FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION
OF HIGHER EDUCATION
ITMO UNIVERSITY

Parallel algorithms for the analysis and synthesis of data on the assignments No.3, 4, 5

Performed by
Ivan Dubinin
J4132c

St. Petersburg
2021

Assignment 3

Compile and run Assignment3.c program.

Explain in detail how it works.

Listing of the program

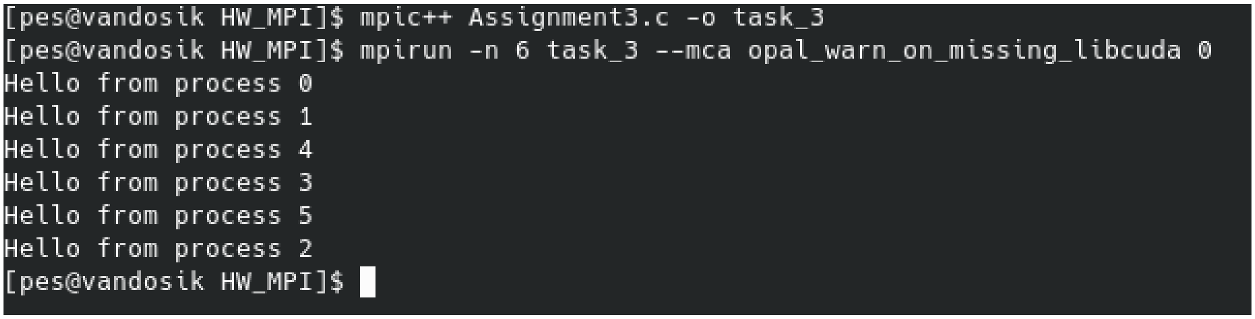
`See it in my github repo`

Description

To compile the program mpic++ compiler was used (installed from arch repoes). The program was executed by 6 processes (coinsiding the number of physical cores). The last parameter is turning off the usage of CUDA-aware, as it is included by default and I had troubles with it on my AMD Ryzen CPU and Radeon GPU config.

The zero proc puts to terminal “Hello from process 0” and when starts waiting for messages from the rest of procs. Other procs send their rank to zero proc by using MPI_Send call. The zero proc put to termonal the “Hello” message for every reseived message. So the number of messages equals to the number of procs.

Example of launch parameters and output



```
[pes@vandosik HW_MPI]$ mpic++ Assignment3.c -o task_3
[pes@vandosik HW_MPI]$ mpirun -n 6 task_3 --mca opal_warn_on_missing_libcuda 0
Hello from process 0
Hello from process 1
Hello from process 4
Hello from process 3
Hello from process 5
Hello from process 2
[pes@vandosik HW_MPI]$ 
```

Assignment 4

Convert the code Assignment4.c to match your individual version of the assignment.

Your task number is your number in the reporting table https://docs.google.com/spreadsheets/d/1m2KqmSc5pMMQ9JezZGybnuKG75FywU cs8gbPuMvVUxc. If your number is more than the options for the task, then calculate your number in modulus 28 (for example, 5 (mod 3) = 2).

Note. If the number of processes in the task is not defined, then we can assume that

this number does not exceed 8. The main process is understood as a process of rank 0 for the communicator MPl_COMM_WORLD. For all processes of nonzero rank in the tasks, the common name of the subordinate processes is used. If the task does not define the maximum size of a set of numbers, then it should be considered equal to 10.

