

Modèles et techniques en programmation parallèle hybride et multicœurs

Programmation hybride MPI – Cuda ou OpenCL

Marc Tajchman

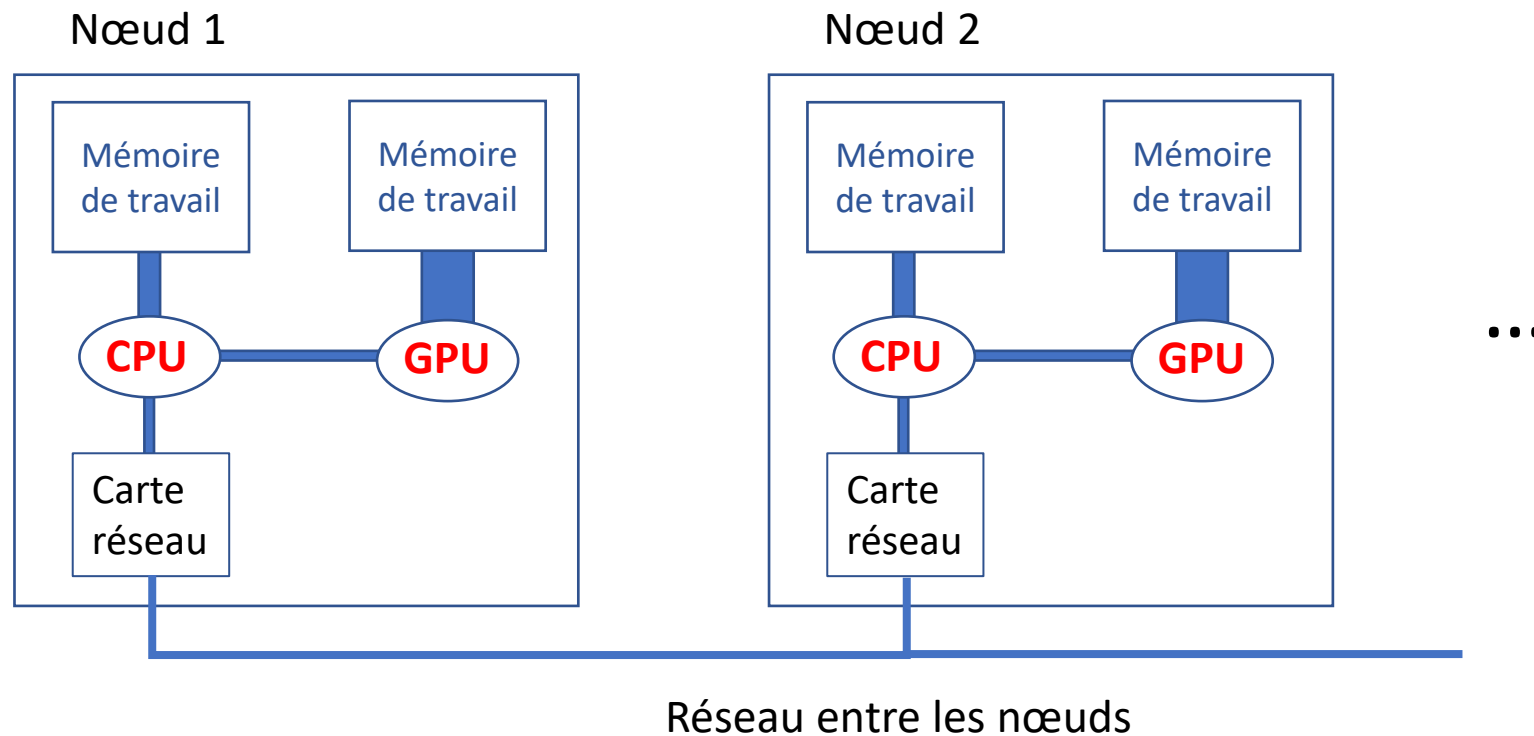
CEA - DEN/DM2S/STMF/LMES

01/02/2021

Environnement matériel:

On considère une machine parallèle avec n nœuds, chacun contenant un (ou plusieurs) CPU et un (ou plusieurs) GPU.

