

Contents

0.1	Modèles de performance	1
0.2	Résolution de l'équation de Laplace	1
0.3	Scalabilité	2

0.1 Modèles de performance

- Sommer n nombres sur p PE en mémoire distribuée

0.2 Résolution de l'équation de Laplace

(Équation de la chaleur)

$\Delta^2 \phi = 0$ sur le domaine de calcul et ϕ a une valeur donnée sur les bords du domaine.

Schéma numérique: discrétiser le domaine de calcul avec un maillage.

On montre qu'on résoud $\Delta^2 \phi$ itérativement avec la relation:

$$\phi^{k+1}(i, j, k) = \frac{1}{6}(\phi^{(k)}(i-1, j, k) + \phi^{(k)}(i+1, j, k) + \phi^{(k)}(\dots, j-i \dots j+1) + (\text{idem avec } k \dots))$$

Parallélisation: on découpe le domaine en sous-domaine, associé à chacun des PE.

Le maillage contient N^3 point (i, j, k) où calculer ϕ .

L'idée est de séparer le domaine en sous-domaine, donné à chacun des processeurs.

On a p PE. Donc: $N^3 = p \times l^3$

$T_{par} = 6 \times l^3 \times T_{cpu}$ (6 opérations sur l^3 points fois le temps d'exécution de chaque opération (calcul utile)) $+ 6 \times l^2 \times T_{comm}$ (6 faces -> 6 messages, un par face... , nombre de points par face l^2 , temps de communication(overhead)) $+ 6\mu s$ (temps de latence pour chacun des 6 messages. On le néglige pour autant que l soit assez grand. On a donc:

$$T_{par} = 6 \times l^3 \times T_{cpu} + 6 \times l^2 \times T_{comm}$$

$$T_{seq} = 6N^3 T_{cpu}$$

$$\begin{aligned} S &= \frac{T_{seq}}{T_{par}} = \frac{6N^3}{6l^3 T_{cpu} + 6l^2 T_{comm}} = \frac{N^3}{l^3 + l^2 \frac{T_{comm}}{T_{cpu}}} \\ &= \frac{N^3 / l^3}{1 + \frac{T_{comm}}{T_{cpu}} \frac{1}{l}} = \frac{p}{1 + \frac{T_{comm}}{T_{cpu}} \frac{1}{l}} \approx p \text{ si } l \rightarrow \inf \end{aligned}$$

$$\frac{overhead}{calcul - utile} = \frac{6l^2 T_{comm}}{6l^3 T_{cpu}} = \frac{T_{com}}{T_{cpu}} \times \frac{1}{l}$$

Donc il faut que le problème soit suffisamment grand pour que $\frac{1}{l}$ soit suffisamment petit.

Donc ce qui intervient dans la déviation du speedup idéal, $\frac{T_{com}}{T_{cpu}} \times \frac{1}{l}$ est proportionnel au rapport $\frac{surface}{volume}$ du sous-domaine.

Cela indique que la forme du sous-domaine impacte notre calcul.

La forme géométrique qui minimise le rapport $\frac{surface}{volume}$ est une sphère.

Donc le cube est un assez bon choix pour ce rapport.

Sous-domaine parallépipède rectangle peu recommandé: grande surface, petit volume.

$\frac{1}{l} \dots l^3 = \frac{N^3}{p} \Leftrightarrow l = \frac{N}{p^{\frac{1}{3}}} \rightarrow$ loi d'Amdahl si N est constant (strong scaling), loi de Gustafson si l est constant (weak scaling)

0.3 Scalabilité

Il y a une notion de scalabilité appliquée au matériel: est-ce qu'on peut augmenter le nombre de PE et que tout augmente proportionnellement: perf, coût, infrastructure, etc... (**scalabilité d'architecture**)

Ici, on va considérer un autre concept: la **scalabilité dite d'application**.

C'est une notion reliée à l'algorithme utilisé et à l'architecture considérée. (exemple: mémoire distribuée)

Ce concept est basé sur la notion d'**isoefficacité**, ainsi que la taille du problème et le nombre de processeurs.

$n = f_E(p)$ fonction d'isoefficacité. Comment la taille du problème, n, doit croître quand on augmente le nombre de PE, p, de sorte à garder l'efficacité constante:
 $E = \frac{S}{p}$