Utilisation d'une Grille de Calcul (GRISBI) pour le Traitement de Données NGS (Next Generation Sequencing)

Florence Maurier(2,1), Alexis Groppi(2,1), Aurélien Barré(2,1), Delphine Naquin(3), Aurélien Roult(3), Clément Gauthey(4), Christophe Blanchet(4), and Tiphaine Martin(1,2,5),

1 tiphaine.martin@labri.fr, LaBRI, UMR5800 CNRS, 351 cours de la Libération, 33400 Talence, Cedex, France

2 {alexis.groppi, aurelien.barre, florence.maurier}@u-bordeaux2.fr, Université de Bordeaux, CBiB, 146 Rue Léo Saignat, Centre de Génomique Fonctionnelle Bordeaux, 33076, Bordeaux, Cedex, France

3 {delphine.naquin,aurelien.roult}@irisa.fr, EPI Symbiose, INRIA/ IRISA, Campus de Beaulieu, 35042, Rennes, Cedex, France

4 {christophe.blanchet, clement.gauthey}@ibcp.fr, Infrastructure Distribuée pour la Biologie, IDB IBCP FR3302 CNRS, 7 passage du Vercors, 69007, Lyon, France

5. INRIA Bordeaux Sud-ouest, 351 cours de la Libération, 33400 Talence, Cedex, France

Overview

Next-generation sequencing technologies (NGS) can now quickly obtain complete genome sequences. However, obtaining sequences of "high quality", fully assembled, with no parts missing and with no ambiguity remains problematic. Indeed, in the case of *de novo* sequencing, the sequence assembly step is very demanding in terms of computational time and memory. Bioinformatics is increasingly required to compare an explosion of new and complex data sets. Storage and analysis of such data and their outputs are overwhelming many local servers, and it is therefore necessary to move to alternative infrastructures. To support and respond to such large scale biological problems, distributed infrastructure GRISBI (Grid Support to Bioinformatics) was initiated as part of the network ReNaBi (French Bioinformatics Platforms Network) and the French Grids Institute.

Drawing a parallel between the local server paradigm, and a paradigm using an experimental cluster (grid computing) for NGS data assembly examples, we have demonstrated the feasibility and utility of a computing grid under certain conditions. If the computing resources, memory capacity, and the storage are not accessible locally, then the grid provides a significant benefit.

Finally, to make it easily accessible to a group of novice users for example experimental biologists, it will be necessary to ensure user friendly access to the system using simple tools and a workflow manager, web service etc. and to offer the choice of the best infrastructure for each application (i.e. local server, cluster, local grid, cloud computing, etc). The virtual organization GRISBI attempts to address these issues.

Keywords

NGS, Assemblage *de novo*, Grille de calcul, Calcul intensif, Gestionnaire de workflow.

Enjeux scientifiques, besoin de la grille

Les approches de séquençage de nouvelle génération (NGS) ont permis et permettent désormais d'obtenir très rapidement des séquences de génome entier. Cependant, l'obtention de séquences « de haute qualité », totalement assemblées, sans régions manquantes ni ambiguïté reste problématique. En effet, dans le cas de séquençage *de novo,* l'étape de l'assemblage est très coûteuse en temps de calcul et en mémoire. La bioinformatique est confrontée à une explosion de ces nouvelles données. Le stockage et l'analyse des données deviennent difficiles sur des serveurs locaux, et il est donc nécessaire de passer à d'autres infrastructures. Les grilles de calculs peuvent permettre de résoudre ce défi. Cependant, ces infrastructures de calculs ne sont pas facilement accessibles aux utilisateurs novices et en

particulier aux biologistes.
Pour accompagner et répondre à de telles problématiques de biologie à grande échelle, l'infrastructure distribuée GRISBI (Grille Support pour la Bioinformatique) [1] a été initiée dans le cadre du réseau ReNaBi [10] et en collaboration avec l'Institut des Grilles. Les objectifs sont de permettre la mutualisation des ressources de plates-formes nationales de bioinformatique pour la communauté bioinformatique nationale et européenne, de faciliter l'utilisation du calcul distribué aux biologistes et aux bioinformaticiens (optimisation, fiabilité de l'utilisation des ressources informatiques, accessibilité aux données, rapidité d'exécution).

