# 1 Отчет 1. Чаплыгин Андрей Викторович, ВМК МГУ, группа 603.

**Первое задание курса 'Суперкомпьютерное моделирование и технологии программирования'**

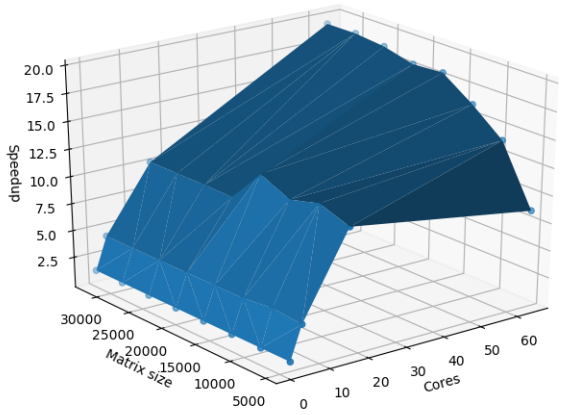
**Вариант 10.** Решение СЛАУ: Функция **PDGESV (double precision) из ScaLAPACK.**

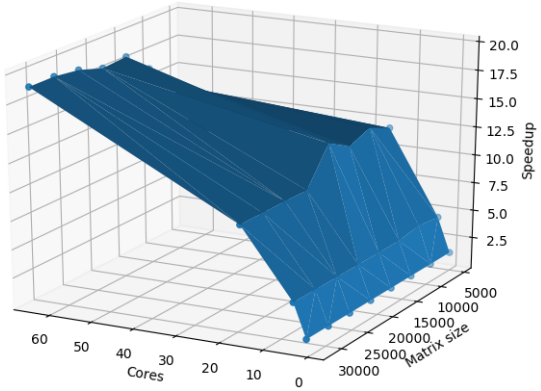
Платформа тестирования: **Polus** (`http://hpc.cs.msu.ru/polus`).

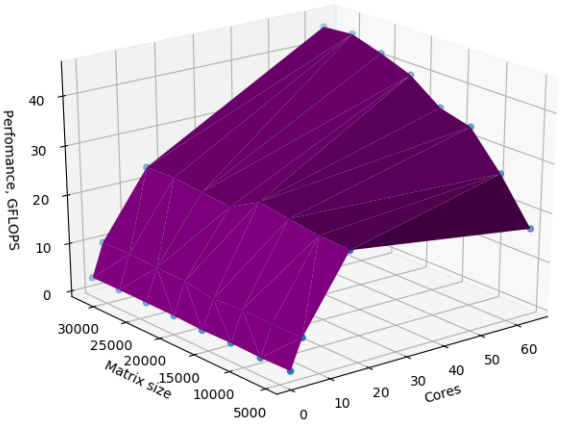
Программа состоит из вызова функции генерации случайной невырожденной матрицы (PDMATGEN), вызова решателя линейных систем (PDGESV) и расчет нормы невязки полученного решения.

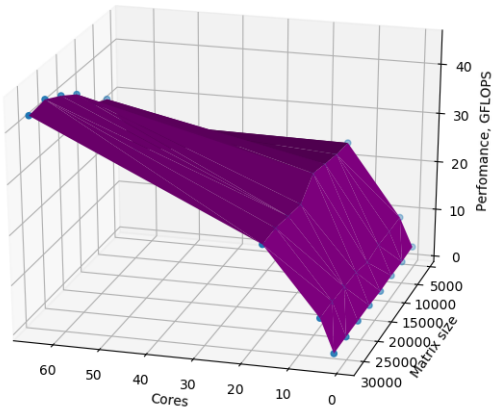
Рабочая программа лежит в git репозитории (`https://github.com/Andrcraft9/linear-solve_scalapack/tree/polus`). Ветка master - для компиляции на локальной машине, Ветка polus - для компиляции на Polus. Тестирование программы на кластере Polus проводилось из ветки polus! Компилирование на Polus проводилось с помощью компилятора Fortran OpenMPI и с линковкой библиотеки netlib ScaLAPACK. Тестирование проводилось для 1 - 64 ядер, размер матриц от 4096 на 4096 до 32768 на 32768, размер блока для всех экспериментов был выставлен 32 на 32.

