

A person in silhouette stands before a city skyline at night, with a network of glowing nodes and lines overlaid on the scene.

6GEN723 – Réseaux d'ordinateurs

Séance #1 – 12 Janvier 2023 (*Séance Virtuelle*)



Programmation parallèle

Chapitre 0

Pourquoi un chapitre 0 ?

3 raisons principales:

- 1) Un complément pertinent suite à des cours que certains ont déjà suivis**
- 2) Excellent exercice pour se familiariser avec les protocoles de communication qui est l'objectif du présent cours**
- 3) Permet de mettre la table pour la suite du cours et donne également l'occasion de se dérouiller un peu pour des développements non triviaux**

01 Contenu

Qu'est-ce que le calcul/traitement parallèle ?
Domaines d'application
Est-ce toujours un choix judicieux/pertinent ?

02 Contenu

Diversité des systèmes parallèles

03 Contenu

Les modèles de programmation

04 Contenu

MPI – Message Passing Interface
Exemples simples de programme

Structure du contenu

01

Qu'est-ce que le calcul ou le traitement parallèle ?

- <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial##Whatis>

01

Domaines d'application

- <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial##Who>

01 Est-ce toujours un choix judicieux ou pertinent ?

- Est-ce que quelqu'un se rappelle de M. Gene Amdahl ?
- La loi d'Amdahl
 - Quelqu'un s'en rappelle ?

Loi d'Amdahl

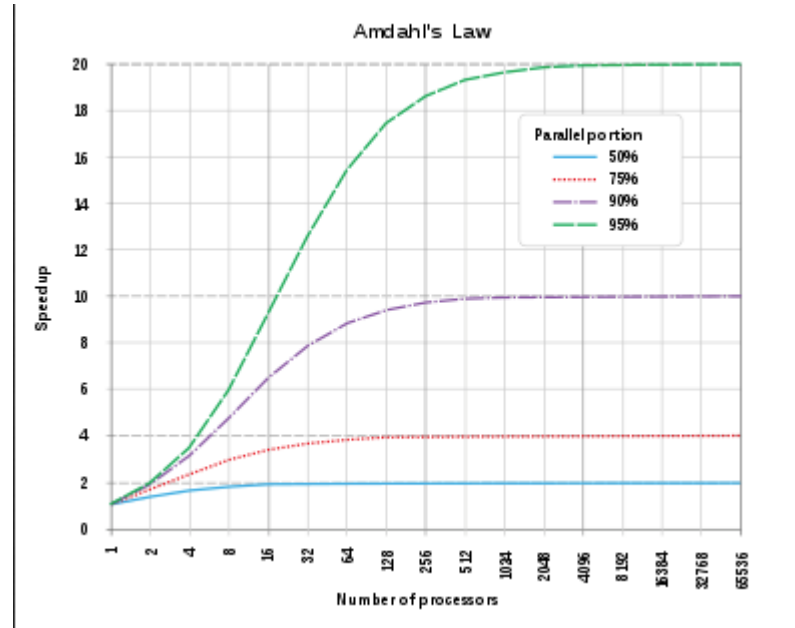
Rappel rapide de la loi d'Amdahl

- Soit une tâche requérant un temps T pour être exécutée sur un système uni processeur.
- Supposons qu'une fraction X (en pourcentage %) de ce temps puisse être exécuté en parallèle par N processeurs. L'autre partie ne pouvant être exécutée que par un seul processeur*.
 - *Les facteurs sont multiples pouvant occasionner cette contrainte: une gestion très lourde, ...*
- Le gain de performance S sera défini comme suit:

$$x_{latence}(S) = \frac{(\text{temp d'exécution séquentielle})}{(\text{temps d'exécution parallèle})} = \frac{T}{(T-xT) + \frac{xT}{N}} = \frac{1}{(1-x) + \frac{x}{N}}$$

Loi d'Amdahl

- $x_{latence}(s) \leq \frac{1}{1-x}$
- $\lim_{N \rightarrow \infty} x_{latence}(s) = \frac{1}{1-x}$
- *Graphique*



Référence: <https://upload.wikimedia.org/wikipedia/commons/thumb/e/ea/AmdahlsLaw.svg/400px-AmdahlsLaw.svg.png>

01

Loi d'Amdahl

➤ *Le gain de performance est intéressant seulement si $\rightarrow X > 0.9$*

Note: On suppose que le programme ne change pas lorsque N augmente ! Ce qui n'est pas toujours vrai!