Développements, déploiement sur la grille
L'infrastructure distribuée GRISBI se base sur la couche logicielle de la grille européenne EGEE, gLite [9]. Ainsi, grâce au mécanisme d'organisation virtuelle de gLite, ici vo.renabi.fr, l'infrastructure distribuée GRISBI rassemble actuellement une partie des ressources des plates-formes de bioinformatique partenaires du projet (dont le CBIB, Genouest, PRABI-IBCP et – LBBE, et plus récemment GenoToul), et de celle des mésocentres de calculs pluridisciplinaires (CRI Univ. Lille 1, IPHC Strasbourg et MCIA Bordeaux). Ce qui correspond à environ 900 processeurs et 26To de mémoire .
Pour adapter l'utilisation de la grille de calcul aux problématiques bioinformatiques, un ensemble d'outils et d'aide ont été mis en place :

- Déploiement de données et de logiciels : un ensemble d'outils et de bases de données communément utilisés en bioinformatique ont été préinstallés sur les sites avec l'attribution de VOTAGs. Ces logiciels sont directement décrits dans $PATH des serveurs de calculs. Les bases de données sont installées sous les NFS-ro des serveurs de calculs (/biodb/grisbi/...) et mises à jour par le logiciel BioMaj [8].
- Création de commandes simplifiées : les commandes gLite les plus fréquemment utilisées ont été adaptées et commencent par « gr ». Par exemple, la commande « grsub » permet, si besoin est, (1) d'initialiser le proxy, (2) de créer un home LFC, (3) de configurer l'environnement, (4) de soumettre le job, et (5) de garder le jobID.
- Accès au stockage de type XtreemFS, en complément au stockage de type gLite : il permet à l'utilisateur d'avoir accès à un volume distribué entre les sites avec un montage local tout en gardant une authentification et une sécurité X509.
- Création d'une communauté tant côté utilisateur que technique : l'adhésion à la communauté GRISBI via le site web permet d'accéder aux ressources de calculs et de stockage. Des listes de diffusion, des formations, et des journées scientifiques ont été mises en place pour échanger et s'entraider à partir des différentes expériences sur les grilles.

Outils, difficultés rencontrées
Afin d'évaluer la faisabilité et l'apport d'une grille de calcul telle que RENABI GRISBI pour l'analyse de données de type NGS, nous avons défini plusieurs cas d'utilisations et testé plusieurs solutions logicielles. Nous avons effectué des comparaisons avec d'autres infrastructures de calculs. Nous avons choisi d'utiliser une source de données maîtrisées : le génome complet de la levure *Saccharomyces cerevisiae* [2]. A partir de ces séquences chromosomiques, nous avons généré des jeux de lectures artificiels simulant des séquençages de type Illumina et de type Roche 454 grâce au logiciel MetaSim avec une couverture de 30X

[3]. Ainsi, un génome de 12Mbp (~ 12Mo) génère environ pour chaque simulation 1 à 3Go de données. À partir de ces jeux de données, nous avons réalisé des alignements sur le génome de référence avec le logiciel Burrows-Wheeler Aligner (BWA) [4] et des assemblages *de novo* avec les logiciels ABySS [5] et Ray [6].
Avant de pouvoir tester des outils sur la grille, il est nécessaire de maîtriser leurs exécutions et, en particulier, leurs entrées et sorties. C'est pourquoi nous avons d'abord effectué des tests sur d'autres infrastructures. Cependant, ces infrastructures ne sont pas spécifiques pour la bioinformatique et il a été nécessaire d'installer les logiciels.
De plus, nous disposons d'une machine virtuelle où le gestionnaire de workflows Ergatis est déjà déployé et dont l'un de ses modules gère les traitements avec BWA. Cette machine virtuelle a été créée par nos collègues de GenoToul, qui ont une expertise pour l'ajout de nouveaux modules et pour son utilisation. Nous y avons donc expérimenté cet outil avec nos jeux de données simulées et leur référence. Les temps d'exécution ainsi obtenus ont ensuite été comparés avec ceux provenant de la grille GRISBI et ceux provenant de la plateforme expérimentale PlaFRIM [13].
Dans un second temps, afin de maîtriser les assembleurs *de novo* ABySS et Ray, nous les avons installés et testés sur une machine personnelle, et non sur la machine virtuelle « Ergatis » car les modules pour les gérer n'existent pas encore. Manquant de mémoire sur la machine personnelle, Ray n'a alors donné aucun résultat. Enfin, nous avons testé ces deux outils sur la grille GRISBI et la plateforme expérimentale PlaFRIM.

