Introduction et vue d'ensemble

Samuel Thibault

Université de Bordeaux

Inria Bordeaux Sud-Ouest

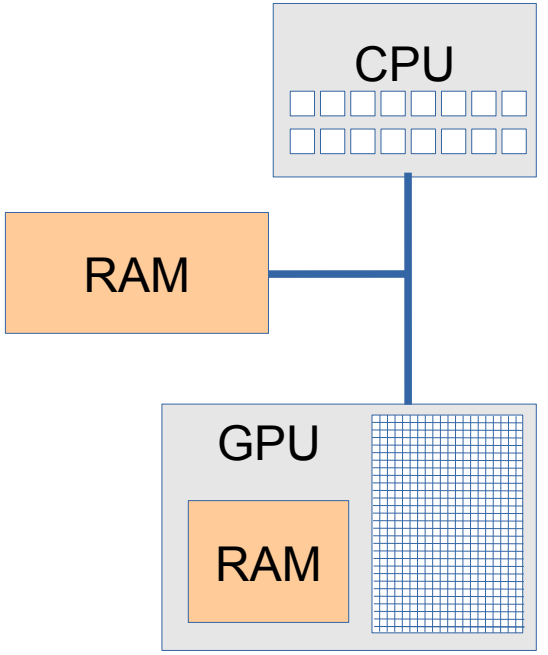
Objectifs

Programmation GPU

- Explorer ce qui est possible maintenant
- Collecter les besoins pour l’appel à projets GPU

Les GPUs

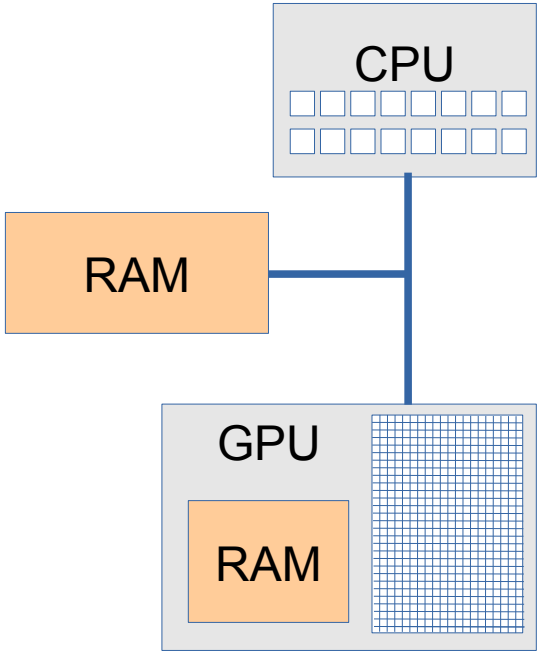
- Programmation particulière
  - Parallélisme massif !
  - Mémoire séparée (la plupart du temps)
- Développement particulier
  - Compilation différente
  - Lancement par offload logiciel
    - On ne peut pas juste recompiler
  - Outils de débuggage différents



Les GPUs

Principe :

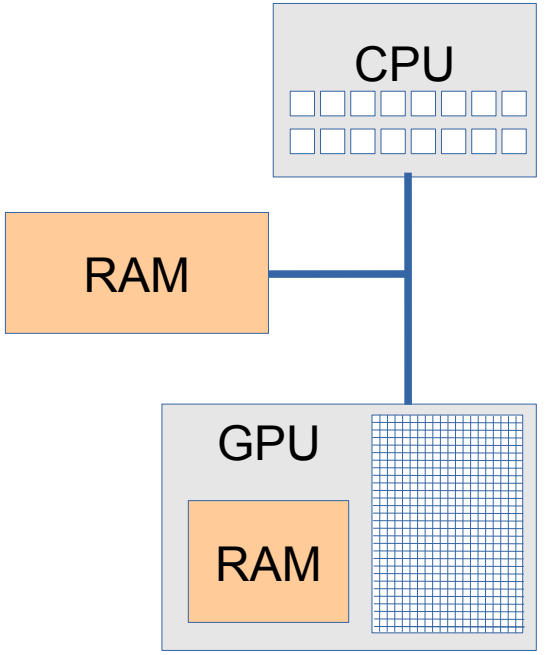
- Transfert de données
- Kernels de calcul
- Orchestration



