**Домашнее Задание №3**

**Липатов Данила Вячеславович**

**МСМТ 243**

**Ссылка на** [**GIT**](https://github.com/DanLip02/HPC_HSE)

**I пункт.**

Для запуска на кластере был скомпилирован код следующей командой

mpicc hw\_1\_mpi.c -o hw\_1\_mpi.out -lm

Предварительно подгрузив все необходимые модули:

module load INTEL/oneAPI\_2022

module load nvidia\_sdk/nvhpc/23.5

В целом, скрипт выполнился хорошо с маленькой погрешность. На рис. 1 предоставлено изображение для 11 точек:

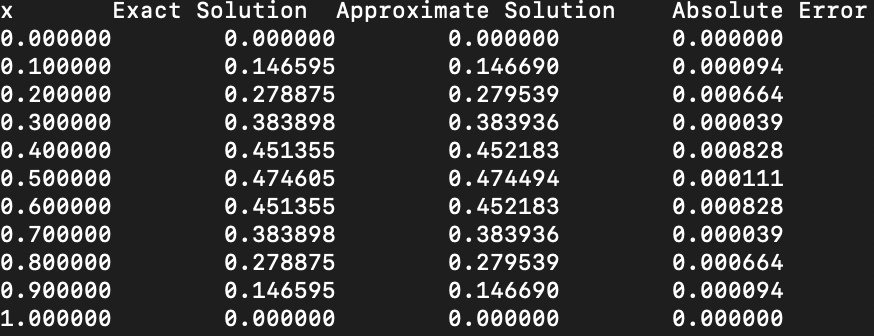


рис. 1

**II пункт.**

Аналогично предыдущему пункту через команду

srun -n 1 --constrain="type\_a" hw\_1\_mpi.out

Запускались для разных N и разных n от 1 до 24 выполнение скриптов

Графики зависимости n от времени предоставлен на рис. 2 и на рис.3 gflops от n.

