High Performance Computing

Problem Name

A.Y. 2021/22

|  |  |  |
| --- | --- | --- |
| Alessio Pepe | 0622701463 | [a.pepe108@studenti.unisa.it](mailto:a.pepe108@studenti.unisa.it) |
| Teresa Tortorella | 0622701507 | [t.tortorella3@studenti.unisa.it](mailto:t.tortorella3@studenti.unisa.it) |
| Paolo Mansi | 0622701542 | [p.mansi5@studenti.unisa.it](mailto:p.mansi5@studenti.unisa.it) |



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

# Index

[1 Index 2](#_Toc85651543)

[2 Problem Description 3](#_Toc85651544)

[3 How to Run 4](#_Toc85651545)

[3.1 Organization of the repository 4](#_Toc85651546)

[3.2 Build phase 5](#_Toc85651547)

[3.3 Test phase 5](#_Toc85651548)

[3.4 Install Python3 requirements phase 5](#_Toc85651549)

[3.5 Measure phase 5](#_Toc85651550)

[3.5.1 Change measurement parameters 6](#_Toc85651551)

[4 Results 7](#_Toc85651552)

[4.1 Results organization 7](#_Toc85651553)

[4.2 OS setup 7](#_Toc85651554)

[4.3 Hardware setup 7](#_Toc85651555)

[4.4 Size 3000 8](#_Toc85651556)

[4.4.1 O0 8](#_Toc85651557)

[4.4.2 O1 8](#_Toc85651558)

[4.4.3 O2 8](#_Toc85651559)

[4.4.4 O3 9](#_Toc85651560)

[4.5 Size 5000 9](#_Toc85651561)

[4.5.1 O0 9](#_Toc85651562)

[4.5.2 O1 9](#_Toc85651563)

[4.5.3 O2 9](#_Toc85651564)

[4.5.4 O3 9](#_Toc85651565)

[4.6 Size 8000 9](#_Toc85651566)

[4.6.1 O0 9](#_Toc85651567)

[4.6.2 O1 9](#_Toc85651568)

[4.6.3 O2 9](#_Toc85651569)

[4.6.4 O3 9](#_Toc85651570)

[License 10](#_Toc85651571)

# Problem Description

Decription

# How to Run

## Organization of the repository

The files containing the source code, the scripts, the license, and everything associated with it can be viewed at the link ~~INSERTLINK.COM/AAA/GITHUB~~ and have been provided together with the following document.

The following directories are present:

* **/include**: it contains all the header files necessary to solve the requested problem.
* **/script:** contains the scripts needed to evaluate the performance of the provided code.
* **/source:** it contains all the source code and the main file needed to solve the problem.
* **/test:** contains the test files, usable with the ctest framework, which demonstrate the actual functionality of the provided application.

The following folder is provided for the purpose of demonstrating the authors' measurements, but it can be deleted, or the content will be overwritten at the time of application launch:

* **/measures:** It contains all the raw and processed data of the evaluations (if provided contains measure made on one of the authors' machines). There are the following subfolders:
  + **/measures/plots:** Contains the plots, for each optimization level and size, of the speedup and efficiency as the number of threads used varies, extrapolated from the processed data. Each image has a name structure like *size\_ZZZZ\_opt\_OY.png*, where *Y* indicates the compiler optimization level used for those measurements, and *ZZZZ* indicates the size of the problem used for those measurements.
  + **/measures/processed:** Contains data extracted from raw measurements, where speedup and efficiency are compared based on the use of a different number of threads. Each file has a name structure like *size\_ZZZZ\_opt\_OY.csv*, where *Y* indicates the compiler optimization level used for those measurements, and *ZZZZ* indicates the size of the problem used for those measurements.
  + **/measures/raw:** Contains all raw measurements performed by the script. Each file has a name structure like *nth\_X\_opt\_OY\_size\_ZZZZ.csv*, where *X* represents the number of threads used for those measurements (0 indicates a sequential measurement), *Y* indicates the compiler optimization level used for those measurements, and *ZZZZ* indicates the size of the problem used for those measurements.

