

T.P. Parallélisme : PVM - Elimination de Gauss

Compte-rendu présentant analyses, algorithmes, résultats et listings dans ma boîte en 5251 pr le 25/10

1 Environnement PVM, programmation SPMD

1. Configurer ssh sur votre compte de façon à ne pas avoir à taper votre mot de passe lors d'un login d'une machine sur une autre (voir <http://www.esiee.fr/~perrotol/ssh-guide.html>)
2. Rajouter "source /user/pvm/pvm/cshrc.pvm" à la fin de votre .cshrc et relancer un nouveau shell pour que ce soit pris en compte.
3. Créez les répertoires \$HOME/pvm3/bin/LINUX64/ pour les binaires et placez vos sources dans \$HOME/pvm3/.
4. Copier depuis /user/pvm/pvm3/ les sources hello.c, tokenring-sibling.c, gauss.c et Makefile dans ~/pvm3. Compiler (make tokenring-sibling) et vérifier que pvm et xpvm fonctionnent correctement en les exécutant.

2 Elimination de Gauss

Le but du TP est de proposer une parallélisation de l'élimination de Gauss avec PVM. L'élimination de Gauss est un algorithme de triangularisation de matrice qui peut être utilisé pour trouver les solutions d'un système d'équations linéaires du type $A.X = B$ où A est une matrice $n \times n$ et X et B deux vecteurs de taille n . L'algorithme consiste à annuler la partie triangulaire inférieure de la matrice en itérant $n - 1$ étapes ($k = 0, \dots, n - 2$) qui annulent les éléments $i = k + 1, \dots, n - 1$ de la colonne k de la matrice par substitution de la ligne k avec les lignes $i = k + 1, \dots, n - 1$.

$$A = \begin{pmatrix} a_{00} & \dots & \dots & \dots & a_{0n-1} \\ & \ddots & & & \vdots \\ & & a_{kk} & \dots & a_{kj} & \dots & a_{kn-1} \\ \vdots & & \vdots & & \vdots & & \vdots \\ & & a_{ik} & \dots & a_{ij} & & \vdots \\ & & \vdots & & & & \vdots \\ a_{n-10} & \dots & a_{n-1k} & \dots & \dots & \dots & a_{n-1n-1} \end{pmatrix}$$

```
FOR k = 0 TO n-2
  FOR i = (k+1) TO n-1
    FOR j = (k+1) TO n-1
      a[i][j] = a[i][j] - a[i][k]/a[k][k] x a[k][j]
```

3 Méthodologie

1. Réfléchir à la **distribution des données sur les processeurs**: quels sont les choix de répartition de la matrice les schémas et coûts de communications engendrés. On suppose pour simplifier que la **taille de la matrice est divisible par le nombre de processeurs**.
2. Copier le source tokenring-sibling.c en gaussp.c. A partir du programme d'élimination de gauss séquentiel gauss.c, fabriquer le squelette de votre programme parallèle : le main() de gauss.c devient dowork() de gaussp.c
3. Commencez par paralléliser le **chargement** (matrix_load()) et la **sauvegarde** (matrix_save()) du fichier de la matrice en incluant sa **diffusion** et sa **concentration** sur les processeurs.

Réfléchissez à la possibilité d'utiliser NFS en accès concurrents.

Valider les procédures développées et observer les communications avec xpvm avec les matrices d'exemples de différentes tailles /user/pvm/pvm3/m*.

4. Paralléliser le calcul de l'élimination de Gauss.
5. Prenez un nombre de tâches inférieur ou égal à n . Commencez par valider votre programme avec xpvm et peu de tâches (2 ou 4) sur des matrices de petite tailles (4, 8, 16).

Puis faites des mesures de Speedup et d'Efficacité en fonction du nombre de machines sur des matrices de taille plus importantes (500, 1000 et au dela...) et sur des réseaux différents (salle 500x, 510x, 5201...)

4 Remarques

1. Ne pas inclure les accès fichiers dans les mesures de temps; l'objectif du TP n'est pas un benchmark de NFS, mais de votre algorithme de calcul parallèle.
2. Inutile de sauver la matrice résultat lors des mesures de Speedup
3. Faut il utiliser pvm ou xpvm pour faire les mesures de speedup ? Faut il faire des printf() ?
4. Est il possible d'améliorer les performances en faisant du recouvrement calculs / communications ? Si oui, faites des mesures avant / après.
5. Que se passe-t-il lorsqu'un pivot est nul ?

5 Annexes : contenu de Makefile et .cshrc

Exemple de Makefile minimum (à copier depuis /user/pvm/pvm3/Makefile) pour pouvoir compiler un source qui s'appelle hello.c et qui possède un fichier d'en-tête hello.h :

```
# Makefile example for PVM at ESIEE

# include local arch. particularities
include $(PVM_ROOT)/conf/$(PVM_ARCH).def

CFLAGS =      -I$(PVM_ROOT)/include $(ARCHCFLAGS)
LIBS      =      -lpvm3 -lgpvm3 $(ARCHLIB) -lm
LFLAGS    =      -L$(PVM_ROOT)/lib/$(PVM_ARCH)
LDFLAGS    =      $(LFLAGS) $(LIBS)

# new default rule including mv to bin/$(PVM_ARCH)
.c
    $(CC) $(CFLAGS) -o $@ $< $(LDFLAGS)
    mv $@ $(HOME)/pvm3/bin/$(PVM_ARCH)
```

Fichier cshrc.pvm :

```
# .cshrc for pvm and xpvm at ESIEE

setenv PVM_ROOT /user/pvm/pvm/pvm
setenv PVM_RSH 'which ssh'
setenv XPVM_ROOT ${PVM_ROOT}/xpvm
setenv PVM_ARCH '$PVM_ROOT/lib/pvmgetarch'

if ( $?MANPATH ) then
    setenv MANPATH ${MANPATH}:${PVM_ROOT}/man
else
    setenv MANPATH /usr/share/man:${PVM_ROOT}/man
endif

# xpvm requires Tcl/Tk: use /user/info/ActiveTcl

source /user/info/IHM/tcltk.csh
set path = ( $PVM_ROOT/lib $PVM_ROOT $XPVM_ROOT/src/$PVM_ARCH $path )
```