

TER - Rendu 1

Emile Cadorel, Guillaume Gas, Jimmy Furet, Valentin Bouziat

20 mars 2016

Chapter 1

Etude des squelettes existants

Afin de mieux comprendre le but de notre travail, nous avons fait des recherches sur la programmation de squelettes. Nous nous sommes ainsi intéressés à deux bibliothèques qui semblent reprendre le concept qui nous intéresse : Thrust et SkePU.

1.1 Thrust

Cette bibliothèque de templates C++ pour Cuda est basée sur la STL (Standard Template Library). Elle permet l'utilisation d'algorithmes parallèles avec un minimum de code. Ainsi, on retrouve les algorithmes suivants :

- Transformation : applique une opération sur un set de données et met le résultat dans un set destination.
- Reduction : utilise une opération binaire pour réduire une entrée en une unique valeur.
- Prefix-Sums : ou opération `scan` permet par exemple, sur un tableau en entrée, d'obtenir en sortie un tableau avec les sommets cumulés.
- Reordering : permet d'arranger ou partitionner des données selon des prédicats de test.
- Sorting : différents algorithmes de tri.

1.2 SkePU

Bibliothèque de templates C++ mettant à disposition un ensemble de squelettes génériques destinés à faciliter la création de code exécuté en parallèle. On y retrouve des fonctions similaires à Thrust.

- Map
- MapReduce
- MapOverlap
- MapArray
- Generate
- Scan

Chapter 2

Comment intégrer ces notions dans Spoc ?

2.1 Modification de l'arbre

blablabla

2.2 Génération de code Sarek

blabla