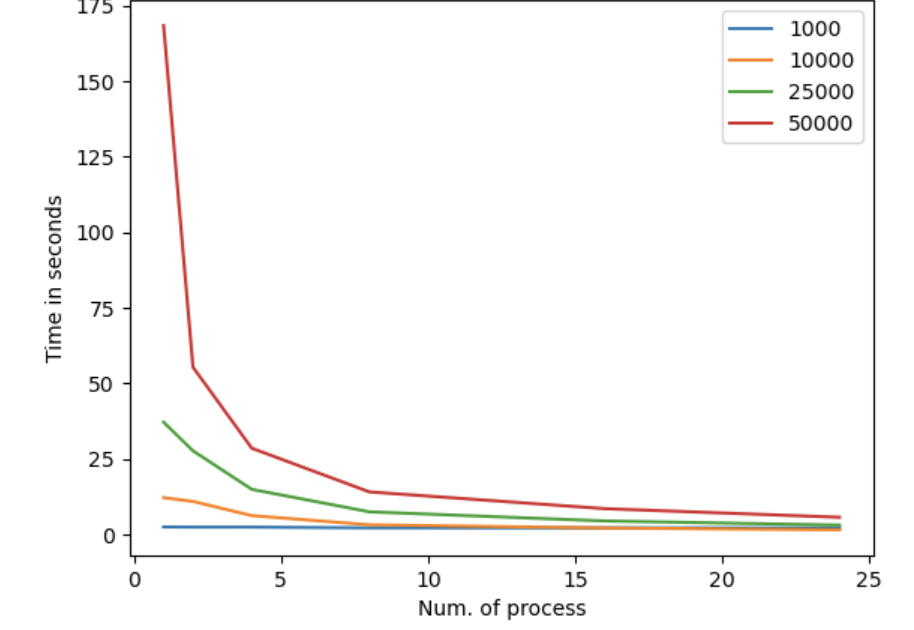


рис. 2

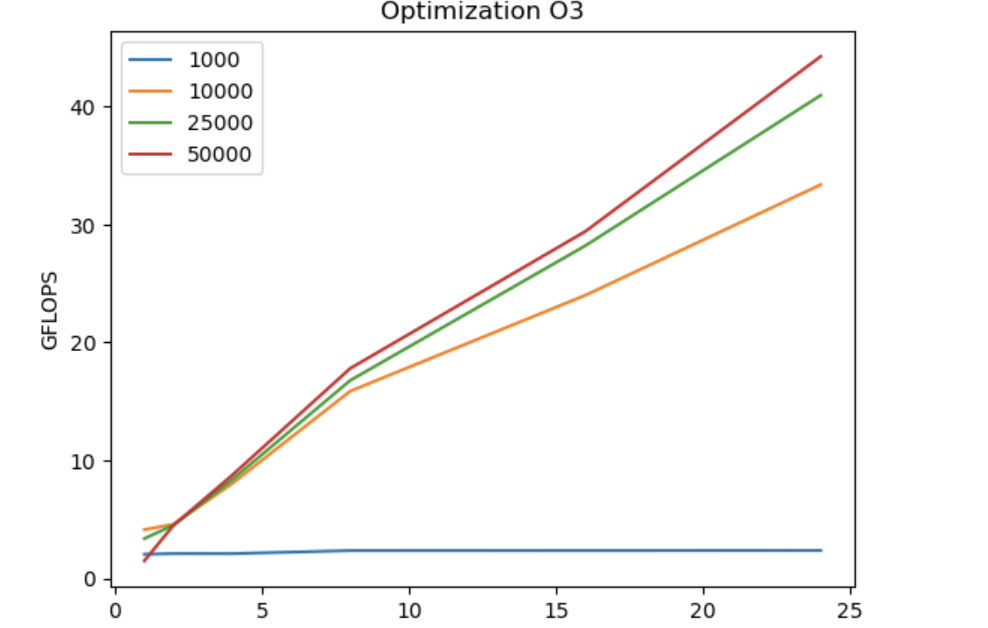


рис.3

**III пункт.**

Для асинхронного выполнения команды MPI\_Sendrecv были заменены на MPI\_Isend, MPI\_Irecv и так же добавлен барьер

MPI\_Waitall(request\_count, requests, MPI\_STATUSES\_IGNORE);

