**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
 РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет информационных технологий**

**Кафедра параллельных вычислений**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ**

«Параллельная реализация решения системы линейных алгебраических уравнений с помощью MPI»

студента 2 курса, группы 22210

**Лутцев Дмитрий Валерьевич**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

М. А. Мичуров

Новосибирск 2024

**ЗАДАНИЕ**

1. Написать программу на языке C или C++, которая реализует итерационный алгоритм решения системы линейных алгебраических уравнений вида Ax=b в соответствии с выбранным вариантом. Здесь A – матрица размером N×N, x и b – векторы длины N. Тип элементов – double.
2. Программу распараллелить с помощью MPI с разрезанием матрицы A по строкам на близкие по размеру, возможно не одинаковые, части. Соседние строки матрицы должны располагаться в одном или в соседних MPI-процессах. Реализовать два варианта программы:

• Вариант 1: векторы x и b дублируются в каждом MPI-процессе,

• Вариант 2: векторы x и b разрезаются между MPI-процессами аналогично матрице A. Уделить внимание тому, чтобы при запуске программы на различном числе MPI-процессов решалась одна и та же задача (исходные данные заполнялись одинаковым образом).

1. Замерить время работы двух вариантов программы при использовании различного числа процессорных ядер: 1,2, 4, 8, 16. Построить графики зависимости времени работы программы, ускорения и эффективности распараллеливания от числа используемых ядер. Исходные данные, параметры N и ε подобрать таким образом, чтобы решение задачи на одном ядре занимало не менее 30 секунд.
2. Выполнить профилирование двух вариантов программы с помощью MPE при использовании 16-и ядер.

**ОТЧЕТ**

В ходе работы была написана программа на языке Си, решающая СЛАУ вида A\*x = b методом простой итерации.

Параллельная программа была реализована двумя разными вариантами:

1. Матрица разрезалась по строкам.

2. Матрица разрезалась по строкам, векторы B и X дублировались между процессами.

Время исполнения каждой программы замерялось 5 раз, выбиралось минимальное.

Время выполнения:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 4 | 8 | 16 | 32 |
| Последовательная | 45,5 | 45,5 | 45,5 | 45,5 | 45,5 | 45,5 |
| MPI v1 | 45,4 | 22,8 | 11,7 | 6 | 3,4 | 3,3 |
| MPI v2 | 47,3 | 26,2 | 13,9 | 7,3 | 4,4 | 4,2 |

Из графика, время исполнения с разрезанием немногим больше времени

исполнения программы без разрезания векторов x и b.

Ускорение = время последовательной программы / время параллельной программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 4 | 8 | 16 | 32 |
| MPI v1 | 1 | 2 | 3,88 | 7,58 | 13,38 | 13,79 |
| MPI v2 | 0,96 | 1,73 | 3,27 | 6,23 | 10,34 | 10,83 |

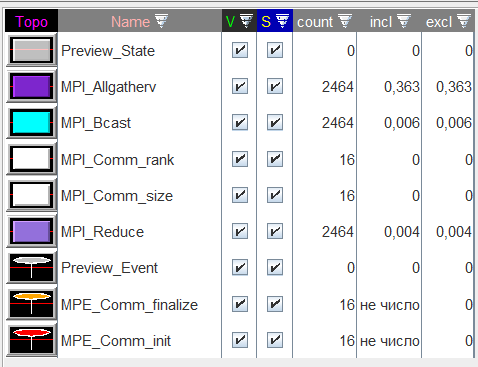
Эффективность = ускорение / кол-во ядер.

