Отчетная работа по курсу

“Суперкомпьютеры и параллельная обработка данных”

Сравнение работы параллельных версий алгоритма

с использованием технологий OpenMP и MPI.

Подготовил:

студент 327 группы ВМК МГУ,

Матевосов Димитрий

Москва, 2021

В предложенный алгоритм были внесены изменения для того, чтобы он выполнялся параллельно. В случае использования технологии **OpenMP** модификация кода сводится к добавлению прагмы **#pragma omp parallel for.** В версии программы с использованием **MPI** использовалась широковещательная рассылка (**MPI\_Bcast**) массивов, последующая операция редукции (**MPI\_Reduce**) для объединения результатов параллельных вычислений, операция **MPI\_Barrier** для синхронизации исполнения, а также **MPI\_Wtime** для замеров времени работы алгоритма.

Измененные файлы (код самого параллельного алгаритма, скрипт на python для создания графиков с данными о среденем времени выполнения каждой конфигурации программы) можно посмотреть на **GitHub**: <https://github.com/MediaX/skipods.git>

Оба параллельных алгоритма были проверены на совпадение выходных данных с оригинальным алгоритмом.

Для проверки алгоритма с разным количеством входных данных использовались макроконстанты (**MINI\_DATASET, SMALL\_DATASET, MEDIUM\_DATASET, LARGE\_DATASET, EXTRALARGE\_DATASET**) объявленные в оригинальном файле алгоритма.