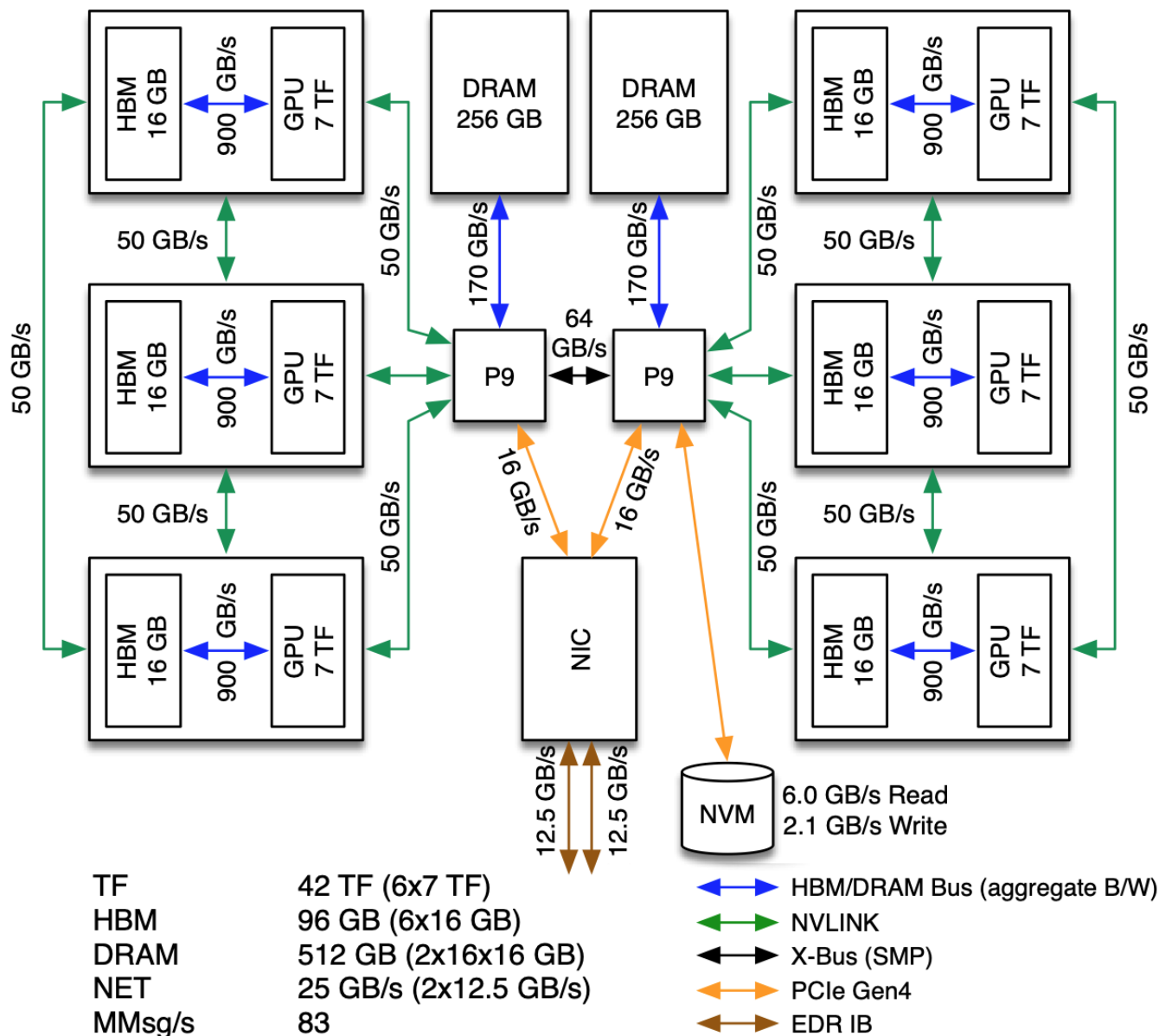
Dans chaque nœud, le(s) CPU et le(s) GPU sont associés à leur mémoire de travail propre.



Exemple : machine Summit
(Oak Ridge Nat. Lab, USA)

4806 nœuds chacun avec 2 CPU
(IBM Power PC) + 6 GPU(Nvidia
Volta)

Schéma de l'intérieur d'un nœud:



Sur ce type de machine, on a la possibilité de

- Créer un (ou plusieurs) processus MPI dans les nœuds utilisés
- Dans chaque processus MPI, créer plusieurs threads (si le processus MPI contrôle plusieurs cœurs de CPU)
- Lancer des calculs sur la (ou les cartes) graphiques contrôlées par le processus MPI

Dans cet exposé, on supposera qu'on lance un processus MPI par nœud utilisé et que chaque nœud contient un GPU

La programmation combinée MPI - Cuda (ou OpenCL) n'est pas compliquée si on garde à l'esprit quelques règles

- C'est le CPU qui fait les appels MPI (pas d'appels MPI dans les noyaux Cuda ou OpenCL)
 - MPI permet d'échanger des données entre les mémoires de travail des CPU
- Si on a besoin d'envoyer une donnée de la mémoire d'un GPU à un autre GPU sur un autre nœud, il faut:
 1. Dans le nœud de départ, copier la donnée de la mémoire du GPU sur la mémoire du CPU sur le même nœud
 2. Envoyer la donnée sur la mémoire du CPU du nœud d'arrivée
 3. Dans le nœud d'arrivée, copier la donnée de la mémoire du CPU vers la mémoire du GPU sur le même nœud
- Faire attention aux barrières : dans MPI (`MPI_Barrier`), dans CUDA (`cudaDeviceSynchronize`)

Remarque: dans les versions récentes de MPI, Cuda et OpenCL, il existe certaines simplifications de ces opérations. Mais cela demande des versions de MPI spécialement compilées et des GPU compatibles. On ne se placera pas dans ce cas ici.

Exemple de programmation hybride MPI – Cuda : Exemple9.tar.gz