Les GPUs

Porter une application

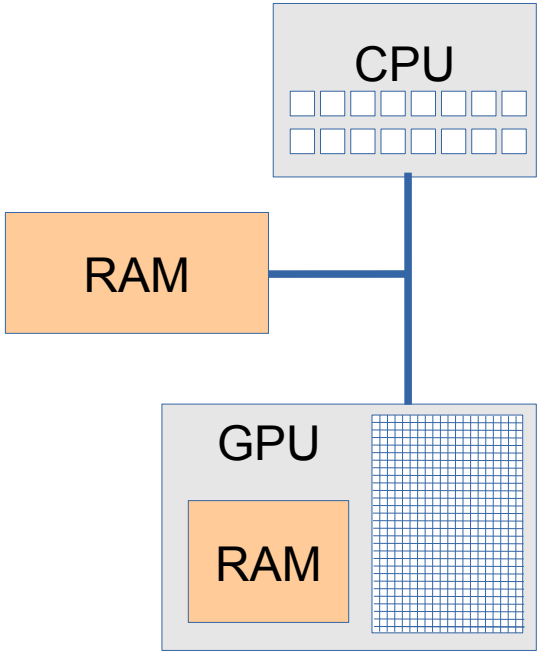
- Porter des portions de calcul pour GPU
  - Exhiber un parallélisme massif
- Intégrer avec le reste du code



Les GPUs

Porter une application

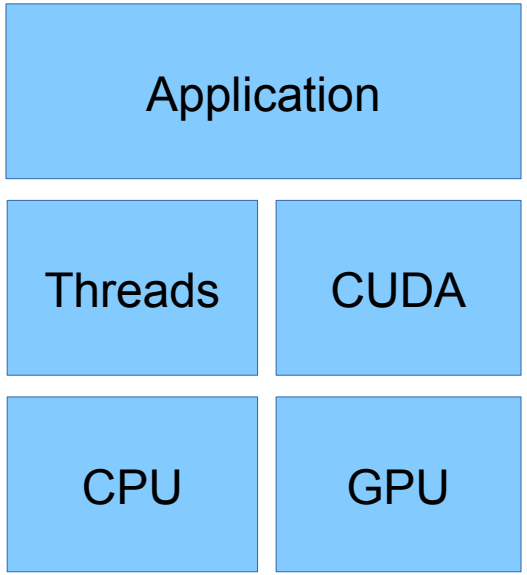
- Plusieurs approches
  - CUDA / HIP / OpenCL / …
  - Framework
  - Bibliothèques
  - Langage
  - Tâches
- Selon faisabilité
- Selon niveau d’investissement



Les GPUs

Approche CUDA / HIP / OpenCL / ...

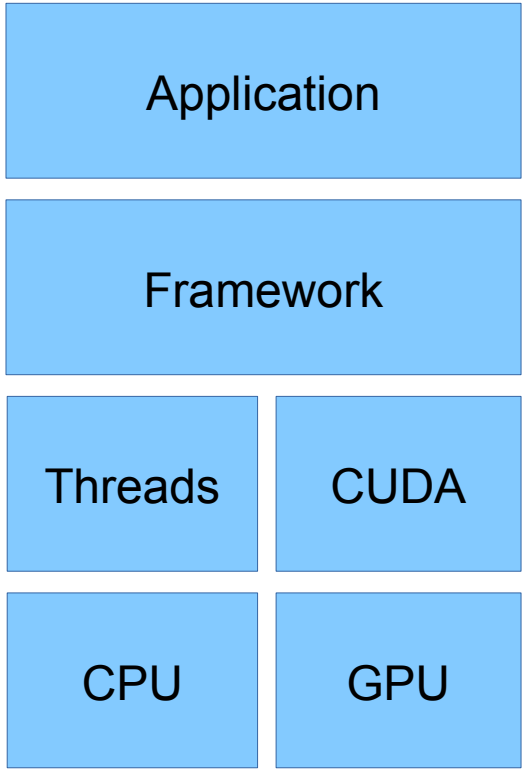
- Porter les kernels
- Orchestrer
- Portabilité ?