Les difficultés que nous avons rencontrées sont de type matériel (mémoire) et de type interface homme-machine (facilité d'accès pour des non-informaticiens).
Pour nos analyses, il est nécessaire de répéter le même calcul en faisant varier un paramètre (cf. § Résultats scientifiques). Les calculs que nous ayons effectués sur la grille utilisent une grande quantité de mémoire. Il arrive que certaines tâches ne s'effectuent pas par manque de ressource mémoire lors de leur exécution. Dans ce cas, il faut alors relancer manuellement ces tâches. Actuellement, il n'est pas possible de spécifier la quantité de mémoire requise par une tâche. L'implémentation de ce paramétrage permettrait de mieux gérer l'utilisation de la mémoire pour des tâches simultanées sur la grille.
Par ailleurs, ces infrastructures de calcul ne sont pas facilement accessibles aux utilisateurs novices et, en particulier, aux biologistes. En effet, les applications bioinformatiques sont communément accessibles au travers d'interfaces web intuitives. Afin de favoriser l'utilisation des logiciels de bioinformatique sur la grille, il serait souhaitable de mettre en place des interfaces répondant à ces critères. Par exemple, il n'est pas possible de lancer des applications via l'interface web d'un gestionnaire de workflows. Cette limitation est due au fait qu'une tâche doit être lancée via un certificat personnel (et non « machine »).

Résultats scientifiques
Nous pouvons remarquer que les temps d'exécution des trois outils bioinformatiques avec des données Illumina single-end sur GRISBI sont sensiblement équivalents à ceux qui sont obtenus sur les deux autres environnements (machine de développement de bureau et grappe de calculs) (cf. Tableau 1). Pour les autres paires outils-données de calcul, la plateforme expérimentale PlaFRIM a des temps d'exécution plus courts que sur une machine personnelle et que sur la grille (cf. Tableau 2).
Cependant, les assembleurs ABySS et Ray prennent en paramètre la longueur de K-mer

souhaitée. La qualité de l'assemblage dépend de ce paramètre dont la valeur ne peut en aucun cas être prédéterminée.
GRISBI présente alors un avantage certain avec les jobs paramétriques qu'il propose. Un job paramétrique permet de lancer, en une seule manipulation, plusieurs fois le même job en lui passant un paramètre dont la valeur varie, dans notre cas : la longueur de K-mer. De même, nous retrouvons ce type de fonctionnalité dans un gestionnaire de ressources (par exemple Torque) pour une grappe de calculs locale telle que PlaFRIM.
Le tableau 3 présente ainsi la comparaison des durées de jobs paramétriques pour dix valeurs de K-mer différentes avec les temps estimés de dix exécutions locales successives. Un écart de temps significatif est alors observable en faveur de la grille GRISBI ou de la grappe de calculs locale PlaFRIM.

Outils	Local*	Ergatis**	PlaFRIM ***	GRISBI
BWA	Pas installé	Index : 20s aln : 553 s samse : 80 s	Index : 7 s aln : 306 s samse : 45s	Index : 15s aln : 485 s samse : 70 s
ABySS	8 min	Absence de module	7 min	10 min
Ray	« std::bad alloc »	Absence de module	Pas installé, problème avec mpich2	85 min

Tableau 1: Temps moyen d'exécution sur des données Illumina single-end sur 10 job disctinct (en excluant le temps de temps de transfert et le temps d'attente des jobs). * Processor: Intel Core I5(2,6 GHz), RAM : 3 Go ** Processor : Six-Core AMD Operon (2,1 GHz), RAM : 16 Go *** Cluster fourmi de la plateforme PlaFRIM, Processor : 2 Quad-core Nehalem Intel® Xeon® X5550, RAM : 24Go

Données	Local*	PlaFRIM**	GRISBI
Illumina single-end	8 min	7 min	10 min
Illumina paired-end	67 min	33 min	81 min
454 single-end	« std::bad alloc »	60 min	162 min