The following folders are not present in the directory provided, but will be generated when launching the application:

* **/.venv:** (will be generated at the script time if the user decide to use a virtual environment as described in the section below). It’s just the virtual environment for python3.
* **/build:** Contains all built file with the make command.
* **/docs:** Contains the Doxygen documentation of the source code. The documentation was provided in 2 formats:
  + **/docs/html**
  + **/docs/latex**

There are also the following files:

* **.gitignore:** Specifies intentionally untracked files to ignore. We used that for share our work on GitHub.
* **CmakeLists.txt:** contains all the directives needed to compile and test all the necessary executables.
* **Doxyfile:** it contains all the directives to automatically generate the documentation associated with the code.
* **licence.txt:** contains a transcript of the license to which the code is subject.
* **ReadMe.md:** probably the file that brought you here.
* **report.pdf:** it is this same file and contains an account of what has been done.
* **requirements.txt:** contains all the packages required by python to run the measurement script.

## Build phase

First, you need to compile all the files. For this you need a GCC compiler, OpenMP, Doxygen and Graphviz installed on your machine. If not, the process may fail.

1. Head to the main directory.
2. Create a new build folder inside:

user@PC:~/dir$ mkdir build

1. Go to the build folder:

user@PC:~/dir$ cd build

1. Launch the build command with the parameter the root directory containing the CMakeLists.txt file:

user@PC:~/dir/build$ cmake ..

1. Then run the make command

user@PC:~/dir/build$ make

At this point, in the build folder there will be all the compiled executables and in the main directory there will be a new folder named docs containing the source code documentation in html and latex.

## Test phase

You can verify the correct functioning of the application provided by using the cmake package.

1. Go to the build folder (if you are following all the directions then you will already be in this one).
2. Launch the ctest command:

user@PC:~/dir/build$ ctest

The desired output is as follows

100% tests passed, 0 tests failed out of 1

## Install Python3 requirements phase

You need to check whether the python requirements described in the requirements.txt file are satisfied or not. If they are not, the authors recommend using a virtual environment to not occupy memory and override the python installation. This can be done as:

1. Head to the main directory.
2. Create the virtual environment from the current python3 installation (we used python3.8). If you don't have it, you need the package python3.8-venv (installable with the command sudo apt install python3.8-venv).

user@PC:~/dir$ python3 -m venv .venv

1. Now you need to install all the required packages:

user@PC:~/dir$ source ./.venv/bin/activate

(.venv) user@PC:~/dir$ pip3 install -r requirements.txt

(.venv) user@PC:~/dir$ deactivate

## Measure phase

At this point, everything is ready to run the measurement script. To run the script, you need to run the following commands

user@PC:~/dir$ source ./.venv/bin/activate

(.venv) user@PC:~/dir$ python3 ./script/measures.py

(.venv) user@PC:~/dir$ deactivate

or, if you have not used a virtual environment

user@PC:~/dir$ python3 ./script/measures.py

You can see the output in the measurements folder as indicated in 3.1.

**Attention**: the measurement phase takes a lot of time and overwrite all files in the measures folder. The script will ask you to confirm if you want to overwrite all measures if a measures folder is already present like

(.venv) user@PC:~/dir$ python3 ./script/measures.py

WARNING: The folder mesures already exist. If you continue the old measures will be permanently deleted (NOT REVERSIBLE). Would you continue? (Y/n)

If you want just extract measures from already present raw measures you can use

(.venv) user@PC:~/dir$ python3 ./script/measures.py --nogenerate

If you want compute just raw measures use

(.venv) user@PC:~/dir$ python3 ./script/measures.py --noextract

**Note**: Once you have performed steps 3.2 to 3.4, you no longer need to perform them. Step 3.5 can be performed directly to obtain new measurements. In case of changes to the source code it will be necessary to repeat steps 3.2 and 3.3.

### Change measurement parameters

it is likely that, depending on the machine on which you want to perform the measurement, you want to edit the measurement parameters such as the dimensions tested, the number of threads used, the optimizations used and the number of measurements for each configuration.

To do this, you need to go to the */script/measures.py* file. Immediately after the license and the fields intended for the file's metadata, some constants are defined as shown here:

# --------- You can edit this fields -----------

SIZE = (3000, 5000, 8000)

OPT\_LEVEL = ('O0', 'O1', 'O2', 'O3')

N\_THREAD = (0, 1, 2, 4, 8)

REPETITION = 100

# ----------------------------------------------

The configurations are given by all the possible triples *(SIZE, OPT\_LEVEL, N\_THREAD)*. Therefore, if these have lengths 2,3 and 4 respectively, configurations will be obtained.