Les GPUs

Approche Framework

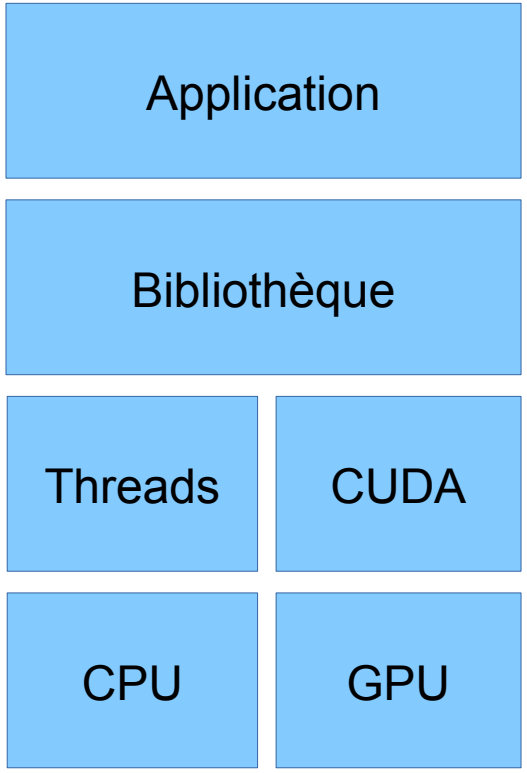
- Déléguer le travail au Framework
- Lorsque le domaine applicatif y correspond
  - Reste « juste » à transcrire les équations
- Exemple : Arcane, CEA



Les GPUs

Approche bibliothèque

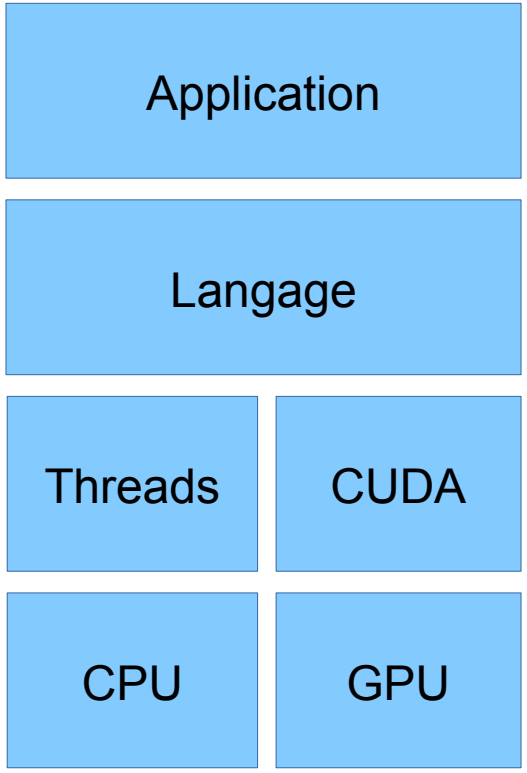
- Déléguer le travail à la bibliothèque
- Lorsque le besoin calculatoire est identifié
- Exemple : algèbre linéaire