Listing of the program

See it in my gothub repo

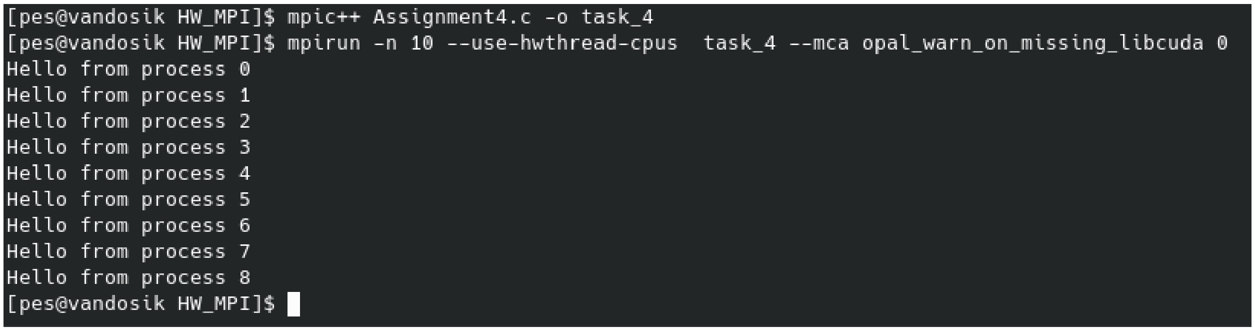
Description

To compile the program mpic++ compiler was used (installed from arch repoes). The program was executed by 10 processes (to increase the number of available slots option –use-hwthread-cpus was used). The last parameter is turning off the usage of CUDA-aware, as it is included by default and I had troubles with it on my AMD Ryzen CPU and Radeon GPU config.

The zero proc calls only MPI_Send finction, the last one calls only MPI_Recv function. Other procs make receive and after it call send, so the order of send and recv calls rotate over all the procs. As a result of it only 9 lines of output were put to the console (there are only 9 pairs of procs with consecutive ranks)

Results of the programm execution are shown on the picture below.

Example of launch parameters and output



```
[pes@vandosik HW_MPI]$ mpic++ Assignment4.c -o task_4
[pes@vandosik HW_MPI]$ mpirun -n 10 --use-hwthread-cpus  task_4 --mca opal_warn_on_missing_libcuda 0
Hello from process 0
Hello from process 1
Hello from process 2
Hello from process 3
Hello from process 4
Hello from process 5
Hello from process 6
Hello from process 7
Hello from process 8
[pes@vandosik HW_MPI]$ 
```

Assignment 5

Compile and run Assignment5.c program.

Explain in detail how it works.

Determine the execution time of the program from the previous task.

