

21 nov 07 15:44 **Info_TP** Page 1/3

```
=====
      COMPILATION (sur ferme)
Avoir dans son PATH :
      /usr/local/bin/
      /usr/java/j2sdk1.4.0_03/bin/

compilation :
      ompcc -o pi pi.c second.c
=====

      INSTALLATION LA PLUS RECENTE

      COMPILATION (sur un noeud)

      /opt/omni-1.6/bin/ompcc

      /opt/omni-1.6/bin/ompcc -help

node2:~$ /opt/omni-1.6/bin/ompcc -omniVersion
RWCP Omni OpenMP Compiler version 1.5
=====

      EXECUTION sur un NOEUD

node2:~/OpenMP/calcul_pi$ export OMP_NUM_THREADS=1
node2:~/OpenMP/calcul_pi$ ./pi
Donner le nombre d'intervalles
1000000
pi = 3.14159
time=0.0218282
node2:~/OpenMP/calcul_pi$ export OMP_NUM_THREADS=2
node2:~/OpenMP/calcul_pi$ ./pi
Donner le nombre d'intervalles
1000000
pi = 3.14159
time=0.0135992
=====

Exos :
--- calcul de pi, code sequentiel ~counilh/OpenMP/calcul_pi/pi-seq.c

ecrire code OpenMP (versions avec directive critical, atomic (ces
deux versions ne sont pas performantes); version avec clause reduction).

-- code OpenMP pour ~counilh/OpenMP/loops/loop1_seq.c

-- code OpenMP pour ~counilh/OpenMP/balance/balance_seq.c
tester differentes methodes de repartition des iterations entre
les threads.

-- tester code OpenMP ~counilh/OpenMP/maximum/maximum.c

-- code OpenMP pour ~counilh/OpenMP/laplace/lap_seq.c

-- code OpenMP produit matrice-vecteur
parallelisme imbrique
```

mercredi 21 novembre 2007

21 nov 07 15:44 **Info_TP** Page

```
-- code OpenMP pour mandelbrot

-- code OpenMP pour ~counilh/OpenMP/life/life_seq.c
code OpenMP SPMD
code MPI+OpenMP
=====
      MPI+OpenMP

>>> exemple de code :
~counilh/OpenMP/prod_scal_mpi_openmp/prod_scal_mpi_omp.c

>>> COMPILATION SUR UN NOEUD :
compilation avec new_omppicc : ~counilh/OpenMP/cmd/new_omppicc

>>> compilation sur un noeud :

node5:~/OpenMP/prod_scal_mpi_openmp$ ./new_omppicc prod_scal_mpi_omp

>>> EXECUTION SUR LES NOEUDS :
node5:~/OpenMP/prod_scal_mpi_openmp$ /opt/mpich2-1.0.3th/bin/mpirun -n 3 p
cal_mpi_omp 2 200000
=====
      ANCIENNE INSTALLATION :
>>> compilation avec ompmpicc :
voir ~counilh/OpenMP/cmd/ompmpicc

>>> compilation sur ferme :
ompmpicc prod_scal_mpi_omp

>>> execution sur un noeud :
avec mpirun -> /opt/mpip4/bin/mpirun

(ou /exec/COMMUN/mpich-p4/bin/mpirun)

./mpirun -machinefile machines -np <nb procs mpi> \
./prod_scal_mpi_omp <nb threads par proc> \
<taille du vecteur par tache OpenMP>

>>> resultats (avec /opt/mpip4/bin/mpirun)
pour une taille totale de vecteur de 1200 000
temps pour x processus mpi avec chacun y threads OpenMP

x/y      1      2      3      4
1      0.0169  0.0112
2      0.0091  0.0062
4      0.0065  0.0049      0.0051
8      0.0093

>>> resultats (avec /opt/mpich2-1.0.3th/bin/)
pour une taille totale de vecteur de 1200 000
temps pour x processus mpi avec chacun y threads OpenMP

x/y      1      2      3      4
1      0.0176  0.0131
2      0.0100  0.0068
```

Info_TP

4	0.0070	0.091
8		