

PRD "parallélisation"

Idée: Sur un programme donné, mise en oeuvre
multithread OpenMP
multicore GPU / Cuda
multiprocessor TPL.

(

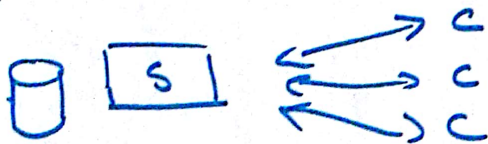
	0	1	2
0	1	2	3
1	1	2	1
2	2	3	1

 si 0 est précédé de 1 puis 2
→ Quelle est la séquence la plus courte ?)

Idée: évaluer toutes les combinaisons (n)!

① . Version de base
* Implicitement, on suppose que toutes les machines sont disponibles.

② . Version adaptable
* Un processus "serveur" gère la file d'attente des séquences à évaluer et les distribue par lots.



Rmq: Reprise en cas d'échec d'un client?
Reprise en cas d'échec du serveur?

↓
plusieurs stratégies possibles:

- * à la demande
- * anticipative. (mécanisme de reporting?)
- * taille des lots?

③ Vers une API pour l'aide au déploiement parallèle des Recherches exhaustives (ou quasi).

- Approche scientifique rigoureuse

- * solution parallèle identique à solution //e
- * les n^{ts} programmes sont comparés sur un ensemble de fichiers d'instances fixe