Отчет по лабораторной работе № 3

Выполнил: студент группы 6313-100503D Шаменков Максим Александрович

Эксперименты проводились на своём ноутбуке и суперкомпьютере.

Цель работы

Исследование эффективности параллельного умножения квадратных матриц с помощью библиотеки MPI.

Описание программы

Разработана программа на C++ (main.cpp), выполняющая перемножение двух квадратных матриц. Для распараллеливания вычислений использована библиотека MPI.

Программа включает:

- Генерацию случайных матриц (элементы в диапазоне [0, 99]).
- Параллельное умножение матриц.
- Замер времени выполнения.
- Сохранение исходных данных, результатов умножения и времени работы в файлы.

Для автоматизации экспериментов использованы вспомогательные скрипты:

- run_mpi_matrix.py — автоматический запуск MPI-программы с разным числом процессов. Замеры для каждого процесса лежат в папке results.
- statistic.py — проверка корректности вычислений (сравнение с результатами numpy) и построение графиков производительности.

Экспериментальные данные

Исследование проводилось на матрицах размеров: 100, 200, 300, 400, 500, 1000, 1500, 2000. Были произведены замеры с 1, 2, 4, 6, 8, 10 процессами.

Результаты и анализ

1. Корректность вычислений
   Результаты параллельного умножения совпадают с вычислениями через numpy, что подтверждает правильность реализации в файле comparison_results.txt.
2. Производительность
   - Наибольшее ускорение достигается при использовании 5–8 процессов.
   - Дальнейшее увеличение числа процессов не приводит к значительному росту производительности.
   - В оптимальном случае параллельная версия работает в 4–10 раз быстрее последовательной.

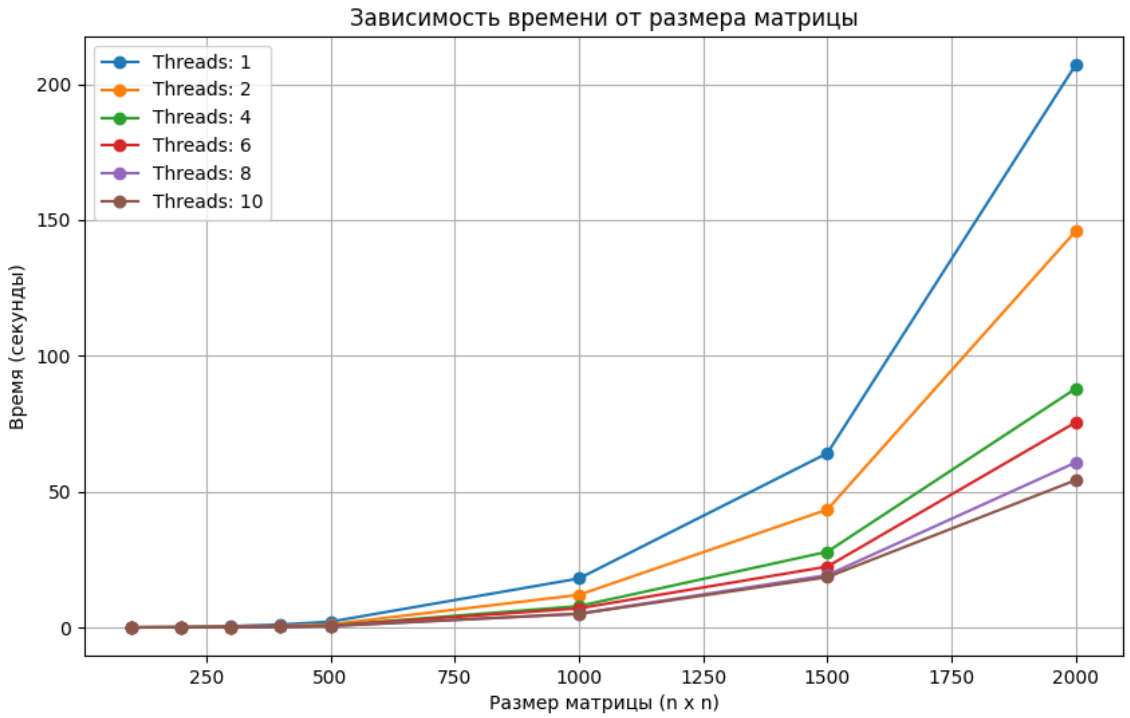


График 1. Результаты замеров на ноутбуке

Запуск MPI программы на суперкомпьютере «Сергей Королев»

