Министерство науки и образования Украины

Харьковский национальный университет им. В.Н. Каразина

Факультет компьютерных наук

Отчет

по лабораторной работе №2

на тему:

**«Применение программной системы ParaLab для организации параллельной реализации методов матричного умножения»**

Выполнил:

Пойманов Павел

студент группы КС-42

Харьков 2017

# Цель работы

Закрепление знаний по системе ПараЛаб, представляющей собой интегрированную среду для изучения и исследования параллельных алгоритмов решения сложных вычислительных задач.

Обучение практическим приемам автоматической оценки и визуализации процесса выполнения эксперимента и анализа полученных результатов, позволяющих изучить эффективность использования тех или иных алгоритмов на разных вычислительных системах, сделать выводы о масштабируемости алгоритмов и определить возможное ускорение процесса параллельных вычислений.

# Задачи

1. Разработать последовательные Си-программу блочного умножения матриц.
2. Последовательно выполните вычислительные эксперименты с использованием метода Фокса и метода Кэннона. Сравните временные характеристики этих экспериментов.
3. Проведите серию вычислительных экспериментов. Постройте графики зависимости времени выполнения алгоритма (ускорение и эффективности) от объема исходных данных и от количества процессоров.
4. Сделайте выводы.

|  |  |
| --- | --- |
|  | Листинг алгоритма ленточного алгоритма …  for(i = 0; i < m; i++)      for(j = 0; j < q; j++)      {          C[i][j] = 0;          for(k = 0; k < n; k++)              C[i][j] += A[i][k] \* B[k][j];      } |

…

## Листинг блочного алгоритма

void Rec\_Mult(int \*C, const int \*A, const int \*B, int n, int rowsize)

{

if (n == 2)

{

const int d11 = 0;

const int d12 = 1;

const int d21 = rowsize;

const int d22 = rowsize + 1;

C[d11] += A[d11] \* B[d11] + A[d12] \* B[d21];

C[d12] += A[d11] \* B[d12] + A[d12] \* B[d22];

C[d21] += A[d21] \* B[d11] + A[d22] \* B[d21];

C[d22] += A[d21] \* B[d12] + A[d22] \* B[d22];

}

else

{

const int d11 = 0;

const int d12 = n / 2;

const int d21 = (n / 2) \* rowsize;

const int d22 = (n / 2) \* (rowsize + 1);

Rec\_Mult(C + d11, A + d11, B + d11, n / 2, rowsize);

Rec\_Mult(C + d11, A + d12, B + d21, n / 2, rowsize);

Rec\_Mult(C + d12, A + d11, B + d12, n / 2, rowsize);

Rec\_Mult(C + d12, A + d12, B + d22, n / 2, rowsize);

Rec\_Mult(C + d21, A + d21, B + d11, n / 2, rowsize);

Rec\_Mult(C + d21, A + d22, B + d21, n / 2, rowsize);

  Rec\_Mult(C + d22, A + d21, B + d12, n / 2, rowsize);

Rec\_Mult(C + d22, A + d22, B + d22, n / 2, rowsize);

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| Порядок матрицы | Последовательный алгоритм ленточный  Время выполнения |  |  |
| 1000 | 26,70700 |  |  |

Вычисления Paralab:

Топология ГиперКуб

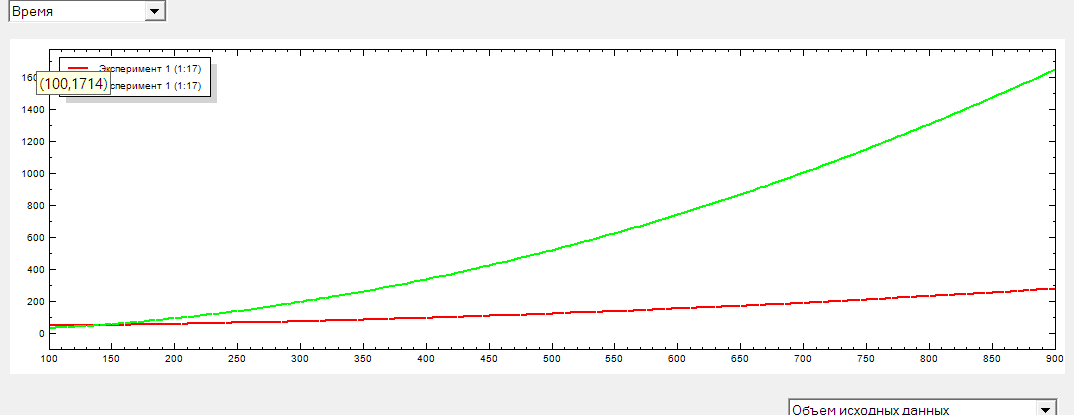


Рисунок 1 Зависимость время от данных

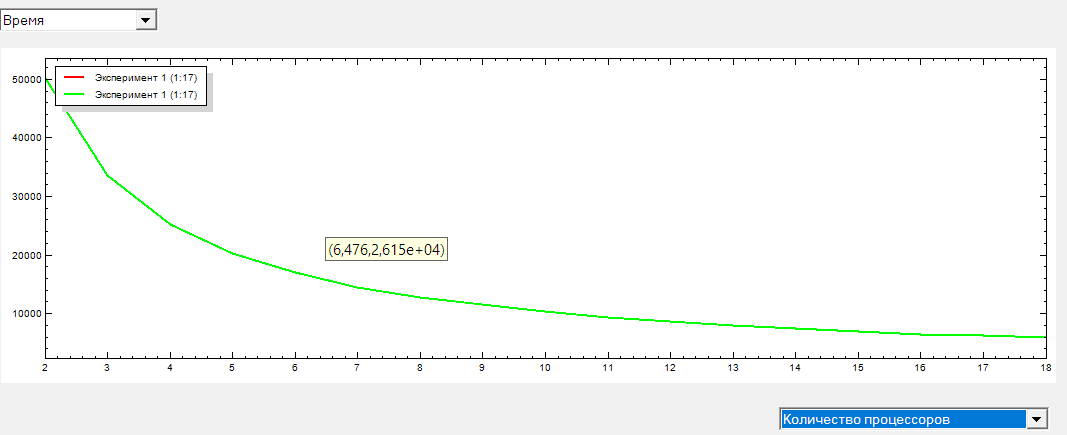


Рисунок 2 Зависимость время от кол-ва проц.

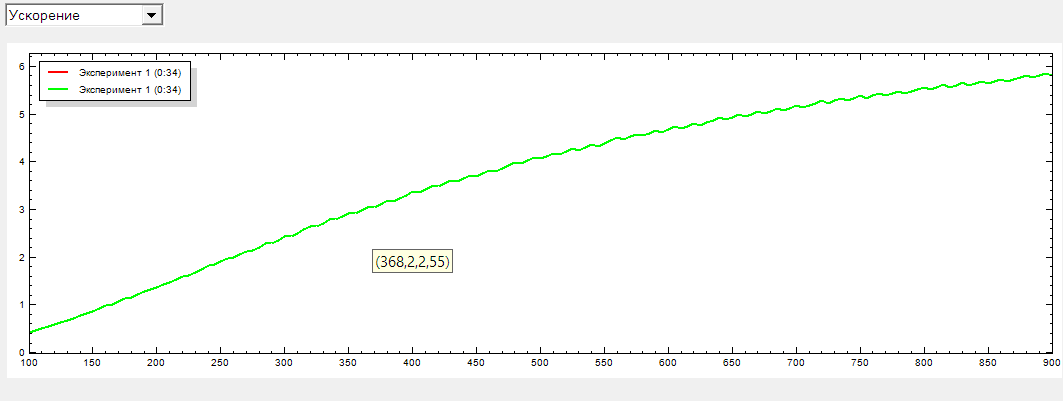


Рисунок 3 Зависимость ускорения от данных

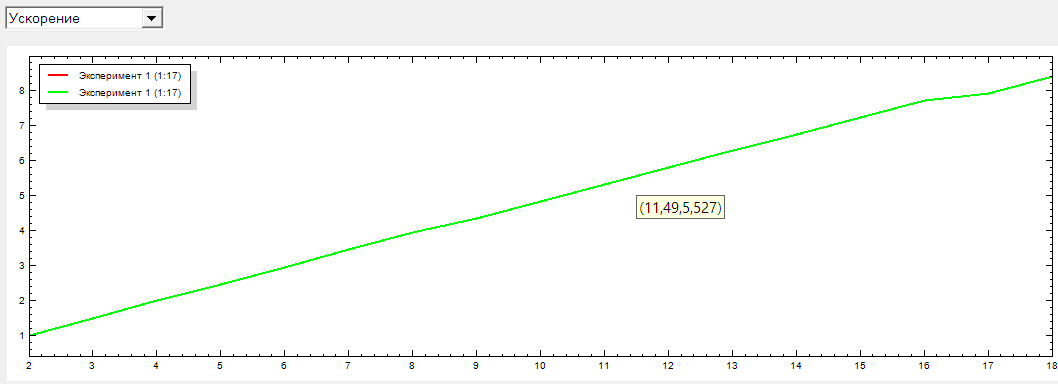


Рисунок 4 Зависимость ускорения от кол-ва проц

# Выводы

Выполнив вычисления на программной системе Paralab я закрепил знания о системе Paralab. Выполнив анализ можно сказать следующее, что выполнения ряда математических операций таких как перемножение матриц задача весьма сложная даже для современных процессоров, но применив алгоритмы параллельных вычислений задача выполняется в разы быстрее. Судя по графикам можно сказать что увеличение числа процессоров может быть оправдано, хотя в какой-то момент времени и не целесообразно. Во всех проведенных экспериментах ускорение системы всегда возрастало и только в какой-то момент незначительно уменьшило свой линейный рост.