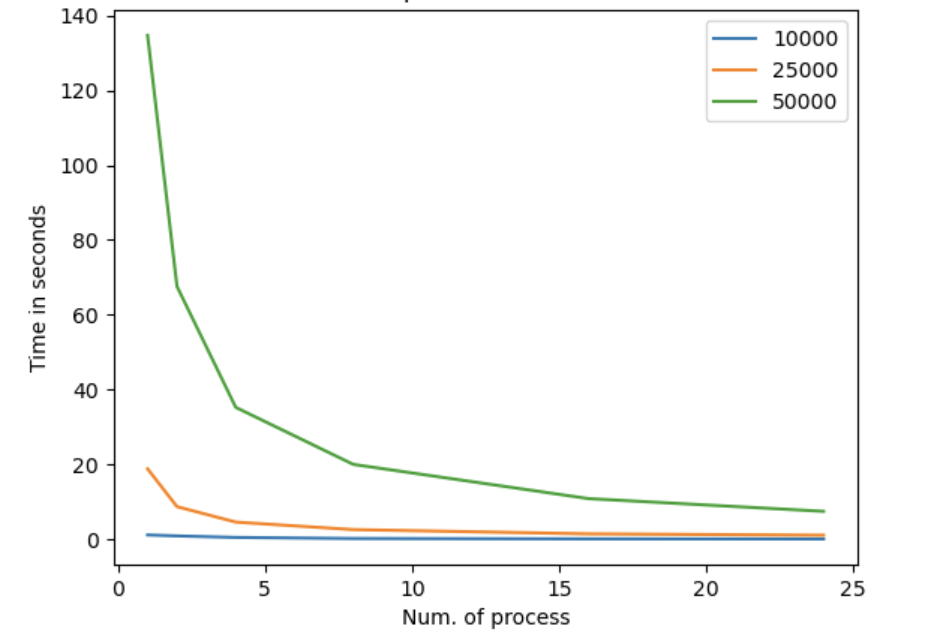


рис. 4

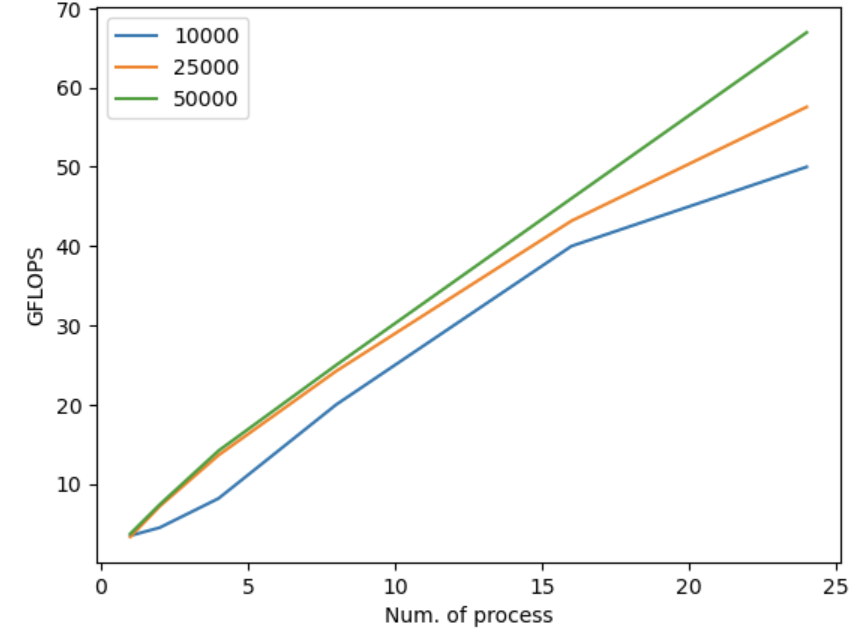


рис. 5

**IV Пункт.**

Помимо MPI\_Gatherv() был так же добавлен MPI\_Scatter()

Создаёт массив начальных температур. Этот массив рассылается другим процессам с помощью MPI\_Scatter.

В целом, сходимость с предыдущими реализациями такая же



Пример для 25000 точек.

**V пункт**

Наиболее трудным моментом является компиляция с CUDA:

nvcc -o hw\_1\_mpi\_cuda hw\_1\_mpi\_cuda.cu -ccbin mpicc -lcudart -lm

Где **-ccbin mpicc**

Указывает, какой компилятор должен использоваться для обработки CPU-кода.

**-lcudart**

Опция для линковки с библиотекой CUDA. Эта библиотека предоставляет функции, необходимые для работы программы с GPU.

srun -n 1 --gres=gpu:1 --constraint="type\_a" hw\_1\_mpi\_cuda.out

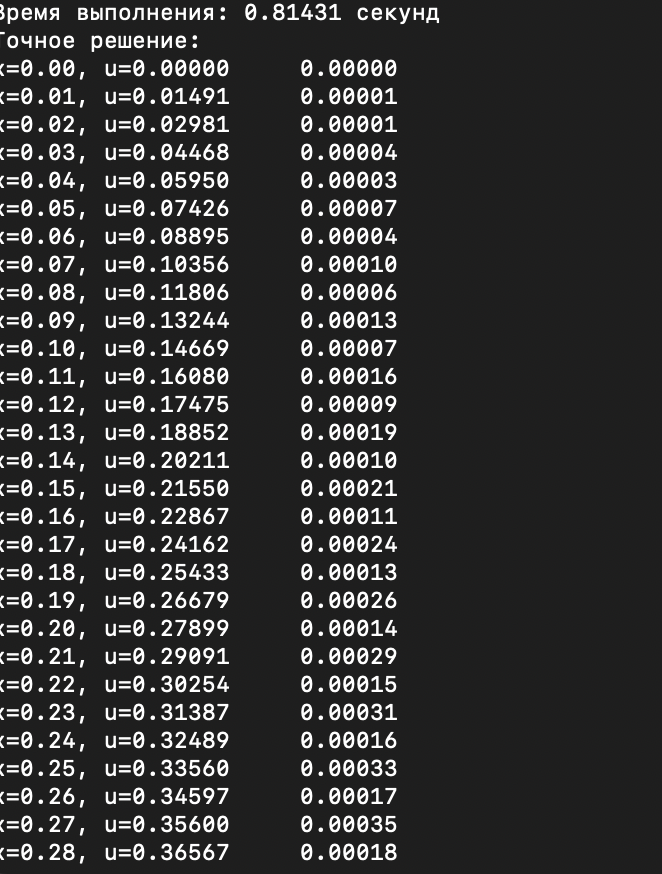


Рис. 7 (пример)