Final details for Assignment 2

NOTE : deadline postponed to Jan, *20th*, 2021

How to upload your work & detailed instructions about naming conventions

Submission

To submit your job please send an email to luca.tornatore@inaf.it accordingly the following details. You are supposed o produce:

- 2 codes (1 MPI version, 1 OpenMP version; alternatively, or in addition, 1 MPI+OpenMP hybrid code).
- 4 blurred images from the image `test_picture.pgm`:
  - 1 blurred with an average kernel and 1 blurred with a weight kernel of central size 0.2; both kernel have size 101 (i.e. radius 50) for both the MPI and OpenMP versions. If you developed only the hybrid code, only 2 images are expected.
- 2 plots with the scalability study, one for the strong scalability and one for the weak scalability (see below about the weak scalability). The 2 plots should be normalized, i.e. the reference value for the serial run (1 process or thread) is 1.0.
  Quote the absolute reference timing in seconds in your report, as well as the configuration that you have used for the scalability (how many MPI tasks, how many threads and the affinity/pinning that you have used if any)
  As for the scalability, you are allowed to use 1 node only also for the MPI version.
- A brief (4 pages max) report about your code, with a description of your code and the implementation details.
- a script file to compile your code; it is your choice whether to use a Makefile or not, so that the script may be something like

  ```
  #!/bin/bash
  make
  ```

  or something like

  ```
  #!/bin/bash
  OPTIMIZATIONS=$what_you_want
  cc $OPTMIZATIONS -fopenmp -o blur.omp blur.omp.c -l$the_libraries_you_need
  mpicc $OPTMIZATIONS -o blur.mpi blur.mpi.c -l$the_libraries_you_need
  mpicc $OPTMIZATIONS -fopenmp -o blur.mpi_omp blur.mpi_omp.c -l$the_libraries_you_need
  ```

  The only important thing is that the compilation is standardized by calling the script (see here for how to name it).
- the scripts that you have used for the scalability study

Place everything in a folder named as `your_family_name.your_name`, and tar it so that the resulting tar will be `your_family_name.your_name.tar` (compress it with zip, gzip, bzip or whatever else).

Then, send me an email at the specified address with the tar in attachment and any relevant comment in the email body.

Naming conventions

The naming convention for the **code files** is:

- serial version, if present: blur.c
- MPI version: blur.mpi.c
- OpenMP version: blur.omp.c
- Hybrid MPI+OpenMP (which is optional): blur.mpi_omp.c

The naming convention for the **blurred image files** is:

`output file name = original_file_name.b_#TYPE_#XSIZEx#YSIZE<_#CENTRALVALUE>.#VERSION.pgm`

where:

- the `original_file_name` is the file name of the original file without the `.pgm` (i.e. “`test_picture`" for our example)
- `#TYPE` is an integer that corresponds to the kernel type: Average = 0, Weight = 1, Gaussian = 2.
- `#XSIZE` and `#YSIZE` are the *x*- and *y*- sizes of the kernel (in your case you are allowed to use square kernels, so `#XSIZE`=`#YSIZE`; however, you are free to generalize).
- `#CENTRALVALUE` is meaningful only for the Weight Kernel, and so it will be present in the name only for that case. In the test check that you are given the `_02` is the central value 0.2; since we are considering only values in the range `[0:1]` follow the same convention.
- `#VERSION` is either `mpi` or `omp`; in the case you developed only the hybrid code, `#VERSION` is `mpi_omp`.

..well ok, to be perfectly clear, we do expect the following 4 files:
`test_picture.b_0_101x101.mpi.pgm`, `test_picture.b_1_101x101_02.mpi.pgm`,
`test_picture.b_0_101x101.omp.pgm`, `test_picture.b_1_101x101_02.omp.pgm`.
If you developed nly the hybrid code, we do expect the following 2 files:
`test_picture.b_0_101x101.mpi_omp.pgm`, `test_picture.b_1_101x101_02.mpi_omp.pgm`.

In the case (not required) that you want to challenge yourself and avoid the loss of luminosity at the borders, you find the corresponding results in the sub-folder `border_effect_accounted`.

The naming convention for the **report** is: report.pdf

The naming convention for the **scalability plots** is:

- strong scalability: strongscalability.png
- weak scalability: weakscalability.png

The naming convention **building script** is: compile

There is no naming convention for the **scripts for the scalability study**; however, please name them appropriately so we can immediately spot them.

