

Architecture parallèle reconfigurable pour la génomique

Stéphane Guyetan, Steven Derrien, Dominique Lavenier

IRISA, Université de Rennes 1,
Campus de Beaulieu,
35042 RENNES Cedex- France
{sguyetan,sderrien,lavenier}@irisa.fr

Résumé

Nous proposons une architecture reconfigurable spécialisée dans la fouille de données génomiques. Parce que d'une part cette fouille par le contenu est très complexe et que d'autre part la masse de données est très importante, notre architecture repose sur la parallélisation de l'accès aux données et sur le filtrage matériel à la volée de ces données. La nature reconfigurable des filtres permet l'accueil d'applications variées : recherche d'alignements dans les séquences de nucléotides, recherche de motifs à partir d'expressions régulières ou encore recherche d'images par le contenu.

Mots-clés : circuits reconfigurables, disques intelligents, fouille de données, génomique

1. Introduction

La génomique consiste en l'étude des génomes, de leur structure à leur traduction physiologique. Ce domaine récent de la biologie connaît aujourd'hui un essor fabuleux qui doit beaucoup à l'évolution technologique des outils qui génèrent des masses de données colossales.

L'automatisation des protocoles expérimentaux de manipulation de l'ADN permet un séquençage des génomes beaucoup plus rapide ; le génome humain par exemple a été déchiffré entièrement en 2001 avec plusieurs années d'avance. La banque de données européenne de séquences de nucléotides, l'EMBL-Bank [1] qui regroupe, organise et distribue toutes les séquences publiées, connaît une croissance exponentielle (elle double sa taille tous les 10 à 12 mois) et a atteint en décembre 2001 plus de 15 milliard de bases. Ajoutons que certaines avancées algorithmiques du domaine utilisent une représentation redondante des données, ce qui gonfle encore ce chiffre lors d'un traitement informatique.

Le séquençage n'est pas la seule source de données en masse : des domaines en pleine expansion, comme l'analyse du transcriptome (étude de la régulation de plusieurs milliers de gènes en parallèle) et l'analyse du protéome (il s'agit d'identifier et de caractériser l'expression des protéines au niveau cellulaire et sub-cellulaire) génèrent respectivement, grâce aux puces à ADN et à l'électrophorèse bidimensionnelle, des images haute définition représentatives de l'activité d'une cellule à un instant et dans un contexte donné.

La recherche en génomique repose en partie sur la fouille de données dans les bases qui rassemblent toutes ces informations et qui font appel à des algorithmes de complexité souvent quadratique. Pour pouvoir répondre avec un temps et un coût raisonnables à ce besoin en capacité calculatoire, les solutions informatiques classiques qui évoluent selon la loi de Moore (doublement de la puissance de calcul tous les 18 mois) ne suffisent plus. La génomique a aujourd'hui un réel besoin de disposer d'architectures spécialisées. Nous proposons une architecture dans laquelle la base de données est répartie sur un grand nombre de disques durs montés en parallèle, en sortie desquels les données sont filtrées à la volée (cf Fig. 1).

Nous allons maintenant présenter les solutions actuelles, qui dérivent vers les architectures tirées du monde des bases de données ; la deuxième partie expose l'architecture que nous proposons et nous terminons en abordant concrètement quelques exemples d'applications pour ce projet.

2. Solutions matérielles

2.1. Accélération des traitements

On trouve dans la littérature un grand nombre de machines qui intègrent un ou plusieurs coprocesseur(s) spécialisé(s) sous forme de cartes ajoutées sur les bus d'un ordinateur. Ces accélérateurs s'appuient sur un réseau de processeurs élémentaires travaillant en parallèle, implémentés sur des circuits FPGA comme GenStorm [2], ou sur des circuits *full custom* comme BioScan [7] ou Samba [6]. Ils présentent une accélération de 2 à 3 ordres de grandeur par rapport à une station de la même époque. Cette solution a l'avantage de la facilité de mise en oeuvre mais on se retrouve vite limité par la bande passante du bus.

Des produits commerciaux, comme Genematcher2 [10] ou DeCypher [9], annoncent des performances supérieures en terme de puissance de calcul mais uniquement dans le cas particulier de comparaison de banque à banque : un sous-ensemble de chacune de deux banques génomiques est lu sur un disque dur puis est placé dans une RAM utilisée comme cache de données. La phase de calcul intensif peut alors avoir lieu en effectuant successivement la comparaison sur toutes les combinaisons obtenues en prenant deux séquences dans chacun des sous-ensembles. Le besoin en RAM n'est alors que de l'ordre du giga-octet contre plusieurs centaines de giga-octets que représente aujourd'hui une base de données en génomique. Nos applications, qui ont pour point commun de lire toute la base de donnée de façon séquentielle pour effectuer le filtrage, ne peuvent tirer parti de telles architectures et restent limitées par le débit des disques durs en lecture continue, c'est à dire quelques dizaines de méga-octets par seconde.

