

## Лабораторная работа № 3

### Знакомство с технологией MPI

#### Цель:

Ознакомиться с технологией MPI. Научиться компилировать и запускать MPI программы. Получить навык работы с простейшими средствами передачи сообщений между процессами.

#### Теоретические сведения

MPI — программный интерфейс (API) для передачи информации, который позволяет обмениваться сообщениями между экземплярами программы, выполняющими одну задачу (которые могут быть запущенными на различных комп

MPI является наиболее распространённым стандартом интерфейса обмена данными в параллельном программировании, существуют его реализации для большого числа компьютерных платформ. В настоящее время существует большое количество бесплатных и коммерческих реализаций MPI.

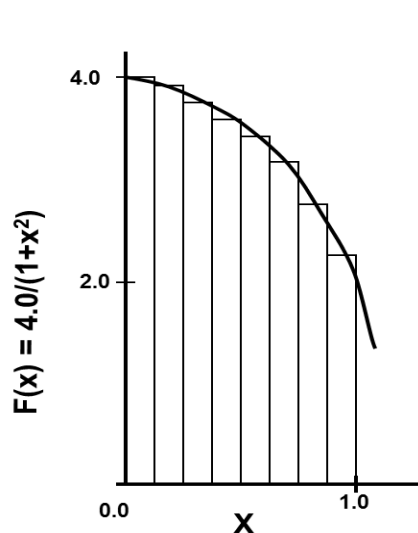
Практически все реализации MPI представляют собой внешнюю подключаемую библиотеку. В связи с этим, при компилировании MPI программ, компилятору необходимо дополнительно указывать заголовочные и библиотечные файлы.

Подробно со всеми директивами и их спецификациями можно ознакомиться в опубликованном стандарте MPI [1, а также в учебном пособии Антонова А. С. «Параллельное программирование с использованием технологии MPI» [6].

#### Примеры программ

Рассмотрим пример параллельной программы для расчета числа  $\pi$ /

#### Вычисление числа $\pi$



$$\int_0^1 \frac{4.0}{(1+x^2)} dx = \pi$$

Мы можем аппроксимировать интеграл как сумму прямоугольников:

$$\sum_{i=0}^N F(x_i) \Delta x \approx \pi$$

Где каждый прямоугольник имеет ширину  $\Delta x$  и высоту  $F(x_i)$  в середине интервала

```

// Подключаем необходимые заголовки
#include "mpi.h"
#include <stdio.h>
#include <math.h>
// Функция для промежуточных вычислений
double f(double a)
{
    return (4.0 / (1.0 + a*a));
}
// Главная функция программы
int main(int argc, char *argv[])
{
    // Объявляем переменные
    int done = 0, n, myid, numprocs, i;
    double PI25DT = 3.141592653589793238462643;
    double mypi, pi, h, sum, x;
    double startwtime = 0.0, endwtime;
    int namelen;
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    // Инициализируем подсистему MPI
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);
    MPI_Get_processor_name(processor_name, &namelen);
    /*
    // Выводим номер потока в общем пуле
    fprintf(stdout, "Process %d of %d is on %s\n",
    myid, numprocs, processor_name);
    fflush(stdout);
    */
    while (!done) {
        // Если это головной процесс, спрашиваем пользователя о
        // количестве интервалов
        if (myid == 0) {
            fprintf(stdout, "Enter the number of intervals: (0 quits)
            ");
            fflush(stdout);
            if (scanf("%d", &n) != 1) {
                fprintf(stdout, "No number entered; quitting\n");
                n = 0;
            }
            startwtime = MPI_Wtime();
        }
    }

```

```

// Рассылаем количество интервалов всем процессам
// (в том числе и себе самому)
MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
if (n == 0)
done = 1;
else {
h = 1.0 / (double) n;
sum = 0.0;
// Обсчитывает точки, "закрепленные" за процессом
for (i = myid + 1; i <= n; i += numprocs) {
x = h * ((double)i - 0.5);
sum += f(x);
}
mypi = h * sum;
// Собираем результаты со всех процессов и складываем все
MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);
// Если это головной процесс, выводим полученные результаты
if (myid == 0) {
printf("pi is approximately %.16f, Error is %.16f\n", pi,
fabs(pi - PI25DT));
endwtime = MPI_Wtime();
printf("wall clock time = %f\n", endwtime-startwtime);
fflush( stdout );
}
}
}
// Освобождаем подсистему MPI
MPI_Finalize();
return 0;

```

## Задание к выполнению лабораторной работы

Ознакомиться со средствами для организации параллельного выполнения программы, предоставляемыми технологией MPI.

Разработать параллельную программу, выполняющую решение задачи согласно варианту.

Выполнить замеры времени решения задачи параллельным и последовательным вариантами программы. Сравнить полученные результаты, построить графики и объяснить их.

Выполнить замеры времени решения задачи параллельным и последовательным вариантами программы. Сравнить полученные результаты, построить графики и объяснить их.

### Задание1:

1. Разработать ПО с использованием технологии MPI для расчета числа  $\pi$  на ресурсе БГТУ имени В.Г. Шухова.
2. Выполнить расчеты на ресурсе, согласно своего варианта (N- номер варианта).
3. Заполнить таблицы и построить графики.

Таблица №1. Расчет числа  $\pi$ .

