

Le projet GénoGRID

Une grille expérimentale pour la génomique

Dominique Lavenier* Rumen Andonov*
Hugues Leroy* Laurent Mouchard†
Michel Hurfin* Frederic Guinand‡

* IRISA – Campus de Beaulieu – 35042 Rennes cedex

† ABISS – Université de Rouen – 76821 Mont Saint Aignan cedex

‡ LIH – Université du Havre – 5, rue Philippe Lebon – BP 540 – 76058 Le Havre

Courriel : lavenier@irisa.fr

Résumé

Cet article présente le projet GénoGRID financé par une ACI GRID (Action Concertée Incitative – Globalisation des Ressources Informatiques et des Données), action mise en place en 2001 par le ministère de la recherche. L'objectif est, d'une part, de structurer une grille de ressources informatiques pour fédérer des calculs lourds de la génomique et, d'autre part, d'en offrir un accès simple et transparent à la communauté scientifique

Mots-clés : *grille, calcul intensif, parallélisation, distribution, génomique*

1 Introduction

Une grille, telle qu'on la définit habituellement, est un ensemble de ressources composées de deux types d'éléments : des calculateurs et des données [5]. Les calculateurs peuvent être variés : serveurs de données, machines parallèles, fermes de PC, architectures spécialisées, *etc.* Les données, dans notre cas, sont celles de la génomique : banques de séquences, génomes complets, profils d'expression, articles scientifiques, *etc.* Le rôle d'une grille est de fournir un service sans que l'utilisateur ne se préoccupe ni de la manière dont ce service est mis en œuvre, ni du lieu où il s'exécute. L'image du réseau électrique est souvent proposée pour illustrer ce concept : l'énergie est produite de diverses façons (centrale thermique ou nucléaire, éolienne, barrage, *etc.*) ; elle est distribuée par un réseau complexe et est finalement accessible à tous par une simple prise électrique. L'utilisateur négocie un contrat avec EDF sur la puissance désirée, la tarification horaire, la formule d'abonnement, *etc.*, mais ne se préoccupe pas de savoir d'où provient cette énergie ni comment elle parvient jusqu'à lui.

Transposée au monde de l'information, et plus spécifiquement au projet GénoGRID, la prise électrique est un portail *WEB* sécurisé par lequel un certain nombre de services relatifs à des traitements coûteux en génomique sont offerts. La grille est avant tout destinée à la communauté scientifique dans le domaine de la génomique. Elle a pour objectif de fournir un accès simple et transparent, à l'image des serveurs existants qui, par exemple, proposent des services standard d'analyse de séquences.

La grille est constituée d'un ensemble de ressources hétérogènes qui appartiennent à des centres de calcul ou à des centres de recherche. Ces ressources sont supposées posséder tout ou partie des données relatives au domaine de la génomique. Lors d'une requête, le portail aiguille automatiquement la tâche vers la ou les ressources les plus appropriées, que ce soit en terme de disponibilité des données, en terme de logiciel ou en terme de ressources informatiques disponibles. L'ensemble est connecté par un réseau à haut débit.

Les applications privilégiées sont des traitements lourds de la génomique, c'est à dire des traitements de plusieurs centaines, voire plusieurs milliers d'heures de calcul. Deux types d'applications sont envisagées, celles dites « généralistes », disponibles pour tous les usagers et qui présentent un intérêt commun, et celles dites « spécifiques » qui relèvent de besoins particuliers d'un laboratoire. Mais dans tous les cas, l'accès est contrôlé et l'utilisateur identifié.

Le projet se décline en plusieurs actions :

- la mise en place d'une grille ;
- le développement d'un portail *WEB* sécurisé ;
- l'accès transparent aux ressources de calcul ;
- la parallélisation d'applications.

Le but de cet article n'est pas de décrire en détail l'ensemble du projet. Il a seulement pour ambition, à travers une présentation succincte des différents points évoqués ci-dessus, de faire connaître la philosophie du projet GénoGRID et, éventuellement, de susciter des coopérations avec les laboratoires ou les équipes de recherche intéressées par ce thème, notamment sur la partie application.