Les GPUs

Approche langage

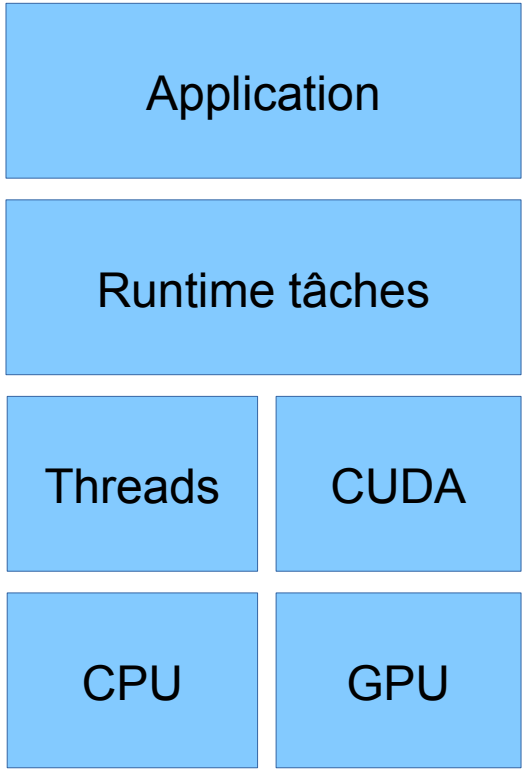
- Déléguer le travail au compilateur et au runtime du langage
- Pouvoir d'expression plus universel
  - Non lié à un domaine ou un besoin
- Exemples : OpenMP, OpenACC, Kokkos, SYCL



Les GPUs

Approche tâches

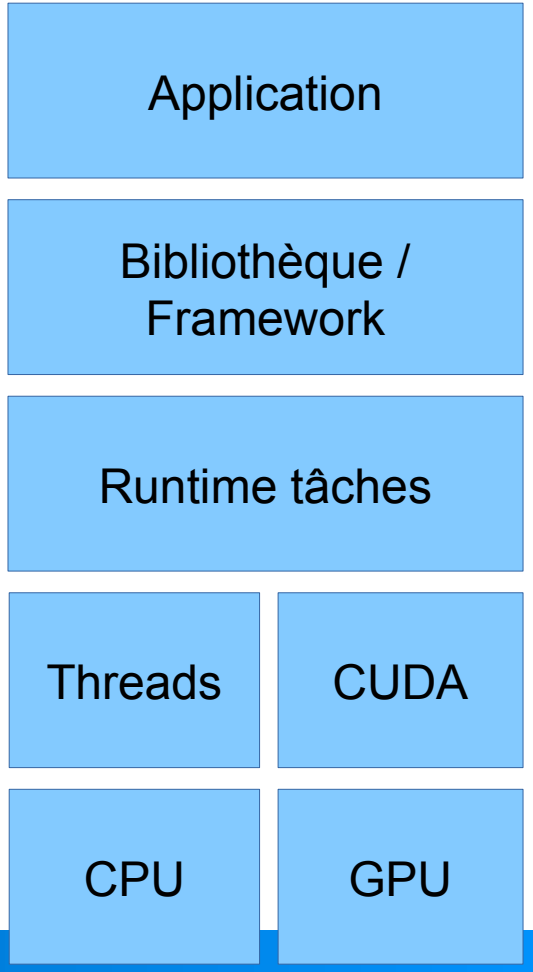
- Déléguer le travail au runtime de tâches
- Pouvoir d'expression universel
  - Non lié à un domaine ou un besoin
- Nécessite un portage des kernels
- Exemples : StarPU



Les GPUs

Approche combinée

- Profiter des différentes couches
  - *Separation of concern*
- Exemples : Chameleon + StarPU



Les GPUs

Exposés (1)

- Les GPUs, CUDA
- Approche framework, Arcane
- Approche bibliothèque, Chameleon
- Approche langage, OpenMP / OpenACC / Kokkos / SYCL
- Approche tâches, StarPU

Les GPUs

Exposés (2)

- Retex tâches pour l'algèbre linéaire
- Retex OpenACC pour plateforme multiphysique (YALE2)
- Retex OpenACC pour simulation turbulent reacting flows (AVBP)
- Retex Kokkos pour hydrodynamique MR (Dyablo)
- Retex Rust + OpenCL pour magnetohydrodynamics (minicl)

Les GPUs

Exposés (3)

- Contributions PEPR NumPEx
- Brainstorm pour l’appel à projets GPU

Les exposés

- Enregistrés
- Présentation d’approche: 15’ + 5’
- Présentation de Retex: 15’ + 15’
  - Beaucoup de temps pour creuser l’expérience acquise