→ Calculateur en ligne de la loi d'Amdahl: <https://www.vcalc.com/wiki/vCalc/Amdahl%27s+Law>

Diversité des systèmes parallèles

Existe-t-il plusieurs types de systèmes parallèles ?

- Il en existe en fait quelques grandes classes dont 2 principales
- SIMD et MIMD
 - SIMD - Single Instruction Multiple Data
 - MIMD - Multiple Instruction Multiple Data
- Il existe d'autre(s) modèle(s) « *plus théorique* » ou du moins qui n'ont pas nécessairement vu le jour
 - *MISD - Multiple Instruction Single Data*
- <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial##Flynn>
(Porter une attention aux différents systèmes commerciaux sous chacune des catégories)

02

Diversité des systèmes parallèles

Quel est le système SIMD le plus* connu? Du moins largement utilisé et fort potentiellement par vous...!

- Carte graphique de votre ordinateur

<https://images.anandtech.com/doci/5840/GK110Block.png>

→ *Chaque processeur prend en charge un groupe de pixels*

Diversité des systèmes parallèles

- Les systèmes SIMD sont cependant beaucoup moins « flexibles » que les systèmes MIMD.
 - Pourquoi ?
 - Tous les processeurs exécutent ***le même programme***
- Les systèmes parallèles MIMD sont plus versatiles.
 - Exemple: Superordinateur d'Environnement Canada (*Dorval*) – Acheté en 2017
 - IBM: 28440 cœurs (*430 millions \$*)
- La question: Comment programme-t-on des systèmes MIMD ?

Les modèles de programmation

Plusieurs « méthodes » ont été développées pour les programmer ces ordinateurs « parallèles »

(→ Pour votre information seulement)

- *Programmation utilisant une mémoire partagée (sans « threads »)*
- *Programmation utilisant des « threads »*
- *Programmation utilisant une mémoire distribuée*
- *Programmation utilisant des données parallèles*
- *Combinaison de modèle → approche hybride*

Vous pouvez consulter le lien suivant pour plus de détails:

<https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial#Models>

Les modèles de programmation

Avec le temps, on a élaboré deux approches *plus générales* qui peuvent s'appliquer à tous les modèles précédents (pour les systèmes MIMD):

- « **Single Program Multiple Data** » (**SPMD**)
→ *Ne signifie pas qu'il s'agit de programmes pour les systèmes SIMD!*
- « **Multiple Programs Multiple Data** » (**MPMD**)

Pour nous permettre de nous familiariser avec la programmation parallèle MIMD, nous allons élaborer des programmes pouvant être exécutés par des systèmes à mémoire distribuée (« *message passing* »)

- Très similaire aux échanges effectués dans les protocoles de communication standards

Les modèles de programmation

Connaissez-vous un système à mémoire distribuée ?

→ Un réseau d'ordinateurs !

Exemple: le local P2-4020

- Tout réseau d'ordinateurs (*ordinateurs interconnectés*) est donc un exemple de système à mémoire distribuée.
- Toute infrastructure d'équipements interconnectés (*Serveurs, ordinateurs, équipements électroniques permettant d'effectuer du traitement possédant de la mémoire, etc...*) fait également partie de cette catégorie
- Les « grappes de serveurs » (*clusters*) → lien
- ...

Les modèles de programmation

Comment les « systèmes à mémoire distribuées » opèrent-ils, fonctionnent-ils ?

- <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial#ModelsMessage>

Pour faire travailler ensemble plusieurs ordinateurs d'un réseau ou d'une architecture on utilise en générale une API (*Application Programming Interface*)

- Un API a particulièrement marqué son époque (1989 on ne rajeuni pas...)
 - PVM – Parallel Virtual Machine
 - Petite référence wiki - https://en.wikipedia.org/wiki/Parallel_Virtual_Machine
 - Site officiel - <https://www.csm.ornl.gov/pvm/>

MPI – Message Passing Interface

PVM a été supplanté par **MPI**

- Il existe plusieurs implémentations de MPI
 - MPICH
 - Open MPI
 - Microsoft MPI (*Ms-MPI*)
 - Etc...

04

MPI – Message Passing Interface

Dans le laboratoire #1 vous utiliserez **Ms-MPI**

- Toute la documentation concernant les fonctions du « standard » MPI sont disponible sur Internet:
 - <https://www.mpich.org/static/docs/latest/>
 - <https://www.open-mpi.org/doc/v1.4/>



Période de questions

Des questions?

Références

- <https://www.mpi-forum.org/docs/>
- <https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf>
- <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial>
- ...