2.2. Parallélisation des entrées/sorties

La mise en parallèle de disques durs multiplie le débit global ; on trouve cette solution dans Genematcher2 où la base de données est répartie sur 6 disques pour nourrir le réseau de processeurs ; mais on est assez rapidement limité par la bande passante de l'accélérateur. Restent des solutions lourdes utilisant du parallélisme à gros grain : les clusters de calculateurs. On répartit la base de données sur l'ensemble des disques des noeuds et les traitements se font en local, indépendamment, sur chaque noeud.

2.3. L'approche *disques intelligents*

Le début des années 80 connut une recherche active dans les machines spécialisées en bases de données, ayant pour but de rapprocher le traitement du lieu de stockage des données. Des processeurs spécialisés (en tri ou en filtrage) étaient soit intégrés à l'intérieur des disques durs, avec un processeur par piste ou par tête de lecture, soit répartis autour d'une grande mémoire cache qui interfaçait plusieurs disques. Ces recherches furent arrêtées, principalement à cause de leur trop grande spécialisation, du manque de bande passante des circuits alors utilisés et de leur coût lié au temps de développement et à l'utilisation de disques non-standards.

Il y a quelques années, ces principes sont revenus au goût du jour en partant de la constatation que les contrôleurs de disques actuels disposent d'un excès de puissance de calcul non utilisée, et que les constructeurs commencent à intégrer des processeurs RISC conséquents dans leurs contrôleurs [4]. Les travaux menés dans les universités américaines de Berkeley [3], de Carnegie Mellon [4] ou encore de Santa Barbara [5] sur ces architectures de disques intelligents ont montré l'intérêt et la faisabilité de l'approche. Dans ces projets, les disques intelligents reliés à une station hôte par un réseau ethernet reçoivent une petite application logicielle qu'ils exécutent à la volée sur les données sortant du disque. Les algorithmes sont par exemple des filtrages (extraction de connaissances), des convolutions (traitements d'images) ou encore des tris.

3. Présentation de notre architecture

L'architecture vers laquelle nous nous orientons garde ce principe de manipuler les données dès la sortie des disques et de faire cela sur un grand nombre de disques (plusieurs dizaines voire plusieurs centaines) centralisés vers une station. Plutôt que d'utiliser un processeur à usage général, comme dans les projets de disques intelligents, nous préférons implémenter des filtres reconfigurables sur des circuits FPGA. Ces filtres seront préparés très spécifiquement selon l'application (voire directement être fonction de la requête) puis compilés à l'avance (en quelques heures) et stockés sur les disques durs. Au lancement de l'application, ils seront téléchargés en quelques millisecondes dans les circuits FPGA. Cette

reconfigurabilité permet de s'adapter presque instantanément à l'algorithme et au format de données considérés, ce directement dans le câblage du circuit; elle permet en plus l'évolutivité en terme d'algorithme et de volume de données, car avec des filtres plus sélectifs, on peut paralléliser plus de disques sans saturer le réseau d'interconnexion. Cette sélectivité dépend des algorithmes mis en oeuvre et surtout de leur répartition entre les filtres et la station hôte : trop élevée elle nuirait à la sensibilité du filtre, mais trop faible elle surchargerait la station hôte sur laquelle l'identification exacte des résultats de la requête sera finalisée. Ainsi nous pensons choisir un compromis entre 1 et 5% des données entrantes. Notre architecture a donc deux niveaux de parallélisme : (1) à gros grain, considérant que les modules disque/filtre travaillent indépendamment sur des données qui ont été découpées en bloc ; (2) à grain fin à l'intérieur des circuits FPGA qui correspond à la structure pipeline des filtres.

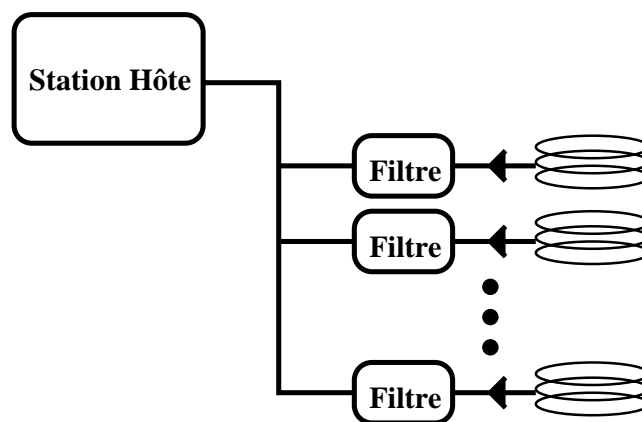


FIG. 1 – Architecture reconfigurable sur le modèle des disques intelligents : la base de données est répartie en blocs de même taille sur l'ensemble des disques durs. Les données traversent le filtre dès la sortie du disque ; seule une fraction pertinente des données converge alors vers la station hôte où les algorithmes sont finalisés. Les filtres sont reconfigurables (implémentation sur composants FPGA) pour traiter une large gamme d'applications.

4. Applications

Nous présentons ici des applications liées à la génomique pour lesquelles le principe du filtrage s'applique particulièrement bien. Ces exemples ne sont pas limitatifs puisque l'utilisation de la technologie reconfigurable ne fixe pas le type de recherche ni la structure des données.