La suite du texte est structurée de la manière suivante : la section 2 présente concrètement la grille et donne un aperçu des modalités d'accès. La section 3 précise la notion d'accès transparent et indique sommairement comment les traitements seront distribués. La section 4 traite du parallélisme ; elle en montre les différentes possibilités au niveau d'une grille de calcul. Enfin, la section 5 conclut par les quelques applications qui seront mises en place et qui serviront de test grandeur nature.

2 La Grille et le portail WEB

La grille de calcul, telle que nous l'envisageons actuellement, est composée des ressources informatiques suivantes :

- un calculateur parallèle *Sun Fire* 6800, 16 processeurs, 16 Go mémoire, localisé au Centre de Ressources Informatiques de l'Université de Rennes 1 ;
- une machine *Sun* E450, INRA centre de Rennes (Le Rheu) ;
- une machine *Sun Fire* 4800 de la Station Biologique de Roscoff
- une machine Compact localisée à IFREMER : 9 nœuds ES45 quadri-processeurs, 92 Go mémoire (Brest) ;
- un cluster de 17 PC bi-processeurs 256 Mo mémoire sous *Linux* à l'université de Rouen ;
- une machine SGI ORIGIN 2000 de 64 processeurs 32 Go mémoire au CRIHAN (Rouen) ;
- une machine Compaq ES40 du CIB (centre intégré de bio-informatique) de la génopôle de Lille.

Cette structure n'est pas figée : elle pourra évoluer au cours du temps en fonction des acquisitions des divers centres impliqués dans le projet ou des centres qui souhaiteraient collaborer à GénoGRID. Actuellement, les pôles sont interconnectés par le réseau régional à haut débit *Mégalis* (ou *Renater 2*).

Le portail Internet, quant à lui, doit fournir à une communauté homogène d'utilisateurs (biologistes et bio-informaticiens) un moyen d'accès simple aux ressources de cette grille. Notre objectif n'est pas de couvrir tous les domaines de recherche ouverts par le *Grid Computing*, mais de proposer rapidement des solutions aux quelques questions suivantes : comment régler l'accès transparent aux ressources distribuées qui sont gérées par des organisations distinctes n'ayant pas forcément les mêmes procédures d'exploitation ? comment assurer la migration des résultats tout en garantissant leur confidentialité ? comment régler l'allocation des ressources et le partage de charge ? comment assurer la sécurité des échanges ? *etc.*

L'authentification des utilisateurs se fera une en seule fois par l'attribution de signatures électroniques (certificats). Le portail sera organisé en répertoires de données avec, pour chaque utilisateur, ses prérogatives (droits d'accès aux applications, aux ressources, *etc.*), et pour chaque application, la liste des sites où elle peut s'exécuter, la liste de ses paramètres, *etc.* Ces répertoires de données seront exploités à l'aide du protocole d'accès réseau LDAP (*Lightweight Directory Access Protocol*) éventuellement répartis et dupliqués. Ces bases d'informations - qui sont similaires à des bases de données - sont le point central du portail. A titre d'exemple nous pouvons citer MDS (*Metacomputing Directory Service*) du projet *Globus* [1] ou celui du projet *MetaWeb* [2].

3 Accès transparents à la grille

Les problèmes de placement de tâches et d'équilibrage de charge sont essentiels dans le cas d'une Globalisation des Ressources Informatique et des Données (GRID). A la complexité intrinsèque de ces problèmes qui peuvent être formulés comme la recherche du meilleur compromis entre équilibrage de charge et minimisation du surcoût

dû aux communications entre les tâches vient s'ajouter celui de la localisation des données, paramètre incontournable dans les applications bio-informatiques. Le problème général peut être formulé comme celui de la maximisation du parallélisme conjointement à la minimisation du temps perdu en communication sous les contraintes de localité de données et de tolérance aux pannes. Un utilisateur doit pouvoir soumettre une requête sans avoir à sélectionner lui-même les machines qui effectueront le traitement de la tâche correspondante. Naturellement, ce placement se fera en fonction des droits d'accès accordés et dépendra de la nature de la tâche ainsi que des contraintes additionnelles éventuellement formulées (puissance minimale des processeurs, disponibilité du code ou des données, *etc.*). Outre le confort d'utilisation que procure cette transparence, l'objectif est bien évidemment de mieux gérer les ressources de calcul afin de diminuer, autant que faire se peut, les temps d'attente.