Tableau 2: Temps moyen d'exécution sur différentes données avec l'assembleur *de novo* ABySS sur 10 job disctinct (en excluant le temps de temps de transfert et le temps d'attente des jobs). * Processor: Intel Core I5(2,6 GHz), RAM : 3 Go ** Cluster fourmi de la plateforme PlaFRIM, Processor : 2 Quad-core Nehalem Intel® Xeon® X5550, RAM : 24Go

Outils	Local*(exécutions successives)	PlaFRIM** (job array pour pbs)	GRISBI (job paramétrique)
ABySS	8 min X 10 ~ 80 min	< 15 min	< 60 min
Ray	« std::bad alloc » (85 min**** X 10 ~ 850 min)	Pas installé, problème avec mpich2	~ 180 min

Tableau 3: Temps d'exécution sur des données Illumina single-end pour 10 exécutions avec changement de k-mer. * Processor: Intel Core I5(2,6 GHz), RAM : 3 Go, ** Cluster fourmi de la plateforme PlaFRIM[13], Processor : 2 Quad-core Nehalem Intel® Xeon® X5550, RAM : 24Go **** si nous avons un serveur local avec les caractéristique du nœud de calcul de GRISBI ayant réussi à faire tourner le job Ray/Illumina single-end

Nos premiers résultats démontrent donc la faisabilité et l'apport considérable de l'utilisation d'une grille de calcul pour ces traitements. En effet, d'une part, les temps de calculs sont réduits par rapport à des serveurs locaux de capacité moyenne, mais il est également possible, de lancer, de n'importe où, plusieurs analyses simultanées dont un paramètre varie, comme pour le cas d'assemblages *de novo*. Une telle infrastructure distribuée est un atout lorsque les ressources de calcul, de mémoire, ou de stockage ne sont pas accessibles localement.
De ces cas d'utilisation, un ensemble de modèles de lancement de traitements (bash, JDL) a été créé et rendu disponible à l'ensemble de la communauté via le site GRISBI [1].
Ce besoin ne pourra qu'augmenter, si nous tenons compte du déluge de données issues des nouvelles technologies en biologie et en particulier, pour les données de séquences.

Perspectives

La suite de notre projet consiste maintenant à intégrer l'envoi de traitements GRISBI au sein du gestionnaire de workflows ERGATIS [7]. L'utilisateur pourra ainsi lancer une chaîne de traitements via l'interface web du gestionnaire de workflows qui enverra, en toute transparence sur la grille, les opérations nécessitant le plus de ressources et qui récupérera les différentes données obtenues.
En attendant de pouvoir lancer des traitements de données via une interface web, la grille GRISBI est utilisée, de manière classique, pour l'assemblage de données réelles. Elle servira également au projet Assemblathon 2 [11], dont l'objectif est de comparer les logiciels et les paramètres d'assemblage. Dans notre cas, nous travaillerons sur le génome du *Boa constrictor* à queue rouge (*Boa constrictor constrictor*) (140 Go de données brutes en Illumina Mate Pair) dans le cadre d'un assemblage *de novo* avec le logiciel ABySS.

Références

1. http://www.grisbio.fr
2. http://db.yeastgenome.org/cgi-bin/seqTools
3. http://ab.inf.uni-tuebingen.de/software/metasim/
4. http://bio-bwa.sourceforge.net/
5. http://www.bcgsc.ca/platform/bioinfo/software/abyss
6. http://sourceforge.net/apps/mediawiki/denovoassembler/
7. http://ergatis.sourceforge.net/
8. O. Filangi, Y. Beausse, A. Assi, L. Legrand, J.M. Larré, V. Martin, O. Collin, C. Caron, H. Leroy and D. Allouche,BioMAJ: A flexible framework for databanks synchronization and processing. Bioinformatics, 24:1823-1825, 2008.
9. http://www.glite.org
10. http://www.renabi.fr
11. http://www.assemblathon.org
12. http://www.idgrilles.fr/
13. https://dihpes.bordeaux.inria.fr/doku.php?id=start

Remerciement

A Christophe Caron, Olivier Collin, Stéphane Delmotte, Christelle Eloto, Gael Even, Pierre Gay, Daniel Jacob, Didier Laborie, Nouridine Melab, Alexis Michon, Frédéric Plewniak, Franck Samson, Bruno Spataro et les membres de FranceGrille pour leurs participations dans GRISBI.