Экспериментальные данные

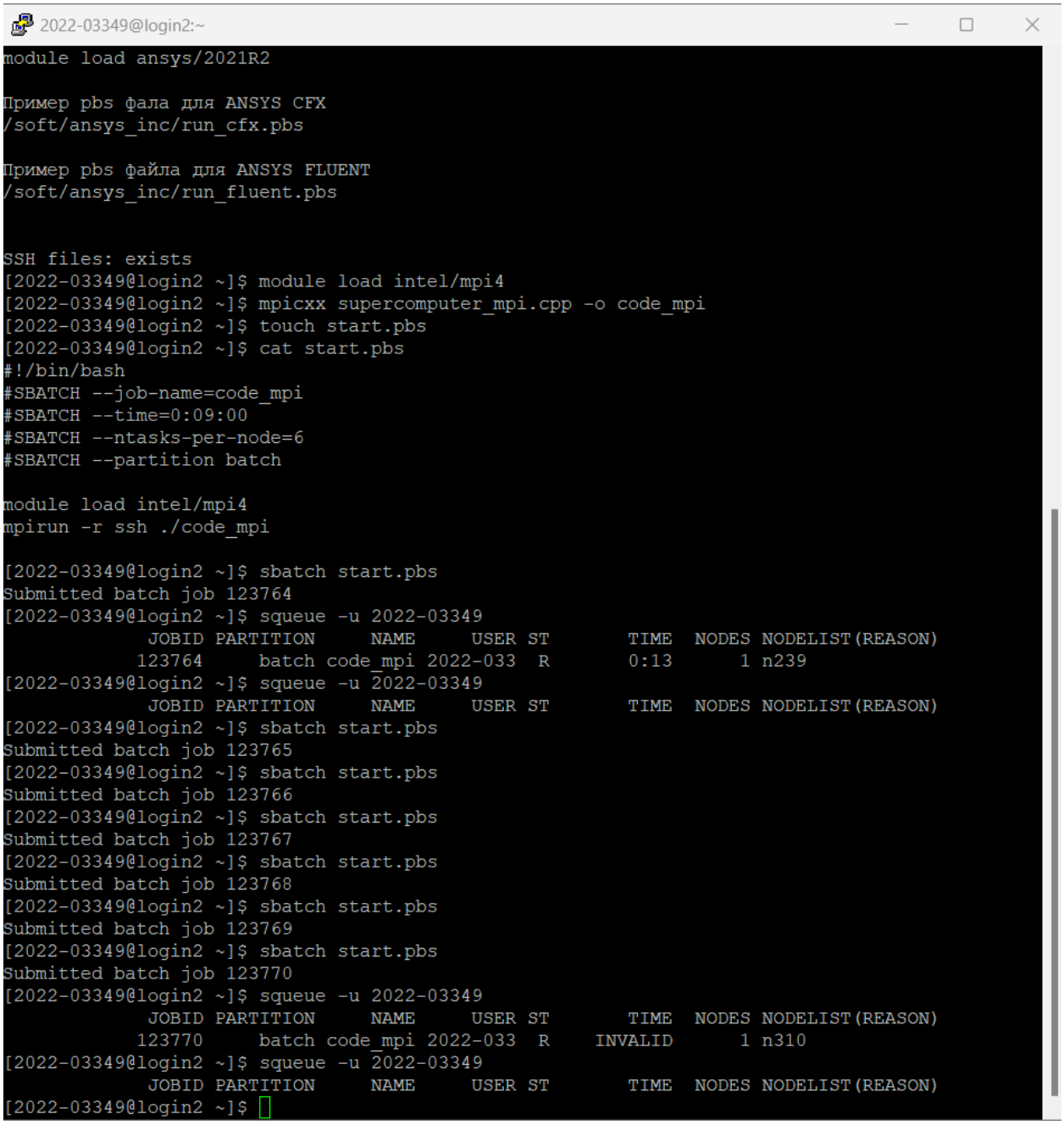
Исследование проводилось на матрицах размеров: 100, 200, 300, 400, 500, 1000, 1500, 2000.

Были произведены замеры с 1, 2, 4, 8, 12, 16, 20 процессами.

Подключение к суперкомпьютеру осуществлялось через программу эмулятор терминала с поддержкой протокола SSH PuTTY, обмен данными через WinSCP.

Программа main.cpp была немного видоизменена для запуска на суперкомпьютере. Исходный код программы находится в файле supercomputer_mpi.cpp.

Результаты замеров были перенесены для анализа в папку results_on_supercomputer. График также был построен с помощью statistic.py средствами matplotlib и находится в файле matrix_multiplication_time_on_supercomputer.png.


2022-03349@login2:~
module load ansys/2021R2
Пример pbs фала для ANSYS CFX
/soft/ansys_inc/run_cfx.pbs
Пример pbs файла для ANSYS FLUENT
/soft/ansys_inc/run_fluent.pbs
SSH files: exists
[2022-03349@login2 ~]$ module load intel/mpi4
[2022-03349@login2 ~]$ mpicxx supercomputer_mpi.cpp -o code_mpi
[2022-03349@login2 ~]$ touch start.pbs
[2022-03349@login2 ~]$ cat start.pbs
#!/bin/bash
#SBATCH --job-name=code_mpi
#SBATCH --time=0:09:00
#SBATCH --ntasks-per-node=6
#SBATCH --partition batch
module load intel/mpi4
mpirun -r ssh ./code_mpi
[2022-03349@login2 ~]$ sbatch start.pbs
Submitted batch job 123764
[2022-03349@login2 ~]$ squeue -u 2022-03349
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
123764 batch code_mpi 2022-033 R 0:13 1 n239
[2022-03349@login2 ~]$ squeue -u 2022-03349
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
[2022-03349@login2 ~]$ sbatch start.pbs
Submitted batch job 123765
[2022-03349@login2 ~]$ sbatch start.pbs
Submitted batch job 123766
[2022-03349@login2 ~]$ sbatch start.pbs
Submitted batch job 123767
[2022-03349@login2 ~]$ sbatch start.pbs
Submitted batch job 123768
[2022-03349@login2 ~]$ sbatch start.pbs
Submitted batch job 123769
[2022-03349@login2 ~]$ sbatch start.pbs
Submitted batch job 123770
[2022-03349@login2 ~]$ squeue -u 2022-03349
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
123770 batch code_mpi 2022-033 R INVALID 1 n310
[2022-03349@login2 ~]$ squeue -u 2022-03349
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
[2022-03349@login2 ~]$

