HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Traitement de Données Génomiques

Cahier des Charges

Travail de Bachelor

Etudiant: Arthur Passuello

Professeurs: Yann Thoma
Romuald Mosqueron

Assistant: Rick Wertenbroek

15 juin 2018

Modifications

Version	Date	Description	Name
0.1	23.02.2018	Version Initiale	Passuello Arthur
1.0	28.02.2018	Version Initiale complètee	Arthur Passuello

Définitions

Abréviation	Terme	Définition
HMC	Hybrid Memory Cube	Mémoire constituée de couches empilées de mémoire de type DRAM.
BWT	Burrows-Wheeler Transform	Prétraitement des données voir ici
FM-Index	Full-text Minute-size Index	Principe de compression sans perte, Voir ici
REDS	Reconfigurable Embedded Digital System	Institut de prestige de l'HEIG
PEHAGA	Power-Efficient Hardware Acceleration of Genomic Algorithms	Institut de recherche dans l'exploration de solutions matérielle à l'accélération du traitement de données génomiques

Table des matières

1 Situation de Départ

1.1 Description

Le séquençage de génome se répand à grande vitesse, et actuellement un séquençage produit 300 GB de données désordonnées qui nécessitent un traitement particulier afin d'optimiser leur utilisation (compression..) ainsi que leur interprétation.

Plusieurs challenges s'offrent à la communauté scientifique : comment compresser ces données, comment les taiter efficaement, comment les transmettres, ... La taille des données rend tout algorithme de traitement de complexité linéaire beaucoup trop coûteux, de même qu'une utilisation naïve de la mémoire, linéaire selon la taille des données, est beaucoup trop importante. Il parait donc primordial d'explosrer des solutions permettant d'optimiser le temps et l'espace utilisés dans le traitement des données génomiques, non seulement par des algorithmes efficaces mais aussi une grande parallélisation.

Le *mapping* de séquence, très fastidieux selon une approcha naïve, offre l'espace pour de grandes améliorations. N'ayant besoin d'accéder aux données qu'en lecture pour générer un résultat, le processus est très parallélisable. De même qu'un encodage adapté de la séquence de référence peut permettre de grands gains en temps et en espace nécessaire, aussi bien sur l'indexage lui-même qu'au moment du *mapping*. C'est dans ce contexte que se situe ce projet, visant à concevoir un système d'indexage et de *mapping* mettant à profit les avantages offerts par les FPGA pour arriver efficacement à un résultat sur de grands ensembles de données.

Ce projet s'inscrit dans un projet actuel auquel participe l'institut REDS : PEHAGA. Le travail de l'étudiant visera à développer un système permettant d'accélérer le *mapping* de séquences par FPGA.

1.2 Matériel Utilisé

L'implémentation du système devra en partie être réalisé sur un système à processeur (pré-traitement de la séquence de référence, interface de communication), l'algorithme de *mapping* en revanche devra se faire sur FPGA, avec la réalisation d'un système complet allant de l'envoi des données à la FPGA jusqu'à la récupération des résultats.

La plateforme matérielle utilisée sera une carte EX-700 [1] comportant 6 modules AC-510 [2] aavec chacun une FPGA et une mémoire HMC (lien séries à haut débit) de 4GB.

1. http ://picocomputing.com/products/backplanes/ex-700/
2. AC-510 de *PicoComputing*.

2 Objectifs

2.1 Requis

Num	Libellé	Description	Critère de Mesure
[1]	Maitrise de la BWT et du FM-Index	Compréhension des concepts et algorithmes utilisés	Rapport théorique
[2]	Implémentation d'un programme d'indexage	Écrire un programme dans un langage quelconque prenant en entrée un fichier FASTA/Q pour l'indexer selon la BWT et le FM-Index et fournir en sortie une structure de donnée binaire indexée	Test de vérification
[3]	Programme de *mapping* exact	Implémenter un programme de *mapping* d'une petite séquence (format FASTA/Q) relativement à une grande dans un langage quelconque, servant de référence pour la suite	Test de vérification
[4a]	Système de *mapping* exact sur FPGA	Implémenter en VHDL un système permettant de *mapper* une collection de séquences fournie en entrée selon une séquence de référence en mémoire	Test de vérification Test de performance Vérification formelle
[4b]	Choix du nombre de *mapping* retournés	Implémenter la possibilité de choisisr combien de *mapping* au maximum doivent être retournés par séquence	
[5]	Interface de communication	Implémenter dans un langage de haut niveau un programme permettant de communiquer avec la carte	Test de vérification
[6]	Intégration du système	Intégration su système précédent sur la carte fournie	Test scripté et manuel

