Problem set 7, Parallelization

TDT4200, Fall 2015

Deadline: 12.11.2015 at 23:59. Contact course staff if you cannot meet the deadline.

Evaluation:

- 5 points for a working MPI implementation
- 5 points for a working CUDA implementation
- 5 points for a working OpenMP implementation

Delivery: Use *itslearning*. Deliver one .zip file, with:

- *yourNTNUusername_code_ps7.zip* containing your modified versions of the files:
  - `newImageIdeaOMP.c`
  - `newImageIdeaGPU.c`
  - `newImageIdeaMPI.c`
- Do *not* include any image files.

The unmodified ppm.c and ppm.h files can be included.

General notes:

The MPI and GPU versions must compile and run on `its-015-XX.idi.ntnu.no` (XX being any of the lab machines in ITS015), while the OpenMP version must compile and run in `Problem_set_7` in the TDT4200_h2015 group on `climb.idi.ntnu.no`.

You should only make changes to the files indicated. Do not add additional files or third party code/libraries.

Code delivery

Problem set 7 starts over again with the same code as in problem sets three and six (PS3 and PS6). This time we focus on making parallel versions in different APIs, however.

Start with the three handout files newImageIdea<API>.c, and make a working MPI, OpenMP (OMP) and CUDA (GPU) version respectively. The supplied `Makefile` should be used as is, without any changes.

The `Makefile` creates an OpenMP executable named `newImageIdeaOMP` by using the makefile target. The program reads the `flower.ppm` image and creates three new images: `flower_tiny.ppm`, `flower_small.ppm` and `flower_medium.ppm`.

Use the rule *make run* to run `newImageIdeaOMP` and create the images. Then use the rule *make check* to create the correct images, and count the number of pixel errors your code produces.

The procedure is similar to run the MPI and CUDA versions – with make targets *newImageIdeaMPI*, *runmpi*, *newImageIdeaGPU* and *rungpu* to build and run MPI and GPU executables. Run *make check* for verification in all APIs.

In this problem set, only a very simple MPI implementation with **exactly four MPI ranks** is required.

As with PS3 and PS6, the code is allowed to have a few pixel errors in the final output in each image. This is tested with the *checker* program, as before. A few thousand pixels with ± 1 differences and a few hundred pixels with a larger difference is okay.

Report

Your report must contain:

- A very short description on how the code is parallelized with MPI
- A very short description on how the code is parallelized with OpenMP
- A very short description on how the code is parallelized with CUDA
- Timings of the MPI and CUDA versions with the *time* command
- The CMB numbers Time (s), Energy (j) and EDP (js)
- Provide your CMB user name if different from the NTNU user name.

Please make the code readable. Remove debug tests and unused code to make it shorter. Some comments can be good as well. If your best code is slow a better report is advised.

Additional details can be found in the recitation slides for this problem set.