBase tels que la grosseur du « batch », la grosseur de la cache ainsi que les valeurs des intervalles utilisés lors du balayage des données. Voici quelques captures d'écran de l'application Web développée :

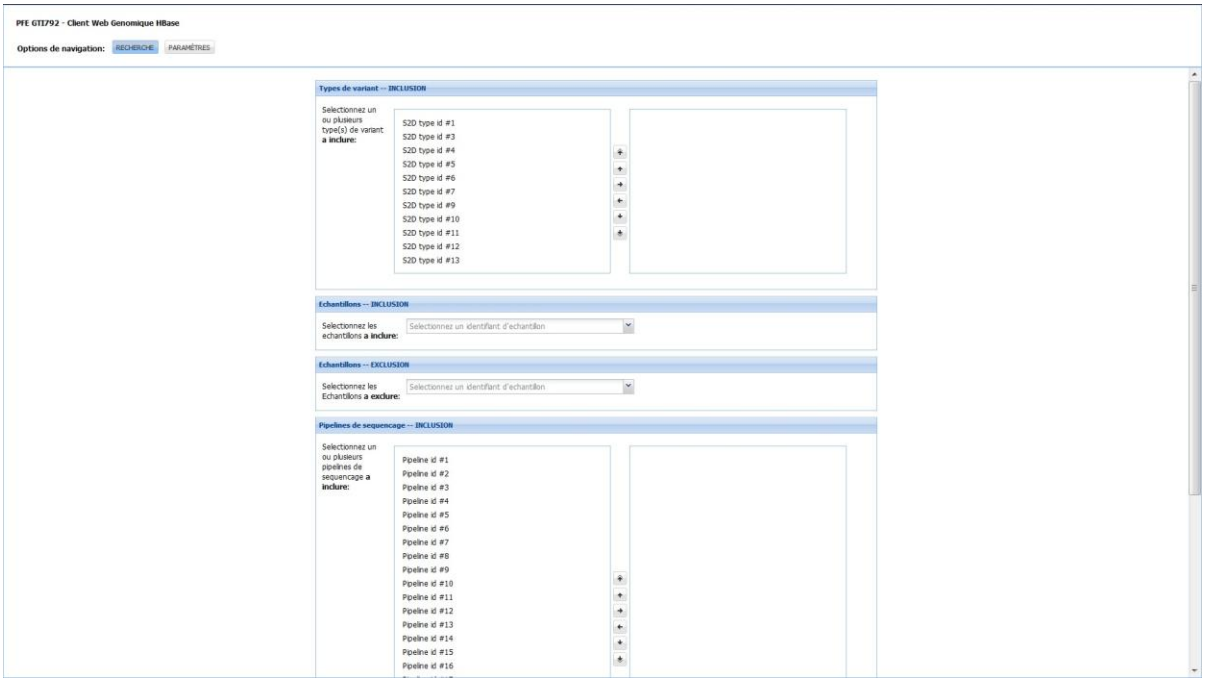

Figure 1: Formulaire de recherche

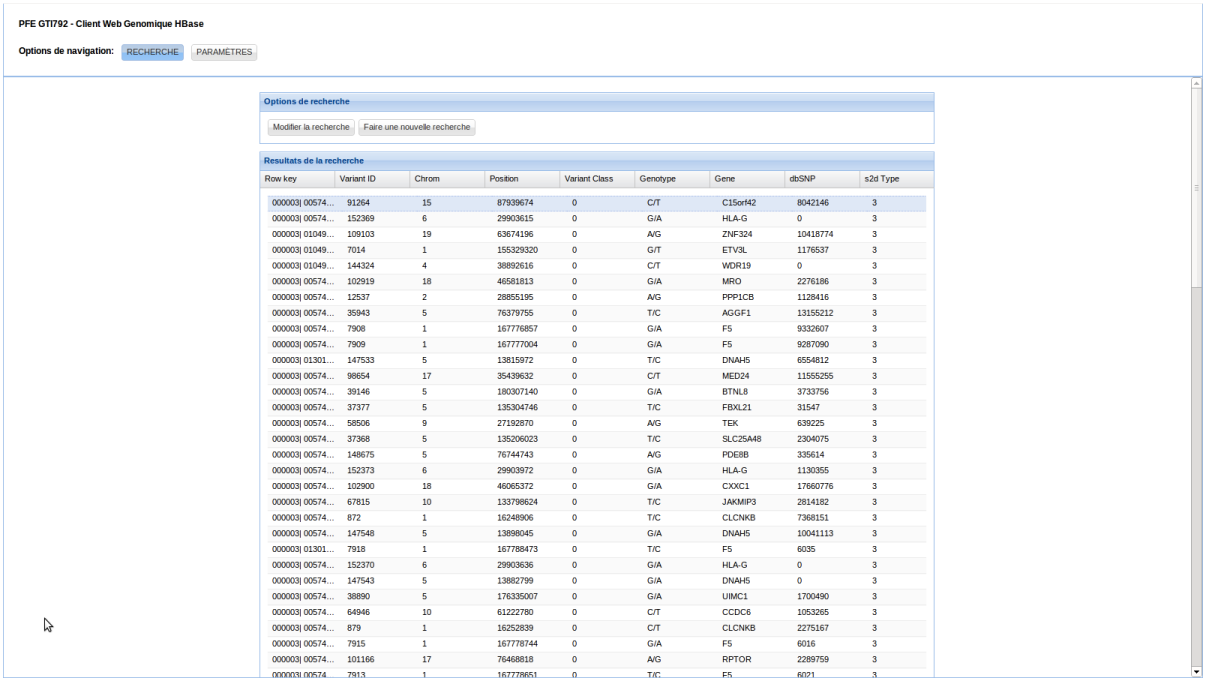

Figure 2: Page de résultats de la recherche

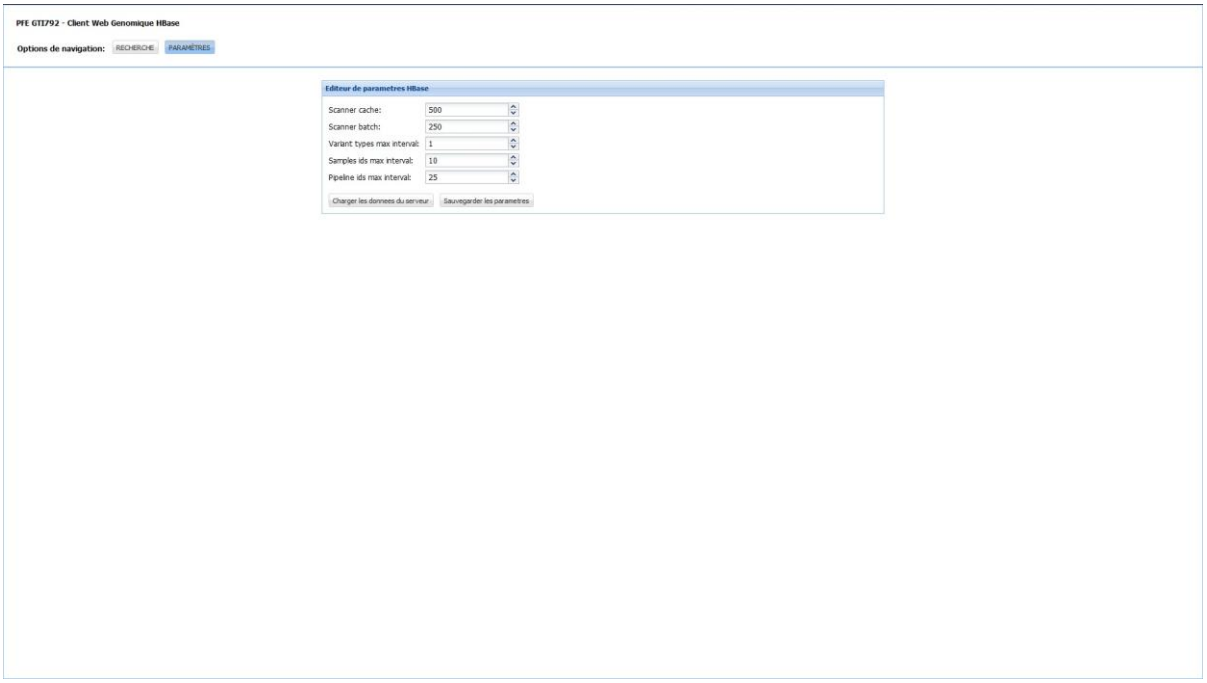

Figure 3: Formulaire de changements de paramètres HBase

CHAPITRE 2

FONCTIONNEMENT DE L'ÉQUIPE

Pour atteindre les objectifs de projet, une approche itérative a été utilisée, chacune des tâches a été exécutée en respectant les étapes du plan soient l'analyse du problème, la conception, le développement, les tests et la documentation. Pour s'assurer d'avoir une documentation complète, celle-ci était entamée lors du commencement d'une tâche et poursuivie jusqu'à ce qu'elle soit terminée. Il était impératif que la documentions soit soignée puisqu'elle sera utilisée pour les travaux du professeur Alain April. L'analyse des problèmes ayant un impact significatif sur le projet comme, par exemple, la définition du schéma HBase et les stratégies de requête ont été faites en équipe, chacun des membres a pu partager ses connaissances pour favoriser le développement d'une meilleure solution. Par ailleurs, pour maximiser les efforts, certaines tâches ont été réalisées en parallèle. Par exemple, lors du développement de l'application permettant de faire une démonstration de notre solution, l'un des membres de l'équipe a réalisé la partie cliente de l'application tandis que l'autre travaillait sur la partie serveur. Lorsqu'une partie du travail était réalisé en parallèle, une rencontre était organisée pour faciliter l'intégration du travail de chacun des membres.

Des périodes de travail d'une durée déterminé étaient prévues chaque semaine, les membres de l'équipe ont toujours respecté les plages établies, ce qui a permis d'atteindre tous les objectifs fixés lors du début du projet. Enfin, tout c'est bien déroulé par rapport au travail d'équipe, chacun des membres à participer de façon active au projet et à livrer le travail qu'il lui avait été attribué.