Une attribution dynamique des tâches aux calculateurs ne pose pas trop de problème lorsque le pool de ressources disponibles est statique. Dans la pratique, ce pool évolue dynamiquement. Ces évolutions sont prévisibles (ajout ou retrait de machines par les administrateurs) ou imprévisibles (pannes de machines, coupures réseaux). À l'inverse d'une grappe de machines, un réseau de dimension nationale est caractérisé par une dispersion géographique des calculateurs. Les calculateurs, qui sont hétérogènes, évoluent dans un environnement qui se révèle donc être non fiable et asynchrone :

un calculateur peut stopper prématurément son exécution à la suite par exemple d'une panne matérielle. De même, les réseaux de communication peuvent également connaître des défaillances (perte de messages, partitionnement temporaire). Ces incidents doivent pouvoir être tolérés et masqués par l'environnement logiciel ;

il est souvent impossible de déterminer au préalable une borne maximale sur les temps de transfert d'informations entre sites distants (fluctuation de la charge du réseau) ou sur les durées d'exécution des actions (fluctuation de la charge des processeurs).

Pour résumer, les réseaux ciblés sont caractérisés par un ensemble d'hypothèses faibles, ce qui complique la mise en œuvre de services vitaux au sein du groupe de ressources (détection des défaillances, observation d'un état global, prise de décision, *etc.*). De fait, la gestion des ressources disponibles qui s'appuie sur ces services élémentaires est elle aussi plus complexe.

L'attribution d'une tâche à une machine peut être faite par un processus unique (approche centralisée) ou, au contraire, être le résultat d'une concertation entre les différents sites susceptibles d'exécuter le traitement. Dans un environnement non fiable où des défaillances peuvent se produire, une approche centralisée n'est pas judicieuse. La répartition du rôle de coordinateur de la grille de calcul entre ses différents composants permet de mieux résister aux défaillances mais pose naturellement des problèmes de cohérence. Les solutions susceptibles d'être apportées sont relativement simples lorsque les ressources disponibles sont interconnectées par un réseau à très haut débit et de faible latence (grappes de serveurs par exemple). Dans le cas d'un réseau asynchrone, le problème est beaucoup plus ardu.

Dans le cadre du projet GénoGRID, le logiciel *Eden*, développé à l'IRISA, sera pleinement exploité [3] [4]. Dans un environnement asynchrone, il permet des prises de décision unanimes au sein d'un groupe de machines dont la composition peut évoluer dynamiquement. D'une part, cet outil nous permet donc de gérer l'évolution dynamique de la composition du groupe ressources (ajout ou retrait de machines, pannes). D'autre part, il permet à toutes les ressources disponibles de prendre des décisions communes afin de s'octroyer ou de refuser temporairement des tâches qui sont en attente de placement. Une fois affectée à un calculateur, une tâche est exécutée par celui-ci dès qu'il est disponible. En cas de défaillance du calculateur, la tâche est réattribuée à une autre ressource sans aucune intervention de l'utilisateur.

Le mécanisme de placement que nous proposons tient compte des défaillances potentielles des ressources de calcul. Cet aspect n'est pas réellement traité dans les systèmes de gestion de ressources distribués existants tels que *Globus*, *Gallop* ou *Condor*. Certes, certains de ces systèmes (en particulier *Globus*) offrent des mécanismes de notification de défaillances. Mais, dans tout les cas, la décision d'allouer telle machine plutôt qu'une autre est prise par un gestionnaire unique. Ces services d'allocation ne sont donc pas fiables au sens où des pannes peuvent venir affecter la disponibilité du gestionnaire et donc du service. Par ailleurs, le remplacement automatique des tâches suite à la défaillance d'une ressource n'est pas pris en compte par des outils tels que *Globus* qui, par contre, suivant d'autres points de vue, offrent une panoplie de services plus générale.

