Отчёт: Миронов Сергей, Гмыря Михаил 20ПМИ-2

**Задание 1:**

**Описание алгоритмов:**

1. **Разбиение по строкам**

Пусть имеется подзадач. Размер матрицы , тогда у каждой подзадачи будет строк. Каждая подзадача умножает свою строку на вектор размере , в итоге у каждой подзадачи получается свой вектор размера . С помощью MPI\_Gather (или MPI\_Gatherv, если не делится на нацело) собираем все строки в нулевом процессе.

1. **Разбиение по столбцам**

Пусть имеется подзадач. Размер матрицы , тогда у каждой подзадачи будет столбцов. Каждая подзадача проходит по каждой строке и умножает свои столбцы на соответственные элементы вектора размера . Получается, что каждая подзадача имеет свою локальную сумму для каждой строки. С помощью MPI\_Reduce суммируем в подзадаче 0 каждую строку. В итоге в нулевом процессе будет собран результирующий вектор

1. **Разбиение по блокам**

Пусть имеется подзадач. Размер матрицы , ставим условие, что можно извлечь квадратный корень из и что делится без остатка. Получается у каждой подзадачи будет блок размера . Каждый процесс умножает свой блок на соответствующие элементы вектора размера . Для того чтобы обеспечить правильный выбор элементов для умножения, вычисляется для каждой подзадачи смещение в зависимости от номера процесса. В итоге у каждой подзадачи получается свой вектор размера . С помощью MPI\_Gather передаём эти вектора в нулевой процесс, который в зависимости от смещения каждого процесса суммирует их в результирующий вектор

**Формат ввода/вывода:**

Ввод: размер матрицы

Вывод: Матрица

**Исследование производительности**

**1)Алгоритм разбиения по строкам**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Время распределенной программы | | |
| 2 процесса | 4 процесса | 9 процессов |
| 1000 | 0.004 | 0.002 | 0.001 | 0.001 |
| 2000 | 0.015 | 0.007 | 0.004 | 0.003 |
| 3000 | 0.031 | 0.017 | 0.011 | 0.007 |
| 4000 | 0.054 | 0.029 | 0.021 | 0.013 |
| 5000 | 0.120 | 0.049 | 0.034 | 0.019 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Ускорение | | |
| 2 процесса | 4 процесса | 9 процессов |
| 1000 | 0.004 | 1.951 | 3.482 | 4.481 |
| 2000 | 0.015 | 2.068 | 3.680 | 5.675 |
| 3000 | 0.031 | 1.776 | 2.934 | 4.626 |
| 4000 | 0.054 | 1.885 | 2.597 | 4.081 |
| 5000 | 0.120 | 2.454 | 3.562 | 6.298 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Эффективность | | |
| 2 процесса | 4 процесса | 9 процессов |
| 1000 | 0.004 | 0.975 | 0.870 | 0.498 |
| 2000 | 0.015 | 1.034 | 0.920 | 0.631 |
| 3000 | 0.031 | 0.888 | 0.734 | 0.514 |
| 4000 | 0.054 | 0.942 | 0.649 | 0.453 |
| 5000 | 0.120 | 1.227 | 0.891 | 0.700 |

**2) Алгоритм разбиения по столбцам**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Время работы распределенной программы | | |
| 2 процесса | 4 процесса | 9 процессов |
| 1000 | 0.004 | 0.003 | 0.002 | 0.002 |
| 2000 | 0.015 | 0.009 | 0.008 | 0.006 |
| 3000 | 0.031 | 0.034 | 0.020 | 0.017 |
| 4000 | 0.054 | 0.047 | 0.038 | 0.030 |
| 5000 | 0.120 | 0.075 | 0.067 | 0.052 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Ускорение | | |
| 2 процесса | 4 процесса | 9 процессов |
| 1000 | 0.004 | 1.100 | 1.478 | 1.885 |
| 2000 | 0.015 | 1.609 | 1.753 | 2.521 |
| 3000 | 0.031 | 0.906 | 1.563 | 1.866 |
| 4000 | 0.054 | 1.159 | 1.412 | 1.830 |
| 5000 | 0.120 | 1.598 | 1.791 | 2.301 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Эффективность | | |
| 2 процесса | 4 процесса | 9 процессов |
| 1000 | 0.004 | 0.550 | 0.369 | 0.209 |
| 2000 | 0.015 | 0.805 | 0.438 | 0.280 |
| 3000 | 0.031 | 0.453 | 0.391 | 0.207 |
| 4000 | 0.054 | 0.579 | 0.353 | 0.203 |
| 5000 | 0.120 | 0.799 | 0.448 | 0.256 |