2.2 Objectifs Optionnels

Num	Libellé	Description	Critère de Mesure
[7]	*Mapping* avec erreur	Implémenter en VHDL un système permettant d'ibtenir un *mapping* inexact dans le cas où un *mapping* exact n'existerait pas pour la séquence courante	
[8]	*Mapping* de paires de séquences	Implémenter en VHDL un système permettant de *mapper* une paire de séquences fournie en entrée selon une séquence de référence en mémoire	Test de vérification Test de performance Vérification formelle

2.3 Délimitation / hors-périmètre

Le projet ne couvre pas :

— La compression

— Les performances du programme d'indexage (**[2]**)

3 Description de la Solution

3.1 Software

Le programme d'indexage **[2]** devra au moins offrir un interface console, prenant en entrée un fichier ou un ensemble de fichier au format FASTA ou FASTQ en donner en sortie sa ou leur version indexée dans un format binaire à définir.

Le programme de *mapping* exact **[3]** devra au moins offrir un interface console, prenant en entrée un fichier ou un ensemble de fichier au format FASTA ou FASTQ en donner en sortie les positions des *mapping* et la séquence donnée dans un format à choisir/définir (e.g. SAM).

Le programme d'interface avec la carte **[5]** devra au moins offrir un interface console. Il devra être possible de fournir dans un premier temps la séquence de référence dans un format binaire puis, dans un second temps, fournir en entrée un ensemble de séquences à *mapper*, en recevant en répose, s'il(s) existe(nt), le ou les positions des séquences dans la référence.
Les commances seront envoyées sur un bus mémoire, tandis que les données seront envoyées et reçues en parallèle par un bus PCI Express.

3.2 Hardware

Le système écrit en VHDL **[4]** devra être en mesure de charger une séquence de référence (indexée) prise en entrée en mémoire, puis de prendre d'autres séquences en entrée et de retourner, s'il(s) existe(nt), le ou les index de début des *mapping* de la séquence trouvé(s) dans la référence ainsi qu'un identificateur pour la séquence en cours. Les entrées et sorties de séquences/*mapping* se feront par un *stream* sur un bus PCI Express, l'envoi de commandes depuis le PC sera en revanche fait par un bus mémoire.

Ce système devra ensuite être inclus dans un projet pouvant être chargé sur la carte fournie **[6]**. Il devra être possible, depuis un programme sur ordinateur, de recevoir plusieurs type de données (séquence de référence, séquence à mapper) et d'envoyer en retour un résultat interprétable par celle-ci.

4 Planification

Task name	Start date	End date	Duration (week)
Total Estimate	*21/02/2018*	*27/07/2018*	*22.34*
Software Development	**21/02/2018**	08/07/2018	19.63
BWT & FM-Index Implementation	21/02/2018	07/04/2018	6.57
Documentation	21/02/2018	09/03/2018	2.43
Naive Implementation	10/03/2018	19/03/2018	1.43
Optimization	20/03/2018	02/04/2018	1.75
Documentation Solution	03/04/2018	07/04/2018	0.71
+ Add a task + Add a milestone			
Interface Implementation	30/06/2018	08/07/2018	1.14
Desktop Interface Implementati	30/06/2018	05/07/2018	0.71
Debugging / Testing	05/07/2018	08/07/2018	0.43
+ Add a task + Add a milestone			
+ Add a task + Add a milestone			
Hardware Development	**08/04/2018**	27/07/2018	15.77
Documentation	08/04/2018	14/04/2018	0.91
Design	14/04/2018	28/04/2018	2
Naive Implementation	28/04/2018	12/05/2018	2
Testing / Debugging	12/05/2018	26/05/2018	2
On-Board Porting / Testing	26/05/2018	09/06/2018	2
Optimization	09/06/2018	23/06/2018	2
Testing / Debugging [Full Time]	23/06/2018	30/06/2018	1
Testing / Debugging [Full Time]	08/07/2018	15/07/2018	1
Finalisation [Full Time]	15/07/2018	27/07/2018	1.71

5 Validation

15 juin 2018

Arthur Passuello
Étudiant

Yann Thoma
Professeur Responsable

Romual Mosqueron
Professeur Responsable