4 Parallélisme

Dans une grille, comme celle que nous considérons, le parallélisme intervient à plusieurs niveaux. Le premier, bien sûr, est la distribution d'un calcul sur différents nœuds. Dans ce cas, le traitement est « *découpé* » en tâches élémentaires, puis chaque tâche est acheminée vers un nœud pour être exécutée. A ce niveau, les défis sont multiples : il faut savoir répartir dynamiquement et de manière équilibrée les tâches entre les différents nœuds de la grille ; il faut savoir traiter une non-terminaison de tâche (défaillance d'un nœud) et sa ré-allocation sur un autre site ; il faut savoir prendre en compte les interruptions ponctuelles du réseau ; *etc.* De plus, l'application doit se prêter à ce type de parallélisme : pour être efficace, les communications entre tâches doivent être extrêmement réduites. Dans les applications que nous avons recensées (comparaison banque à banque, *protein threading*, détection de répétitions, *etc.*) , bon nombre d'applications partagent ce critère : elles peuvent être découpées en tâches indépendantes qui demandent, à l'issue du calcul, une simple fusion des résultats.

Le second niveau de parallélisme concerne les nœuds qui, dans la plupart des cas, sont des systèmes parallèles : calculateurs ou fermes de PC par exemple. La parallélisation d'une application est alors plus subtile qu'un simple découpage en tâches indépendantes. Ces systèmes possèdent des réseaux d'interconnexions fiables et performants entre les processeurs. La contrainte de minimisation des communications entre les tâches est alors affaiblie : les tâches peuvent coopérer efficacement par le biais de protocoles tels que *MPI* ou *OpenMP*. La classe des applications qui peuvent être supportées est plus large que précédemment, mais chaque application demande un travail de parallélisation plus conséquent.

Enfin, on peut considérer un troisième niveau de parallélisme, celui inhérent aux nœuds spécialisés. Ces derniers sont en général dédiés à un calcul spécifique et sont supportés, par exemple, par des puces de silicium spécialement conçues pour cet objet. L'exemple de la recherche de motifs avancés (ou signature), en cours d'étude à l'IRISA, illustre ce niveau : un nœud est un ensemble de modules composés d'un couple (disque, carte reconfigurable), l'ensemble étant interconnecté par un réseau rapide. Une requête configure la carte en un filtre matériel qui ne laisse passer que les séquences qui contiennent un motif particulier. Les filtres sont en réalité des architectures parallèles constituées de dizaine de petits processeurs spécifiques qui ont pour rôle unique de filtrer à la volée les données dès la sortie des disques pour ne faire remonter que l'information pertinente. C'est un parallélisme à grain fin qui demande, cette fois, une étude approfondie et un temps de développement important.

Ces différents niveaux de parallélisme peuvent, bien sûr, être combinés. Tout dépend de la politique de gestion de la grille. On peut choisir d'affecter une seule application par nœud, ce choix résultant de la meilleure adéquation entre la requête et le triplet « données, ressources, logiciels » disponible sur le nœud. Dans ce cas, seul le second niveau de parallélisme intervient, le portail servant simplement à sélectionner le meilleur site où le traitement peut s'exécuter. A l'inverse, on peut d'emblée répartir le traitement sur tous les nœuds susceptibles de contribuer au calcul, le travail du portail – ou plus exactement du gestionnaire de la grille – étant alors d'assurer une répartition équitable entre les nœuds. Dans ce deuxième cas de figure, si on suppose que les nœuds sont des systèmes parallèles, les deux premiers niveaux de parallélisme sont activés simultanément. Enfin, dans l'hypothèse où la grille possède plusieurs nœuds spécialisés attachés, par exemple à des bases de données particulières, les trois niveaux de parallélisme sont exploités.

5 Applications

Dans ce dernier paragraphe, nous donnons un aperçu du type d'applications que peut supporter une grille dans le domaine de la génomique. Les applications qui, dans un premier temps, seront testées sur la grille proviennent essentiellement des champs de compétences des partenaires impliqués dans le projet GénoGRID. Elles ne représentent en aucune façon une liste exhaustive. Elles illustrent les potentialités en la matière.