**3)Алгоритм разбиения по блокам**

|  |  |  |  |
| --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Время распределенной программы | |
| 4 процесса | 9 процессов |
| 1000 | 0.004 | 0.002 | 0.001 |
| 2000 | 0.015 | 0.009 | 0.006 |
| 3000 | 0.031 | 0.018 | 0.015 |
| 4000 | 0.054 | 0.038 | 0.034 |
| 5000 | 0.120 | 0.062 | 0.057 |

|  |  |  |  |
| --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Ускорение | |
| 4 процесса | 9 процессов |
| 1000 | 0.004 | 1.532 | 2.861 |
| 2000 | 0.015 | 1.693 | 2.579 |
| 3000 | 0.031 | 1.702 | 2.041 |
| 4000 | 0.054 | 1.414 | 1.594 |
| 5000 | 0.120 | 1.922 | 2.107 |

|  |  |  |  |
| --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Эффективность | |
| 4 процесса | 9 процессов |
| 1000 | 0.004 | 0.383 | 0.318 |
| 2000 | 0.015 | 0.423 | 0.287 |
| 3000 | 0.031 | 0.426 | 0.227 |
| 4000 | 0.054 | 0.354 | 0.177 |
| 5000 | 0.120 | 0.481 | 0.234 |

**Задание 2:**

**Описание алгоритма:**

Алгоритм Кэннона. Дано две матрицы и в предположении, что они квадратные размера . Необходимо найти результирующую матрицу . Имеется подзадач. Тогда у каждой подзадачи будет свой блок матрицы и свой блок матрицы размера . Пусть . Тогда количество блоков в одной строке решётки подзадач будет равнять . Тогда для каждого блока матрицы будет справедливо:

В результате инициализации блоков у нас получается решётка подзадач

Необходимо задать начальное распределение блоков

1. Для каждой строки решётки подзадач блоки матрицы сдвигаются на позиций влево
2. Для каждого столбца решётки подзадач блоки матрицы сдвигаются на позиций вверх

Для того чтобы сделать такое распределение используется циклический сдвиг по строкам и по столбцам для каждой матрицы. Для того чтобы обеспечить правильное взаимодействие между процессами используется смещение, которая для матрицы равняется , а для матрицы - . Такое взаимодействие можно осуществить с помощью MPI\_Sendrecv. После этого следует умножение блоков каждой подзадачи.

После задания такого распределения нужно сделать циклических сдвигов влево для каждой строки решётки подзадач матрицы и циклических сдвигов вверх каждого столбца решётки матрицы и последующим умножением блоков. В итоге у каждой подзадачи будет подсчитан блок результирующей матрицы . Далее необходимо сделать сборку с помощью MPI\_Gather и смещением для каждого потока.

**Формат ввода/вывода:**

Ввод: две матрицы размера

Вывод: матрица размера

**Исследование производительности:**

|  |  |  |  |
| --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Время распределенной программы | |
| 4 процесса | 9 процессов |
| 1000 | 5.791 | 2.297 | 1.742 |
| 1200 | 14.107 | 3.633 | 2.529 |
| 1400 | 26.356 | 6.507 | 5.476 |
| 1600 | 35.021 | 9.352 | 7.535 |
| 1800 | 66.476 | 14.448 | 10.049 |
| 2000 | 93.040 | 18.645 | 16.331 |

|  |  |  |  |
| --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Ускорение | |
| 4 процесса | 9 процессов |
| 1000 | 5.791 | 2.521 | 3.325 |
| 1200 | 14.107 | 3.883 | 5.578 |
| 1400 | 26.356 | 4.051 | 4.813 |
| 1600 | 35.021 | 3.745 | 4.648 |
| 1800 | 66.476 | 4.601 | 6.615 |
| 2000 | 93.040 | 4.990 | 5.697 |

|  |  |  |  |
| --- | --- | --- | --- |
| Размер матрицы | Время последовательной программы | Эффективность | |
| 4 процесса | 9 процессов |
| 1000 | 5.791 | 0.630 | 0.369 |
| 1200 | 14.107 | 0.971 | 0.620 |
| 1400 | 26.356 | 1.013 | 0.535 |
| 1600 | 35.021 | 0.936 | 0.516 |
| 1800 | 66.476 | 1.150 | 0.735 |
| 2000 | 93.040 | 1.248 | 0.633 |