Parallel Programming : TP

- Docker environment
  - Docker QuickSheet : Docker.md
  - Command to start development environment

    > docker pull gratienj/parallel-programming-tp:v2023.01

    > docker images

    REPOSITORY TAG IMAGE ID CREATED SIZE

    gratienj/parallel-programming-tp v2023.01 b14690d029fa 2 hours ago 1.81GB

    gratienj/pptp v2023 683c32fb1561 4 hours ago 1.81GB

    > docker run -it --rm -v $PWD:/workdir -w /workdir gratienj/parallel-programming-tp:v2023.01 bash

    # git clone https://github.com/jgratien/ParallelProgrammingCourse.git

Parallel Programming : TP

Structures des Tps :

- Compilations

> cd ParallelProgramingTP
> source cemef.env
> mkdir build ; cd build
> cmake ..
> make install

-Exécutions des TPS

Les executables sont dans :

ParallelProgrammingTP/bin

> bin/<exe-name>.exe –help

pour connaître les options des tests

Parallel Programming : TP

Listes des TPs:

- TP1 :

Objectifs : maîtrise basique des technos (threads,openmp,tbb,mpi)

- TP2 :

Objectifs : mise en œuvre dans le cadre des produits matrice-vecteurs dense ou creux

- TP3 :

Objectifs : niveau avancé, mise en œuvre des algorithmes de type wavefront pour paralléliser LU

- TP4 :

Objectifs : niveau avancé, mise en œuvre de openmp et mpi pour le traîtement d’images

Parallel Programming : TP1

- TP1 :
  - > helloword_stdthread.cpp
  - > helloword_openmp.cpp
  - > helloword_task.cpp
  - > helloword_tbb.cpp
  - > helloword_mpi.cpp

Complétez les sources avec les différentes tecnos

- > cd ParallelProgrammingTP/src/TP1
- > make -C ../../build install
- > ../../bin/helloword_<...>.exe –nb-threads 4

Parallel Programming : TP2

- TP2 :
  - > densemv.cpp (cas dense)
  - > spmv.cpp (cas creux)
  - > densemv_mpi.cpp (cas dense + mpi)
  - > spmv_mpi.cpp (cas creux + mpi)

Complétez les sources avec les différentes tecnos

> cd ParallelProgrammingTP/src/TP2
> make -C ../../build install
> ../../bin/spmv.exe –help
> ../../bin/spmv.exe –nb-threads 4 –nx 100

nx : taille du cube permettant de générer la matrice laplacienne

nb-threads : nombre de threads

Parallel Programming : TP3

- TP3 :
  - > lu.cpp
  - > wave.cpp

Complétez les sources avec les différentes tecnos

> cd ParallelProgrammingTP/src/TP3
> make -C ../../build install
> ../../bin/lu.exe –help
> ../../bin/lu.exe –nb-threads 4 –nx 100