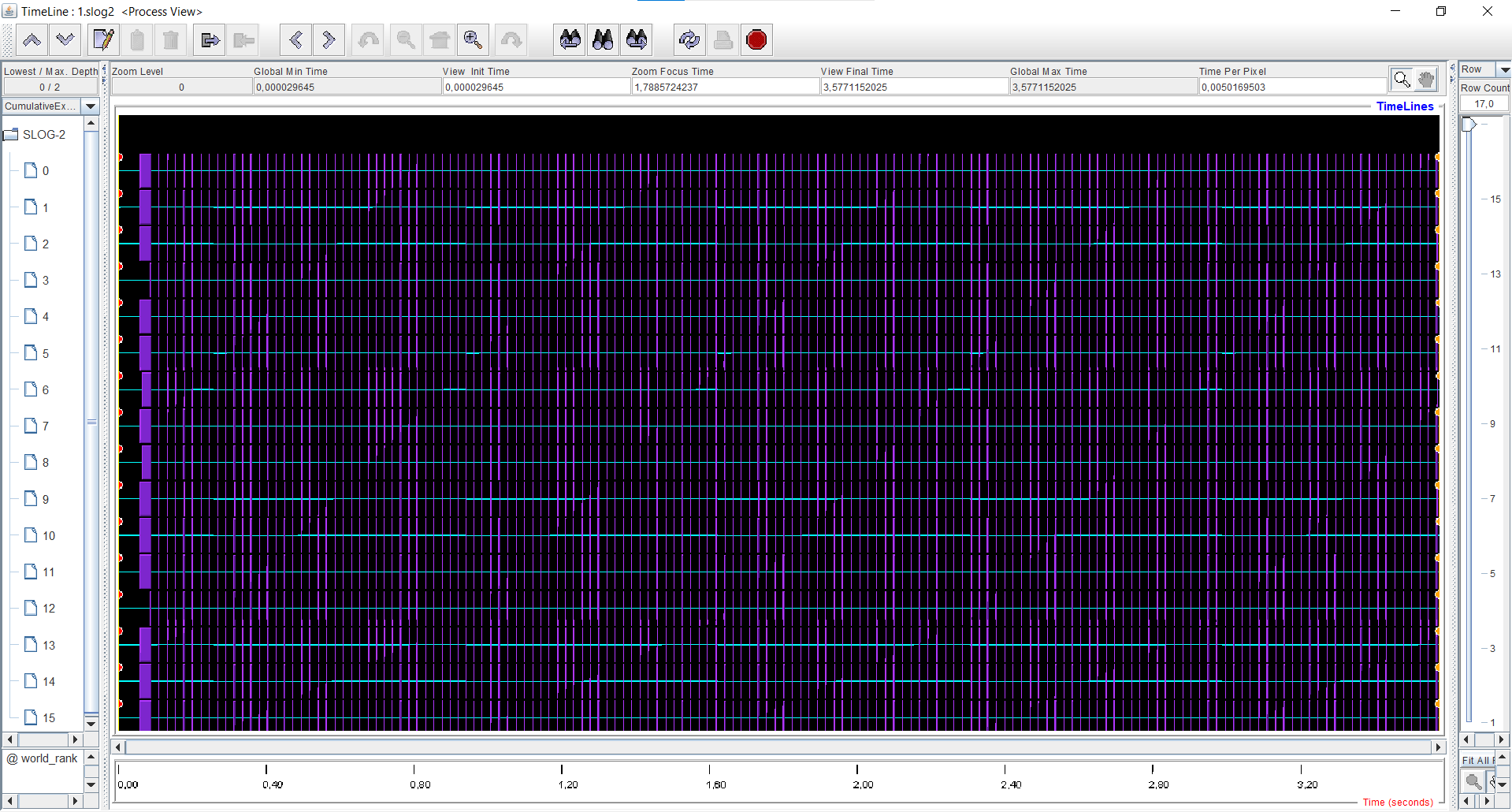
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 4 | 8 | 16 | 32 |
| MPI v1 | 1 | 1 | 0,97 | 0,95 | 0,83 | 0,43 |
| MPI v2 | 0,96 | 0,87 | 0,81 | 0,78 | 0,64 | 0,34 |

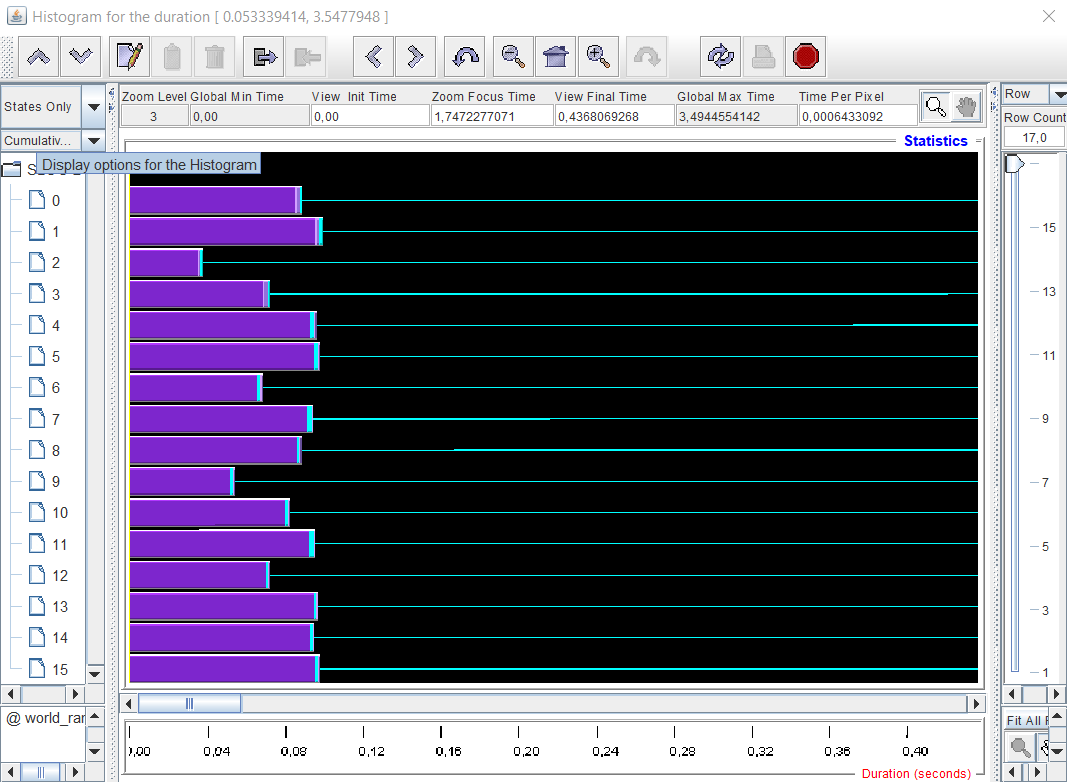
Было проведено профилирование двух вариантов реализации алгоритма на