Modifications to the requirements for the scalability

Given the limited computational resources (..although we're a bit surprised, accounting the the estimates that we made..) the requirements for the scalability study are relaxed:

1. do not use the large `101x101` and `501x501` kernels; use `11x11` and `31x31` kernels insted.
2. as for the **strong scalability** you can use the smaller image `earth-notsolarge.pgm` (size: 10000x10000) instead of the originally proposed `earth-large.pgm` (size 21600x21600). You still find both under either `/scratch/dssc/tornatore/` or `/storage/dssc/tornatore` on the Orfeo cluster.
   Using the smaller image and the `31x31` kernel should make it possible for you to finalize the scalability study.
3. As for the weak scalability - that you can conduct just by generating you images with random noise or by using the example routine to generate a gradient - **that is no longer required for the people that can not obtain the cpu-time**.
4. since the run-times are expected to be large enough for the fluctuations in repeated runs to be "small" (i.e. $\lesssim 1\%$ or even $\ll 1\%$), you can perform **only 1 run per each case**. In other words, no repeated runs are required (in the case of wek scaling, pls choose a case whose exec time respects the assumption).

Note, however, that to perform a decent weak scalability it is not needed that the reference time lasted hours, isn't it ?
In any case, it will be appreciated that you motivate in the report why you could not accomplish the task and the estimated run-time that you would have needed.

We will take care of a fair treatment for those of you that will succeed in all the originally required tasks, included the scalability studies. Our penchant is more for rewarding those students than for penalising the others.

Test files

In the G. drive repository where the lectures' videos are stored you find an additional folder `Materials for Assignment2` (https://drive.google.com/drive/u/0/folders/1iDXlDuOTxzCcaJAPoHIJ0IYV4rMXt8fs) where there are the test files to be used for automatic validation of your work.

- `test_picture.pgm` is the original picture that you have to blur; it is a small one (~20MB) because not all of you may have a broadband connection during the vacations.
- `test_picture.b_0_101x101.pgm` is the blur with an Average Kernel of size 101 (i.e. radius 50; that is the first kernel mentioned in the Assignment presentation).
- `test_picture.b_1_101x101_02.pgm` is the blur with a Weight Kernel of size 101 and central value of 0.2 (that is the second kernel mentioned).

As noted by a student, there was an inconsistency in the uploaded test picture blurred with the Weight Kernel. The correct version has been uploaded

How the blur result will be automatically checked

For the purpose of acceptance of your results, your final blurred image will be contrasted with the uploaded reference R image.
In order to your image V be considered valid, the following must hold

$$|V_{i,j} - R_{i,j}| < 0.05\ \times \text{max\_value} \qquad (1)$$

for each pixel (i, j), where `max_value` is the maximum value possible for a pixel (i.e. 65535) (note that is a **very** generous approach).
In case of failure, a visual, or in any case more detailed, inspection will be conducted with the precise intent of admit you to the oral examination where any possible problem will be discussed and checked. Our intent is *not* to stop you, so please do not be too much anxious about the admission to the oral examination.

The details about your code will be discussed at the oral examination.

Please, report the absolute running time for your result (i.e. the configuration that you have used - how many MPI tasks, how many threads and the affinity).

File for scalability study

In order to have your scalability study, use the slightly more larger file (21600x21600 pixels) `earth-large.pgm` that you find already on the Orfeo cluster in my scratch area `/storage/dssc/tornatore/earth-large.pgm` which should be readable by all of you.
For those that do not have access to the Orfeo cluster, the same file is available in the G. drive under the subfolder `Materials for Assignment2/Image for scalability study` (https://drive.google.com/drive/u/0/folders/1Ea5N1CUbvVNl-8o4fu6tF2VkhF3XHJ_C).

Assess the scalability of your codes using this image and:

- Weight Kernels of size 11, 31

  [the previsously required 101x101 and 501x501 kernels are now dropped]
- from 1 to the maximum number of physical cores on a Orfeo node for the omp version.
- using 1 node that you are allowed to use on Orfeo [previously, it was requested to use more than 1 nodes for the MPI version; due to the limited computational resources, this requirement is dropped and you are entitled to use only 1 node].
- please report the absolute timings (of course, the run-time for 1 MPI task / 1 thread is sufficient).

Of course, **in addition** you may want to generate whatever extremely large image that you want (for instance with just white noise) to perform a better scalability study.

Bug in `read_write_pgm_image.c`

Note: this is an old warning, no new bugs where there
I realized only now that in the provided example file `read_write_pgm_image.c` there was a bug in the `swap_image()` routine:

```
for ( int i = 0; i < size; i+= 2 )
```

must be replaced by

```
for ( int i = 0; i < size; i++ )
```