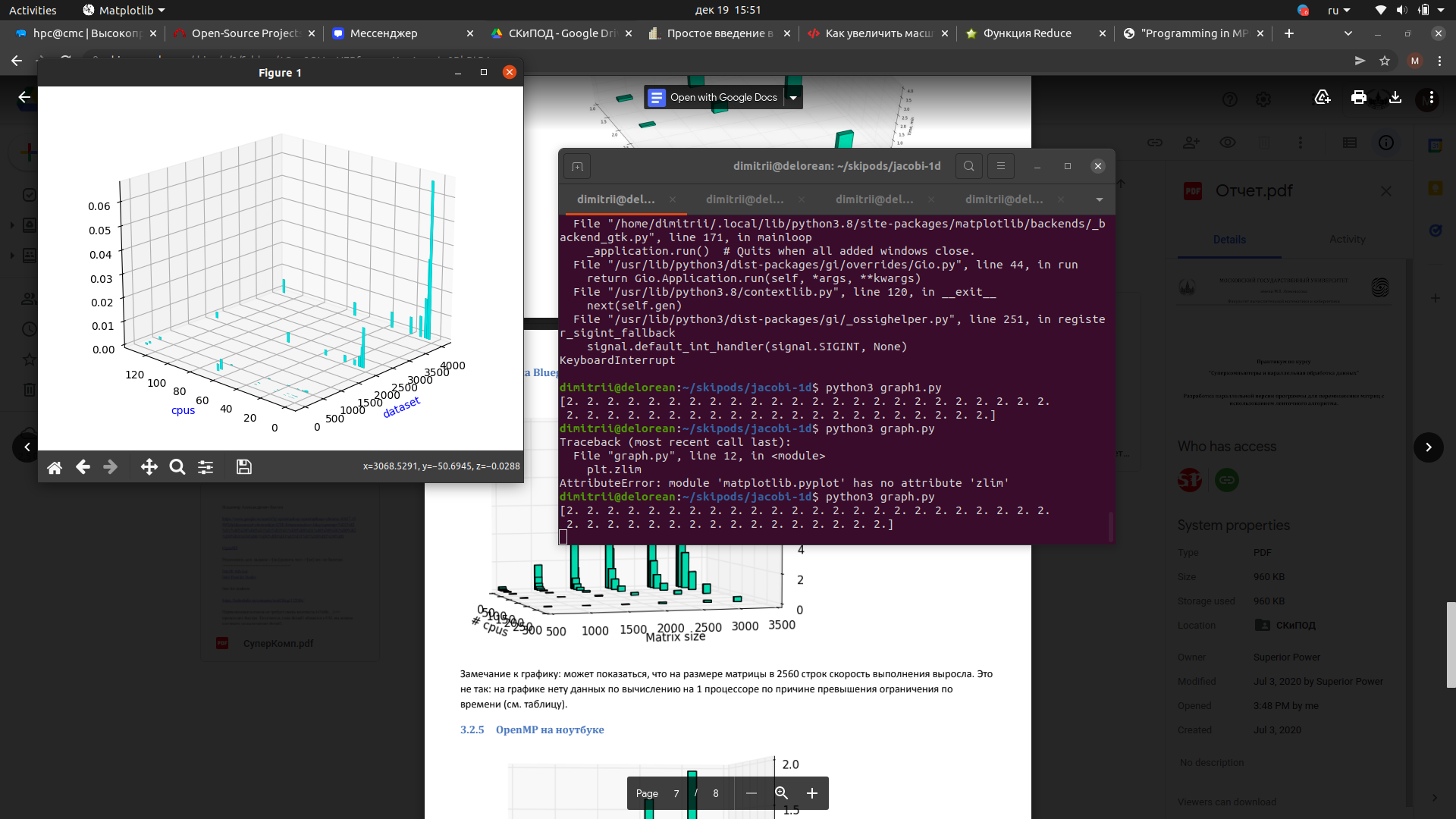
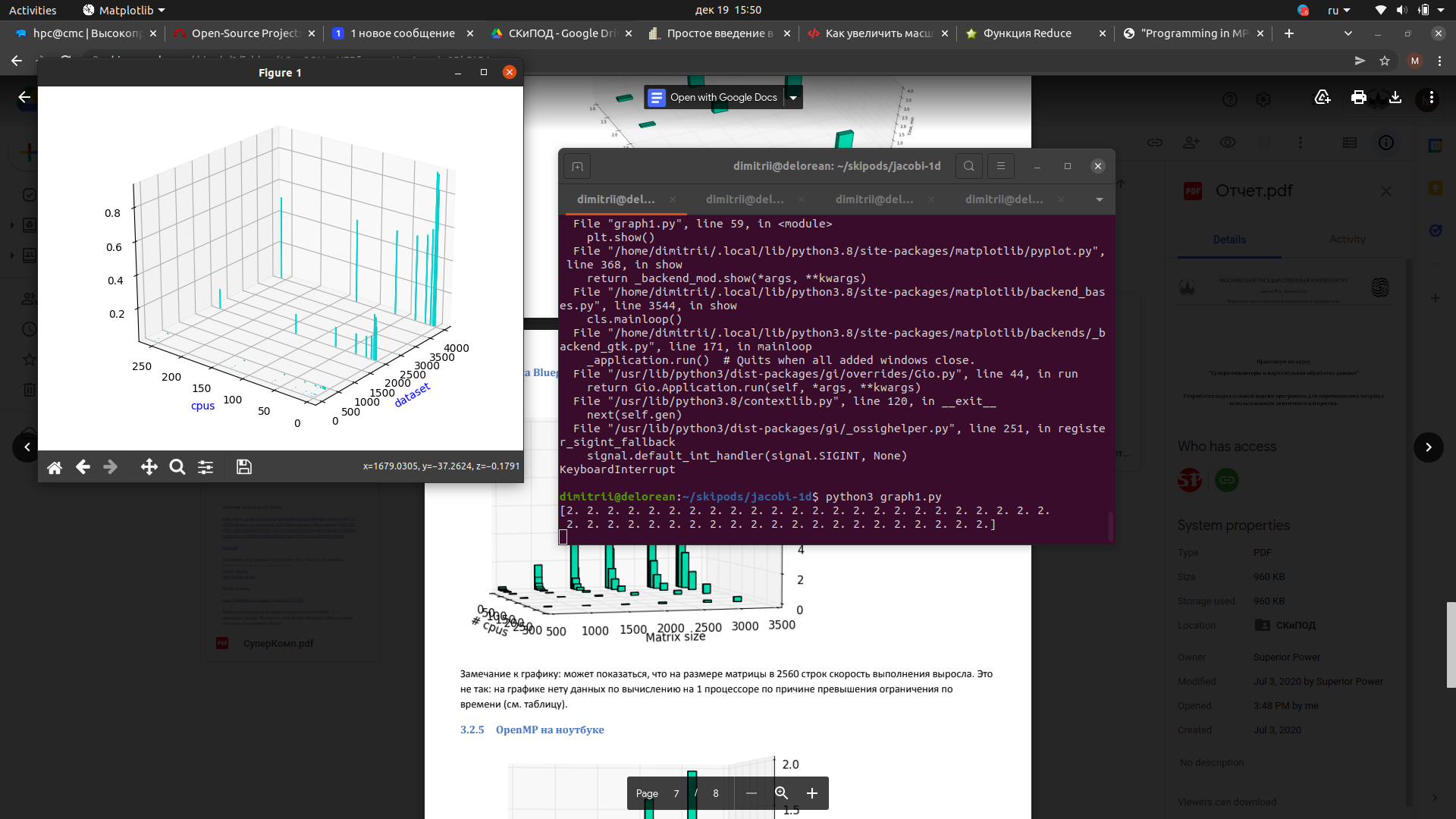
Кроме того, была проведена проверка алгоритмов с использованием разного количества ядер:

– **1, 2, 4, 6, 8, 16, 32, 64, 128** на машине **Polus** для алгоритма **OpenMP**

– **1, 2, 4, 6, 8, 16, 32, 64, 128, 256** на машине **Bluegene** для алгоритма **MPI**

[Import code](https://github.com/MediaX/skipods/import) В результате выполнения задания выяснилось, что делать алогритм параллельным с использованием технологии **MPI**, когда используется малое количество ядер, не имеет смысла, так как инициализация **MPI**, широковещательная рассылка и подобные операции занимают больше времени, нежели получаемый выйгрыш от параллельных вычислений. В то же время, технология **OpenMP** даже на малом количестве ядер показывает хороший результат.

По результатам всех проверок (для усреднения результатов каждый тест проводился по 3 раза), получилось построить 3-х мерные графики зависимости от времени, количества ядер и размера входных данных. Графики можно увидеть на следующей странице данного отчета.

**MPI на Bluegene**

**OpenMP на Polus**