Listing of the program

Task_5 src

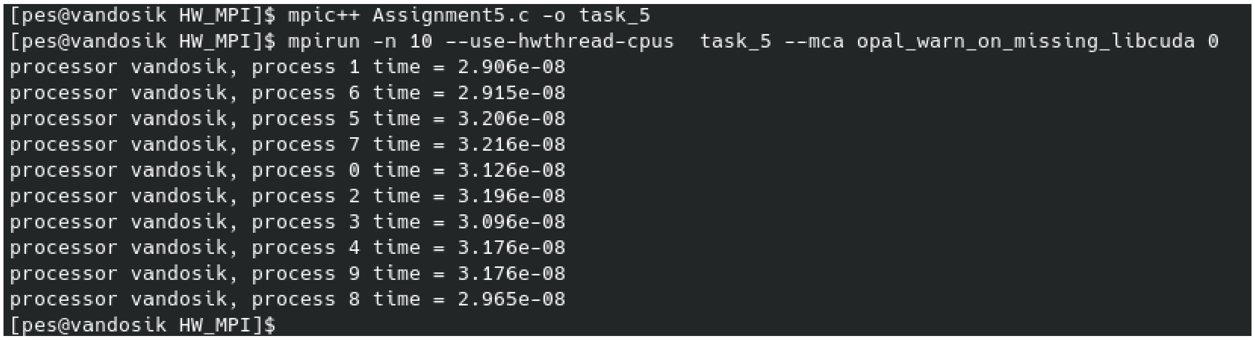
Modified Task_4 src

Description

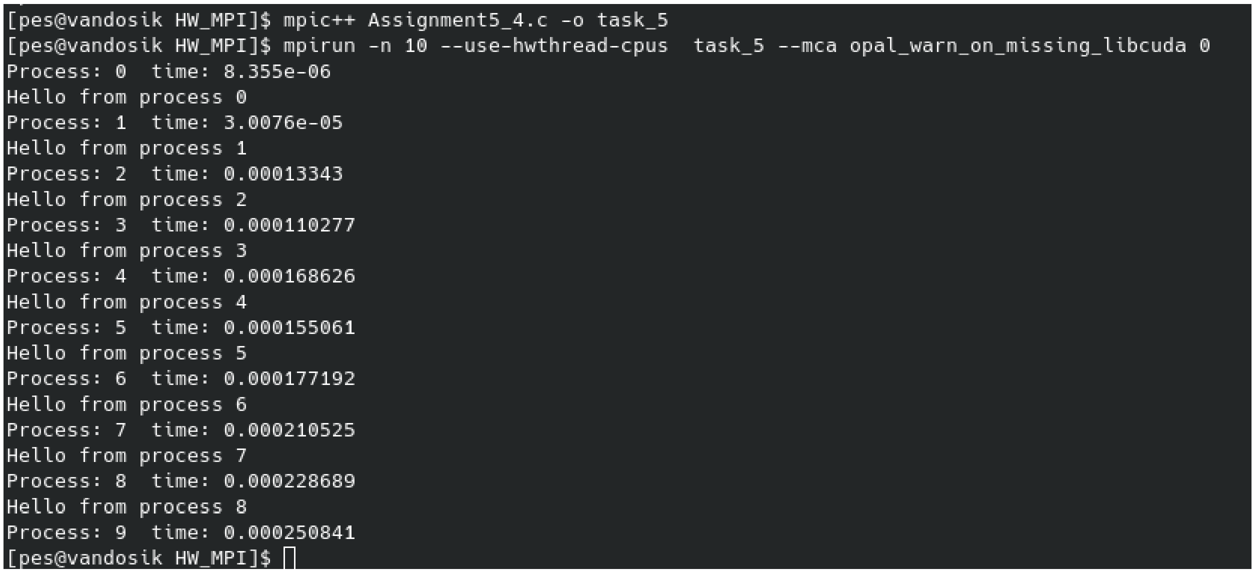
Assignment5 program initialize n processes. Every proc measures execution time of NTIMES loops (in each loop `time_finish` variable is updated with current time) and outputs processor name, process rank and average time of loop iteration execution.

The first screeshot demonstrates the execution of task_5 src file. The second one shows execution time of the modified program from the previous task

Example of launch parameters and output



```
[pes@vandosik HW_MPI]$ mpic++ Assignment5.c -o task_5
[pes@vandosik HW_MPI]$ mpirun -n 10 --use-hwthread-cpus  task_5 --mca opal_warn_on_missing_libcuda 0
processor vandosik, process 1 time = 2.906e-08
processor vandosik, process 6 time = 2.915e-08
processor vandosik, process 5 time = 3.206e-08
processor vandosik, process 7 time = 3.216e-08
processor vandosik, process 0 time = 3.126e-08
processor vandosik, process 2 time = 3.196e-08
processor vandosik, process 3 time = 3.096e-08
processor vandosik, process 4 time = 3.176e-08
processor vandosik, process 9 time = 3.176e-08
processor vandosik, process 8 time = 2.965e-08
[pes@vandosik HW_MPI]$
```



```
[pes@vandosik HW_MPI]$ mpic++ Assignment5_4.c -o task_5
[pes@vandosik HW_MPI]$ mpirun -n 10 --use-hwthread-cpus  task_5 --mca opal_warn_on_missing_libcuda 0
Process: 0  time: 8.355e-06
Hello from process 0
Process: 1  time: 3.0076e-05
Hello from process 1
Process: 2  time: 0.00013343
Hello from process 2
Process: 3  time: 0.000110277
Hello from process 3
Process: 4  time: 0.000168626
Hello from process 4
Process: 5  time: 0.000155061
Hello from process 5
Process: 6  time: 0.000177192
Hello from process 6
Process: 7  time: 0.000210525
Hello from process 7
Process: 8  time: 0.000228689
Hello from process 8
Process: 9  time: 0.000250841
[pes@vandosik HW_MPI]$
```