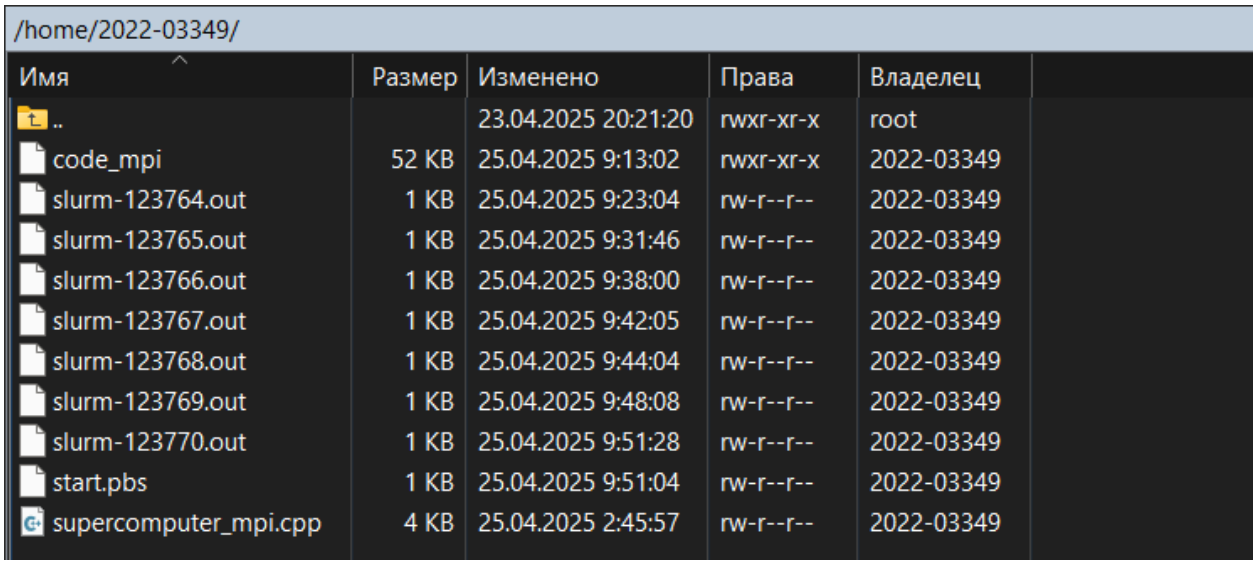

/home/2022-03349/
Имя
Размер
Изменено
Права
Владелец
..
23.04.2025 20:21:20
rwxr-xr-x
root
code_mpi
52 KB
25.04.2025 9:13:02
rwxr-xr-x
2022-03349
slurm-123764.out
1 KB
25.04.2025 9:23:04
rw-r--r--
2022-03349
slurm-123765.out
1 KB
25.04.2025 9:31:46
rw-r--r--
2022-03349
slurm-123766.out
1 KB
25.04.2025 9:38:00
rw-r--r--
2022-03349
slurm-123767.out
1 KB
25.04.2025 9:42:05
rw-r--r--
2022-03349
slurm-123768.out
1 KB
25.04.2025 9:44:04
rw-r--r--
2022-03349
slurm-123769.out
1 KB
25.04.2025 9:48:08
rw-r--r--
2022-03349
slurm-123770.out
1 KB
25.04.2025 9:51:28
rw-r--r--
2022-03349
start.pbs
1 KB
25.04.2025 9:51:04
rw-r--r--
2022-03349
supercomputer_mpi.cpp
4 KB
25.04.2025 2:45:57
rw-r--r--
2022-03349

График 2. Результаты замеров на суперкомпьютере

Вывод по лабораторной работе

В ходе выполнения лабораторной работы было исследовано параллельное программирование с использованием MPI. Программа была успешно запущена как на локальном ноутбуке, так и на суперкомпьютере, что позволило сравнить производительность в разных условиях.

Основные выводы:

1. Ускорение вычислений: Параллельная реализация с использованием MPI позволила значительно сократить время выполнения задачи по сравнению с последовательным алгоритмом, особенно на большом числе процессов.
2. Масштабируемость: На суперкомпьютере удалось достичь лучшей масштабируемости благодаря большему количеству вычислительных узлов и оптимизированной инфраструктуре.