Номер расчета	Количество итераций	на 1 узле		на 8 узлах		на 16 узлах		на 24 узлах	
		Значение числа $\pi$	Время расчета	Значение числа $\pi$	Время расчета	Значение числа $\pi$	Время расчета	Значение числа $\pi$	Время расчета
1	N- номер варианта								
2	$N \cdot 10^2$								
3	$N \cdot 10^4$								
4	$N \cdot 10^6$								
5	$N \cdot 10^8$								
...									

Таблица №2. Коэффициент ускорения ( $K_i = T_i^1 / T_i^j$ , где  $i$  – номер расчета,  $j$  - кол-во узлов)

№	P – количество расчетов	Коэффициент ускорения K на 1 узле	Коэффициент ускорения K на 8 узлах	Коэффициент ускорения K на 16 узлах	Коэффициент ускорения K на 24 узлах
1	N				
2	$N \cdot 10^2$				
3	$N \cdot 10^4$				
4	$N \cdot 10^6$				
5	$N \cdot 10^8$				
i	...				

Построить графики зависимости коэффициента ускорения (ось у) от количества ядер (ось х) для разных значений Р (количество расчетов).

### Задание2:

Разработайте программу с использованием технологии MPI на ресурсе БГТУ имени В.Г. Шухова., для вычисления значения функции  $f(x)$  на отрезке  $[1, N+1]$  с шагом  $h=N/P$ , где N – номер варианта и составьте таблицы №1 и №2.

Таблица №1. Время расчета значений функции.

№	Р – количество расчетов	время расчета на 1 узле	время расчета на 8 узлах	время расчета на 16 узлах	время расчета на 24 узлах
1	N				
2	$N \cdot 10^2$				
3	$N \cdot 10^4$				
4	$N \cdot 10^6$				
i	...				

Таблица №2. Коэффициент ускорения ( $K_i = T_i^1 / T_i^j$ , где i – номер расчета, j - кол-во узлов)

№	Р – количество расчетов	Коэффициент ускорения K на 1 узле	Коэффициент ускорения K на 8 узлах	Коэффициент ускорения K на 16 узлах	Коэффициент ускорения K на 24 узлах
1	N				
2	$N \cdot 10^2$				
3	$N \cdot 10^4$				
4	$N \cdot 10^6$				
i	...				

Построить графики зависимости коэффициента ускорения (ось у) от количества ядер (ось х) для разных значений Р (количество расчетов).

$$a = N, b = N \cdot 2, k = N/2, z = N^2$$

$$1. \quad y = \sqrt{x-1} + \frac{1}{x-3};$$

$$2. \quad y = \frac{1}{k \cdot \sqrt{2\pi}} * e^{\frac{-(x-a)^2}{2k^2}};$$

$$3. \quad y = \frac{1}{2b} * e^{\frac{-|x-a|}{b}};$$

$$4. \quad y = \frac{\sin x}{2 \cos^2 x} - \cos x - \frac{3}{2} \operatorname{tg} x;$$

$$5. \quad y = \cos x (\ln |2 - e^{-|a+x|}|);$$

$$6. \quad y = \frac{e^{\sin^2 x} + \ln |\operatorname{tg} x|}{\sin x};$$

$$7. \quad y = a * e^{-ax} * \sin x;$$

$$8. \quad y = \frac{\sin^3 |ax^3 + bx^2 - ab|}{\sqrt{|ax^3 + bx^2 - ab|}};$$

$$9. \quad y = \frac{\sqrt{|\cos x|}}{\sqrt{1+x^2}};$$

$$10. \quad y = \frac{1 + \sin \sqrt{x+1}}{\cos(12z-4)};$$

$$11. \quad y = x - \frac{x^3}{3} + \frac{x^5}{5};$$

$$12. \quad y = \frac{x^2 - 7x + 10}{x^2 - 8x + 12};$$

$$13. \quad y = \frac{\cos x}{a-2x} + 16x \cdot \cos(xz) - 2;$$

$$14. \quad y = |x^2 - x^3| - \frac{7x}{x^3 - 15x};$$

$$15. \quad y = e^{-x} - \cos x + \sin(2xz);$$

$$16. \quad y = e^{\operatorname{tg} x} + \cos(x-z);$$

$$17. \quad y = \sin \sqrt{x+1} - \sin \sqrt{x-1};$$

$$18. \quad y = e^x - \frac{z^2 + 12xz - 3x^2}{18z-1};$$

$$19. y = x - 10 \sin x + |x^2 - xe^5|;$$

$$20. y = x - 10e^{\sin x} + \cos(x - z);$$

$$21. y = 2ctg(3x) - \frac{1}{12x^2 + 7x - 5};$$

$$22. y = e^x - x - 2 + (1 + x)^2;$$

$$23. y = 2ctg(3x) - \frac{\sin \cos x}{\sin(1 + x^2)};$$

$$24. y = e^x - 4x + (z - \sqrt{|x|});$$

$$25. y = \left( \frac{z+1}{x-1} \right)^2 + 18xz^2;$$

$$26. y = \left( 1 + \frac{1}{x^2} \right)^2 - 12x^2z;$$

### ***Содержание отчета***

1. Название и цель работы.
2. Задание к выполнению лабораторной работы согласно варианту.
3. Описание алгоритма решения задачи в виде блок-схем. Описание используемых параллельных методов вычислений.
4. Программа в виде исходных кодов (с поясняющими комментариями), а также в откомпилированном виде для демонстрации на кластере **БГТУ имени В.Г. Шухова**.
5. Примеры работы программы на тестовых данных (копии экранов).
6. Таблицы и графики, полученные в результате расчета.
7. Выводы по работе.