* To change the configurations simply add or remove values from *SIZE*, *OPT\_LEVEL* and *NTHREAD* (in the latter 0 must be present to correctly evaluate the speedup).
* To change the number of repetitions for each configuration just change the *REPETITION* value

# Results

## Results organization

Measurements are made on different dimensions of the problem, using a sequential version of the algorithms and a parallelized version by trying with different numbers of active threads. All the possibilities analyzed are divided by size of the problem and the data and their plot are reported.

## OS setup

All measurements were made using the Windows 10 operating system. Ubuntu 20.04 LTS virtualized using WSL 2 was used.

## Hardware setup

The hardware configuration is reported at the time of measuring the performance of the solutions provided. The configuration is extracted from /proc/cpuinfo and /proc/meminfo.

user@PC:~/dir$ cat /proc/cpuinfo

(…omitted firt 7 threads…)

processor : 7

vendor\_id : AuthenticAMD

cpu family : 23

model : 17

model name : AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx

stepping : 0

microcode : 0xffffffff

cpu MHz : 1996.185

cache size : 512 KB

physical id : 0

siblings : 8

core id : 3

cpu cores : 4

apicid : 7

initial apicid : 7

fpu : yes

fpu\_exception : yes

cpuid level : 13

wp : yes

flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr\_opt pdpe1gb rdtscp lm constant\_tsc rep\_good nopl tsc\_reliable nonstop\_tsc cpuid extd\_apicid pni pclmulqdq ssse3 fma cx16 sse4\_1 sse4\_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf\_lm cmp\_legacy cr8\_legacy abm sse4a misalignsse 3dnowprefetch osvw topoext ssbd ibpb vmmcall fsgsbase bmi1 avx2 smep bmi2 rdseed adx smap clflushopt sha\_ni xsaveopt xsavec xgetbv1 xsaves clzero xsaveerptr virt\_ssbd arat

bugs : sysret\_ss\_attrs null\_seg spectre\_v1 spectre\_v2 spec\_store\_bypass

bogomips : 3992.37

TLB size : 2560 4K pages

clflush size : 64

cache\_alignment : 64

address sizes : 48 bits physical, 48 bits virtual

power management:

user@PC:~/dir$ cat /proc/meminfo

MemTotal: 5587448 kB

MemFree: 4452404 kB

MemAvailable: 4709292 kB

Buffers: 27012 kB

Cached: 417884 kB

SwapCached: 0 kB

Active: 207008 kB

Inactive: 783184 kB

Active(anon): 144 kB

Inactive(anon): 545212 kB

Active(file): 206864 kB

Inactive(file): 237972 kB

Unevictable: 0 kB

Mlocked: 0 kB

SwapTotal: 2097152 kB

SwapFree: 2097152 kB

Dirty: 12 kB

Writeback: 0 kB

AnonPages: 545324 kB

Mapped: 91368 kB

Shmem: 72 kB

KReclaimable: 37652 kB

Slab: 66744 kB

SReclaimable: 37652 kB

SUnreclaim: 29092 kB

KernelStack: 3472 kB

PageTables: 15132 kB

NFS\_Unstable: 0 kB

Bounce: 0 kB

WritebackTmp: 0 kB

CommitLimit: 4890876 kB

Committed\_AS: 756860 kB

VmallocTotal: 34359738367 kB

VmallocUsed: 23056 kB

VmallocChunk: 0 kB

Percpu: 2432 kB

AnonHugePages: 182272 kB

ShmemHugePages: 0 kB

ShmemPmdMapped: 0 kB

FileHugePages: 0 kB

FilePmdMapped: 0 kB

HugePages\_Total: 0

HugePages\_Free: 0

HugePages\_Rsvd: 0

HugePages\_Surp: 0

Hugepagesize: 2048 kB

Hugetlb: 0 kB

DirectMap4k: 101376 kB

DirectMap2M: 3590144 kB

DirectMap1G: 2097152 kB

Additionally, the available physical memory is reported:

user@PC:~/dir$ echo $(($(getconf \_PHYS\_PAGES) \* $(getconf PAGE\_SIZE) / (1024 \* 1024))) "MB"

5456 MB

Finally, the available virtual memory is reported:

user@PC:~/dir$ echo $(($(getconf \_AVPHYS\_PAGES) \* $(getconf PAGE\_SIZE) / (1024 \* 1024))) "MB"

4348 MB

## Size 3000

### O0

### O1

### O2

### O3

## Size 5000

### O0

### O1

### O2

### O3

## Size 8000

### O0

### O1

### O2

### O3

# License



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.