На первом рисунке продемонстрировано ускорение программы в зависимости от количества ядер и размера матрицы.



На втором рисунке продемонстрирована производительность программы, которая считалась как:

$$\text{Perfomance} = \frac{\frac{2}{3}N^3}{\text{time}}$$

где $N$ - это размер матрицы.

В таблице показаны значения общего времени решения, ускорения и производительности:

| Matrix | Cores | Time (sec) | SpeedUp | Perfomance (GFLOPS) |
|---|---|---|---|---|
| 4096 | 1 | 23.338857 | 1.000000 | 1.962949 |
| 4096 | 4 | 5.714939 | 4.083833 | 8.016356 |
| 4096 | 16 | 2.013850 | 11.589176 | 22.748959 |
| 4096 | 64 | 2.613160 | 8.931278 | 17.531641 |
| 8192 | 1 | 198.442571 | 1.000000 | 1.846901 |
| 8192 | 4 | 45.775619 | 4.335115 | 8.006530 |
| 8192 | 16 | 15.830993 | 12.535068 | 23.151035 |
| 8192 | 64 | 13.962872 | 14.212160 | 26.248459 |
| 12288 | 1 | 608.582600 | 1.000000 | 2.032511 |
| 12288 | 4 | 151.488635 | 4.017348 | 8.165303 |
| 12288 | 16 | 51.622919 | 11.789000 | 23.961268 |
| 12288 | 64 | 37.034892 | 16.432682 | 33.399600 |
| 16384 | 1 | 1532.0 | 1.000000 | 1.913858 |
| 16384 | 4 | 383.002799 | 3.999971 | 7.655377 |
| 16384 | 16 | 117.962025 | 12.987230 | 24.855720 |
| 16384 | 64 | 83.630529 | 18.318669 | 35.059338 |
| 20480 | 1 | 2628.0 | 1.000000 | 2.179080 |
| 20480 | 4 | 657.329388 | 3.997996 | 8.711953 |
| 20480 | 16 | 266.598261 | 9.857529 | 21.480347 |
| 20480 | 64 | 144.768822 | 18.153080 | 39.557019 |
| 24576 | 1 | 4452.0 | 1.000000 | 2.222732 |
| 24576 | 4 | 1113.463342 | 3.998335 | 8.887230 |
| 24576 | 16 | 449.233085 | 9.910223 | 22.027773 |
| 24576 | 64 | 236.318336 | 18.838995 | 41.874045 |
| 28672 | 1 | 6868.0 | 1.000000 | 2.287981 |
| 28672 | 4 | 1717.0 | 4.000000 | 9.151924 |
| 28672 | 16 | 693.050973 | 9.909805 | 22.673446 |
| 28672 | 64 | 359.139040 | 19.123513 | 43.754234 |
| 32768 | 1 | 10356.0 | 1.000000 | 2.264991 |
| 32768 | 4 | 2589.0 | 4.000000 | 9.059964 |
| 32768 | 16 | 1044.0 | 9.919540 | 22.467671 |
| 32768 | 64 | 541.945194 | 19.108943 | 43.281587 |

Из полученных результатов можно сделать вывод, что когда размер матрицы довольно большой, а количество процессоров невелико - наблюдается линейное ускорение. Причем, наблюдается ухудшение ускорения для матрицы размером 4096 на 4096 на 64 ядрах по сравнению с 16. Это можно объяснить накладными расходами

на перессылки между процессорами. Максимальная производительность для этой задачи наблюдается на матрице 32768 на 32768, на 64 процессорах и равна $\sim$ 43 GFLOPS.