16 процессах с помощью программы jumpshot.

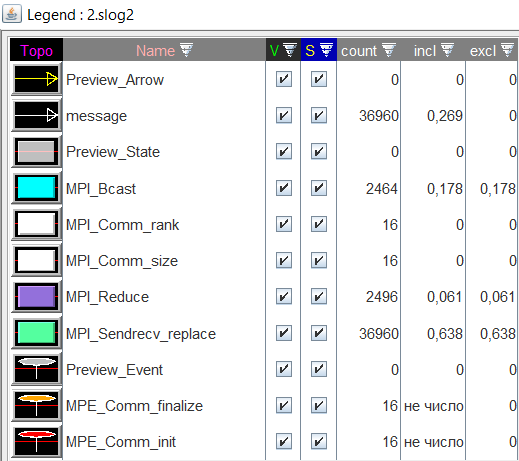
MPI v1:

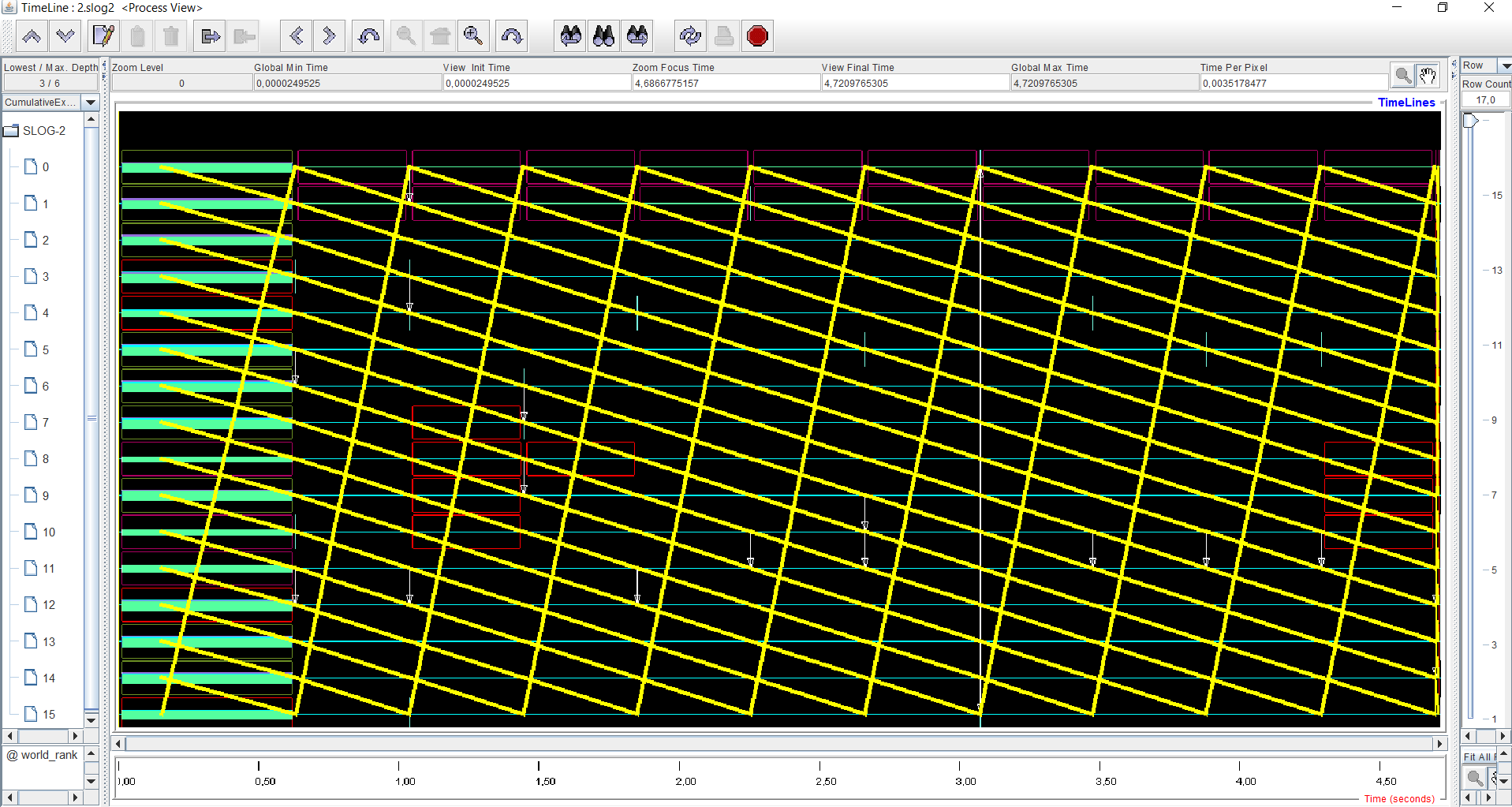


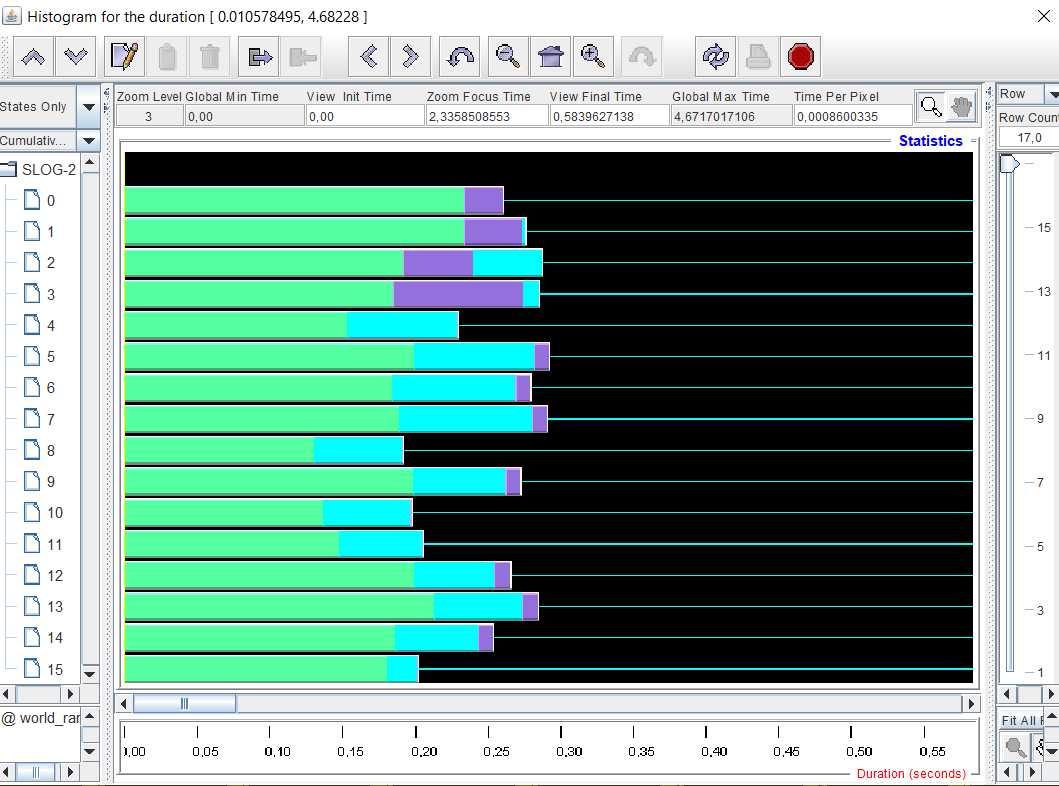




MPI v2:







Одним из преимуществ реализации алгоритма с разрезанием векторов

является меньшая требовательность к памяти у каждого процесса и у всех

процессов в целом, однако это нивелируется временем, затрачиваемым на

обмен частями вектора между процессами.

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы были изучена библиотека MPI, MPE и средство визуализации Jumpshot 4. Было выявлено,

что алгоритм решения системы линейных уравнений без разрезания

векторов x и b выполняется чуть быстрее и имеет большую эффективность.