CONCLUSION

En conclusion, ce projet a réellement apporté son lot de défis et de difficultés. Partant d'un travail effectué par une ancienne étudiante de l'ÉTS, notre équipe a dû composer avec plusieurs embuches. Tout d'abord, les données qui nous ont été données en début de projet ne représentent qu'une partie des données. Il a été difficile pour nous de bien comprendre les liens entre toutes les données, puisque celles-ci n'étaient pas toutes présentes dans la base de données. Ensuite, la génomique étant un domaine scientifique qui utilise des termes techniques extrêmement compliqués, cet aspect a aussi contribué à freiner notre compréhension du problème initial. Par ailleurs, les technologies utilisées pour implémenter la solution étant émergentes et peu matures, plusieurs problèmes ont fait surface, notamment dans l'installation de la grappe et de l'installation en mode pseudo-distribué sur Mac OS X. En outre, trouver la meilleure stratégie de requête possible nous aura demandé beaucoup de tentatives avec diverses combinaisons d'approches et de paramètres. Somme toute, ce projet est, à nos yeux, une brillante réussite puisque tous les objectifs fixés dans le rapport d'étape et la proposition de projet ont été atteints et nous avons été en mesure de livrer un prototype fonctionnel qui permet l'exclusion d'échantillons génomiques. Bien entendu, les améliorations possibles sont infinies, tel que mentionné dans la section des recommandations, mais le travail accompli dans ce projet de fin d'études représente un bon pas en avant face à ce qui a été fait auparavant. Quoi qu'il en soit, ce prototype n'est nullement prêt à remplacer le système S2D du laboratoire Rouleau. Il représente par contre, selon nous, un très bon travail pratique à donner aux étudiants d'un futur cours portant sur le « big data ». Pour terminer, le « big data » est définitivement la voie vers laquelle les chercheurs en génétique s'orienteront dans un futur rapproché. Il n'en tient qu'à l'ÉTS à se tailler une place dans ce marché qui ne demande qu'à être conquis.

RECOMMANDATIONS

Tout d'abord, il faudrait utiliser la stratégie de Scan développée par notre équipe dans un contexte MapReduce optimisé. Plus précisément, il faudrait utiliser une « job » Map par objet Scan créé par l'application et lancer le tout en parallèle dans un environnement distribué plus grand que trois nœuds. Le Reducer serait utilisé pour effectuer les exclusions de variants en fonction des échantillons.

Ensuite, il faudrait revoir le schéma de données du laboratoire Guy Rouleau dans son ensemble. Nous avons uniquement eu un ensemble de données limité et qui date de l'an dernier sur lequel travailler. Selon ce que M. Ousmane Diallo nous a mentionné lors de notre visite au laboratoire, le schéma SQL du CRCHUM a changé depuis. Il faudrait ensuite réviser entièrement le schéma HBase sur réception du schéma SQL complet du laboratoire Rouleau. L'objectif de cette opération serait de valider que le schéma HBase développé convienne réellement aux besoins du laboratoire et est intègre par rapport au schéma actuellement utilisé. Il faudrait, si nécessaire, revoir le schéma HBase et le modifier pour qu'il reflète les derniers changements du CRCHUM.

Il serait aussi souhaitable de revoir la composition de la clé de rangée selon l'occurrence des paramètres de recherche. En effet, la composition de la clé est très importante, puisqu'elle définit l'ordonnancement effectué par HBase. Cependant, avec le peu d'information que nous disposons sur les requêtes effectuées par les employés du laboratoire Guy Rouleau, nous sommes loin d'être certains que la composition actuelle de la clé concatène de manière optimale les paramètres de recherche.

Par ailleurs, il serait intéressant de montrer le travail accompli au laboratoire du CRCHUM afin de récolter une rétroaction de leur part. De cette façon, il serait possible de connaitre leur opinion sur les fonctionnalités de l'application ainsi que sur les requêtes effectuées et les résultats présentés. Avec la rétroaction du client, il serait possible de produire de nouveaux

requis, qu'ils soient logiciels ou non, et d'itérer dessus afin d'améliorer continuellement le prototype et peut-être le voir un jour remplacer le système S2D actuel.

Afin de comparer les produits disponibles sur le marché, il serait pertinent d'installer et de tester les performances d'autres systèmes de base de données distribuées. Par exemple, il serait possible de reproduire le même genre de solution, mais avec HyperTable ou Cassandra. Ensuite, il faudrait analyser les performances de ces systèmes, les comparer à HBase et les classer selon la complexité de mise en place. Puisque ce projet a aussi pour but de fournir un travail pratique à un futur cours de l'ÉTS, la solution ne doit pas être excessivement complexe à mettre en place.

Enfin, il n'y a de limites que celles que l'on s'impose lorsqu'il est question d'amélioration logicielle. Les possibilités d'amélioration d'un tel système sont infinies.

LISTE DE RÉFÉRENCES

George, L. (2011). *HBase: The Definitive Guide.* O'Reilly Media.

The Apache Software Foundation. (2012). *Apache Sqoop*. Consulté le août 09, 2012, sur Sqoop: http://sqoop.apache.org/

BIBLIOGRAPHIE