Produit matrice–vecteur parallèle

Xavier Juvigny

10 mars 2020

Le but de ce document est de présenter deux algorithmes permettant d'effectuer le produit d'une matrice carrée A de dimension n par un vecteur u en parallèle sur N processus :

$$v = A.u$$

On suppose que la dimension de la matrice est divisible par le nombre de processus pour simplifier la présentation des deux algorithmes.

1 Répartition par ligne

La matrice A de dimension n est distribuée par ligne, un processus p contenant les $n_{\text{loc}} = \frac{n}{nbp}$ lignes allant de $p * n_{\text{loc}}$ (compris) à $(p+1) * n_{\text{loc}}$ (non compris).

Le vecteur u est quant à lui répliqué sur chacun des processus.

En décomposant la matrice par ligne, le produit matrice–vecteur peut s'écrire de façon algébrique comme suit (où $A_{i\star}$ est le $i^{\text{ème}}$ bloc–ligne de la matrice) :

$$A.u = \begin{pmatrix} A_{1\star} \\ A_{2\star} \\ \vdots \\ A_{N\star} \end{pmatrix} .u = \begin{pmatrix} A_{1\star}.u \\ A_{2\star}.u \\ \vdots \\ A_{N\star}.u \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} = v$$

Chaque processus calcule un $A_{p\star}.u = v_p$, c'est à dire un bloc de v de dimension n_{loc}. A la fin de l'exécution, le vecteur v se retrouve donc distribué sur les N processus.

Pour retrouver (dans le cadre d'un processus itératif par exemple), un vecteur v dupliqué sur chacun des processus, il faut reassembler le vecteur v à partir des v_i, avec un `Allgather` par exemple.

2 Répartition par colonne

La matrice A de dimension n est distribuée par colonne, un processus p contenant les $n_{\text{loc}} = \frac{n}{nbp}$ colonnes allant de $p * n_{\text{loc}}$ (compris) à $(p+1) * n_{\text{loc}}$ (non compris).

Le vecteur u est distribué également en bloc u_i de dimension n_{loc}.

La décomposition de la matrice A par bloc colonne et de u par bloc, conduit à formuler le produit matrice–vecteur de la manière suivante (où $A_{\star i}$ représente le $i^{\text{ème}}$ bloc colonne) :

$$A.u = \begin{pmatrix} A_{\star 1} & A_{\star 2} & \dots & A_{\star N} \end{pmatrix} . \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_N \end{pmatrix} = \sum_{i=1}^{N} A_{\star i}.u_i = \sum_{i=1}^{N} v_i = v$$

Ainsi chaque processus calcule un $v_i = A_{\star i}.u_i$ de dimension n. Il suffit alors à l'aide d'une réduction (avec l'opération de sommation) de calculer le vecteur v final (`Reduce` ou `Allreduce`).

Si le produit est utilisé dans un algorithme itératif, il suffit alors de n'extraire de v que le bloc concerné par le processus courant.