nx : taille du cube permettant de générer la matrice laplacienne

nb-threads : nombre de threads

Parallel Programming : TP4

- TP4 :
  > img.cpp
  > img2.cpp

Paralléliser avec les technos openmp et mpi

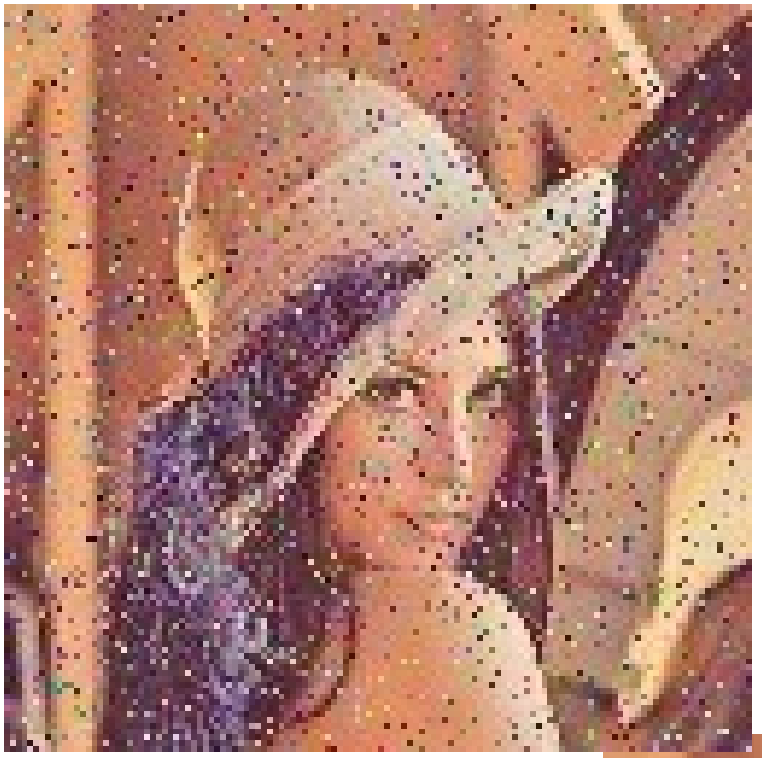
  > cd ParallelProgrammingTP/src/TP4
  > make -C ../../build install
  > ../../bin/img.exe –help
  > ../../bin/img.exe –file lena.jpg –show 1
  > ../../bin/img.exe –file lena.jpg –add-noise 1 –noise-density 20

Fitre médian

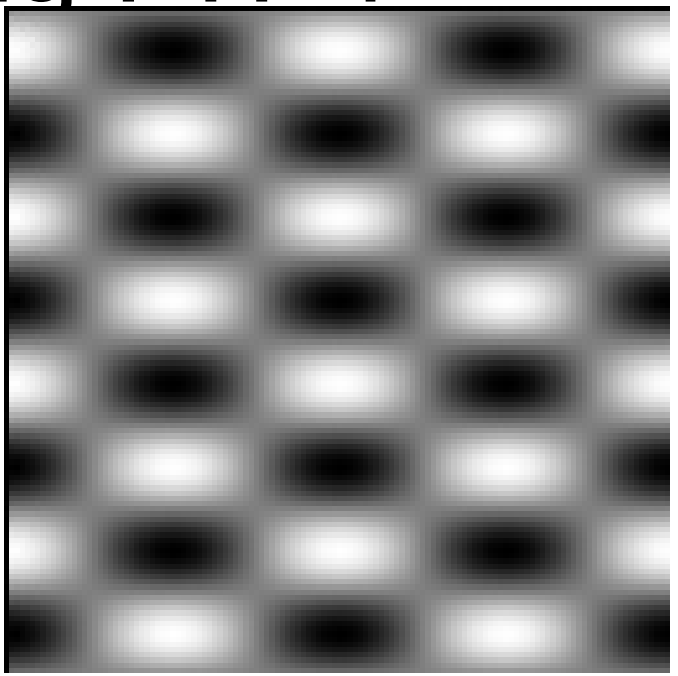
  > ../../bin/img.exe –file noisy-lena.jpg –filter 1

Labélisation

  > ../../bin/img.exe –nx 128 –ny 128 –LX 16 –LY 32 –filter 200
    pour générer une image avec des zones connexe
  > ../../bin/img.exe –nx 128 –ny 128 –LX 16 –LY 32 –filter 200 –compute-nb-cc 1
    pour compter le nombre de zone connexe



Parallel Programming : TP4



- TP4 :
  - > img.cpp
  - > img2.cpp

Paralléliser avec openmp et mpi

> cd ParallelProgrammingTP/src/TP4

> make -C ../../build install

> ../../bin/img2.exe –help

> ../../bin/img2.exe –nx 128 –ny 128 –LX 16 –LY 32 –filter 200

pour générer une image avec des zones connexe

> ../../bin/img2.exe –nx 128 –ny 128 –LX 16 –LY 32 –filter 200 –compute-nb-cc 1

pour compter le nombre de zones connexes