Comparaison intensive de séquences : l'objectif est principalement de pouvoir comparer un grand nombre de séquences dans un temps raisonnable. Par exemple, le laboratoire du GERM (unité INSERM U 435 à Rennes) confronte régulièrement la banque virale de GenBank contre une banque maison, qui aujourd'hui représente plus de 200 000 séquences d'ADN, pour essayer de mettre en évidence des similarités qui peuvent refléter des infections virales potentielles. Les similarités recherchées étant faibles, elles impliquent des méthodes sensibles, typiquement l'usage de *tblastx*, ce qui correspond à plusieurs centaines d'heures de calcul sur un serveur performant. L'exécution de ce traitement sur une grille dotée de plusieurs calculateurs parallèles réduit ce temps à quelques heures.

Détection de séquences répétées : les séquences répétées sont omniprésentes dans les séquences biologiques, que l'on considère les répétitions en tandem (micro et mini-satellites, répétitions avec évolution) ou les répétitions dispersées (éléments transposables). L'utilisation de la grille de calcul permettra une recherche systématique de tels éléments dans tous les principaux génomes séquencés à ce jour. Ainsi, une base de données de séquences répétées sera construite et mise à disposition de la communauté. Elle sera interrogeable au travers d'une interface *WEB*, à la manière de *Sequence Retrieval System*. L'utilisateur aura également la possibilité de rechercher des répétitions dans ses propres séquences, en choisissant un type particulier de répétitions décrit par une méthode de recherche et un jeu de paramètres.

Alignement de séquences protéiques sur des structures tridimensionnelles : les *méthodes de reconnaissance de repliements* reposent sur l'alignement d'une séquence avec une structure protéique tridimensionnelle. On cherche dans un ensemble de repliements (cœurs) celui qui se révèle le plus compatible – sur la base d'une fonction de score empirique – avec la séquence requête. Dans le cas le plus général, un tel alignement est un problème NP-complet. Un des meilleurs algorithmes exact est celui de Lathrop et Smith [6]. Il aligne des protéines de l'ordre de 200 acides aminés en quelques minutes. Cependant, pour des raisons statistiques, cet alignement est réitéré une centaine de fois et, dans le cas présent, sur une bibliothèque de 1700 cœurs. Dans le cadre du projet GénoGRID, l'utilisateur donnera, dans un premier temps, sa séquence requête pour l'aligner avec la bibliothèque des repliements existants avec une fonction de score empirique prédéfinie. Dans une seconde étape, nous envisageons de laisser la possibilité de spécifier une fonction de score propre, ainsi que la modification et la création de bibliothèques personnelles de cœurs caractéristiques des repliements.

Recherche de motifs complexes : nous entendons des motifs dont le niveau de description est au delà des expressions régulières pour décrire, par exemple, une signature représentative d'une famille de protéines. Analyser des génomes complets pour exhiber de tels motifs requiert du temps. Les nœuds spécialisés actuellement à l'étude visent l'accélération de ce type d'application en spécialisant sur un support matériel reconfigurable un *parser* de grammaire *context-free* directement en sortie des disques où sont stockées les banques. Avec cette application, l'idée est d'appliquer le concept d'hétérogénéité d'une grille, et ainsi donner un accès complètement transparent à des ressources hyper-spécialisées.

Références

- [1] I. Foster, C. Kesselman, *Globus: A Metacomputing Infrastructure Toolkit*, Intl J. Supercomputer Applications, 11(2):115-128, 1997.
 - [2] D. Laforenza, R. Baraglia *Metaweb, a web-based metacomputing problem solving environment for building complex applications*, Ercim news no 45, avril 2001.
 - [3] M. Hurfin, R. Macedo, M. Raynal et F. Tronel, *A General Framework to Solve Agreement Problems*, In Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems, pages 56--65, Lausanne, Octobre 1999.
 - [4] F. Brasileiro, F. Greve, M. Hurfin, J.P. Le Narzul et F. Tronel, *Eva: an Event-Based Framework for Developing Specialized Communication Protocols*, IEEE Int. Symposium on Network Computing and Applications, Cambridge, MA, USA, February 2002.
 - [5] I. Foster, C. Kesselman, *The Grid : Blueprint for a New Computing Infrastructure*, Morgan Kaufmann publishers, ISBN 1-55860-475-8, July 1998.
 - [6] R.H. Lathrop and T. Smith, Global optimum protein threading with gapped alignment and empirical pair score functions. *Journal of Molecular Biology*, 1996, **255**, 642-665.
-