4.1. Recherche d'alignements dans les séquences d'ADN

La première application que nous allons mettre en oeuvre pour tester notre architecture est la recherche de similitudes dans les séquences d'ADN. Il s'agit d'une recherche approximative de chaîne dans un texte, avec un alphabet à quatre symboles correspondant aux quatre bases A,T,C et G constitutives du code génétique. Approximative parce que l'on tient compte des mutations et des insertions/délétions, ce qui rajoute une grande complexité algorithmique. Certains algorithmes classiques en bio-informatique, comme BLAST [8], utilisent une heuristique pour éliminer toutes les séquences trivialement non pertinentes avant les calculs coûteux d'alignement des séquences. C'est précisément cette idée que nous allons adapter à notre architecture : l'utilisateur va donner une chaîne requête qui va être transmise à tous les modules. Dans le scénario le plus simple, chaque module parcourt séquentiellement son disque dur et filtre les séquences à la volée. Ce filtre implémente donc la première partie de l'algorithme de recherche de similarité cablé dans le composant FPGA. Une fraction des séquences stockées sont finalement utilisées pour calculer les scores de similarité.

4.2. Recherche de motifs avec erreurs

La recherche de motifs se prête bien à notre architecture puisqu'il faut parcourir toute la base de données en recherchant les séquences qui possèdent des propriétés particulières, comme par exemple des expressions régulières dans des séquences protéiques. Comme pour la recherche d'alignements, l'introduction de possibilité d'erreurs ou d'approximation augmente énormément la complexité de la tâche. On peut alors appliquer le même principe de filtrage en faisant une recherche de sous-motifs : on ne laisse passer que les séquences qui présentent une conjonction de sous-motifs, le but étant toujours d'éliminer un maximum de données en amont pour ne pas surcharger le processeur de la station hôte avec des calculs inutiles. Concrètement, le filtre contient un automate synthétisé à partir du motif à rechercher.

4.3. Recherche d'images par le contenu

La recherche sur les annotations d'une image est souvent insuffisante en particulier pour les images à contenu scientifique, et pour le cas qui nous intéresse, celles issues de l'étude du transcriptome et du protéome, parce qu'elles sont inadaptées ou décrivent l'image qualitativement. La recherche par le contenu sur des images passe par le calcul de descripteurs d'image qui la caractérisent. La recherche de proches voisins se fait alors sur ces descripteurs, plutôt que sur les images brutes, ce qui serait beaucoup trop complexe. Ces descripteurs forment donc un ensemble de vecteurs sur lesquels on va calculer une distance par rapport à ceux de l'image que l'on recherche. Ce calcul simple et répétitif peut être délocalisé dans nos filtres pour ne transmettre que les images pertinentes.

5. Conclusion

De nombreuses applications qui nécessitent des calculs intensifs sur des bases de données de très grande taille peuvent tirer parti de cette architecture, dès lors que le traitement se découpe en plusieurs étapes dont la première peut se faire de façon indépendante et qui consiste à filtrer une grande partie des données. Les exemples cités, étant des applications concrètes du monde de la génomique, nous permettront de valider notre approche. Nous en sommes actuellement aux spécifications de l'architecture et allons aborder la réalisation d'un prototype avec en premier lieu la construction de quelques modules (disque dur et circuit FPGA). Les étapes suivantes seront la conception des composants logiciels qui réguleront la machine et l'implémentation des filtres selon les applications.

Bibliographie

1. Stoesser (G.), Baker (W.), Van Den Broek (A.) et al. – The EMBL Nucleotide Sequence Database – *Nucleic Acids Research*, n.30, p.21–26, 2002.
2. Mosanya (E.), Sanchez(E.) – A FPGA-Based Hardware Implementation of Generalized Profile Search Using Online Arithmetic – *ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA '99*, p.101-111, 1999.
3. Keeton (K.), Patterson (D.), et Hellerstein (H.) – A Case for Intelligent Disks (IDISks) – *SIGMOD Record*, n.27(3), p.42–52, 1998.
4. Riedel (E.), Gibson (G.) et Faloutsos (C.) – Active Storage For Large-Scale Data Mining and Multimedia – *Proc. of 24th Very Large Data Bases Conf.*, 1998.
5. Acharya (A.), Uysal (M.), et Saltz (J.) – Active disks: programming model, algorithms and evaluation – *Proc. 8th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS-VIII)*, 1998.
6. Guerdoux-Jamet (P.), Lavenier (D.) – SAMBA: Hardware Accelerator for Biological Sequence Comparison – *CABIOS*, 13 (6), 1997.
7. Singh (R.K.) et al. – BioSCAN: A Dynamically Reconfigurable Systolic Array for Biosequence Analysis – *Proc. of CERCS96*, National Science Foundation, Arlington, VA, 1996.
8. Altschul (S.F.) et al. – Basic local alignment search tool – *Journal of Molecular Biology*, n.215, p.403–410, 1990.
9. TimeLogic Corporation, 22 State Route 28, Crystal Bay, NV 89402-0248 USA, www.timelogic.com
10. Paracel, 1055 East Colorado Boulevard, Pasadena, CA 91106-2341 USA, www.paracel.com/products/