

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

Институт информатики, математики и электроники  
Факультет информатики  
Кафедра технической кибернетики

ЛАБОРАТОРНАЯ РАБОТА №1

**Запуск параллельных программ на OpenMP и MPI**

по курсу  
Параллельное программирование



Студент группы 6407

Преподаватель

Исаев М.А.

Козлова Е.С.

Самара 2018

## ЗАДАНИЕ



Произвести запуск программ «Hello world», которые используют технологии MPI и OpenMP, на различном количестве процессоров (потоков). В ходе анализа работы программы, оценить время ее выполнения на 16 исполняющих нитях (процессорах).

# СОДЕРЖАНИЕ



ВВЕДЕНИЕ .....	4
ЗАПУСК ПРОГРАММ MPI И OPENMP .....	5
Технология OpenMP .....	5
Технология MPI .....	5
ЗАКЛЮЧЕНИЕ .....	7
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	8
ПРИЛОЖЕНИЕ А .....	9
Код программ на OpenMP и MPI .....	9

## ВВЕДЕНИЕ

В связи с развитием науки появились задачи, для решения которых требуются компьютеры с большими мощностями. Увеличение производительности ПК не позволило решать многие проблемы. Это послужило причиной возникновения суперкомпьютеров и параллельных алгоритмов.

На сегодняшний день наиболее развивающимся направлением является введение параллелизма в основу вычислительной системы.

Ключевой идеей распараллеливания является представление задачи в виде набора меньших задач, которые могут решаться одновременно.

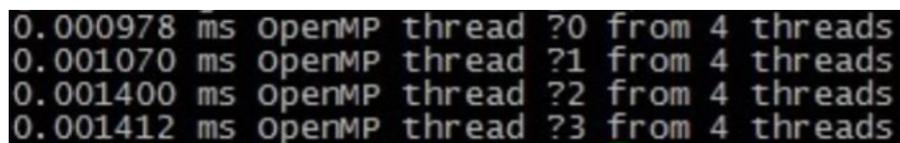
В данной работе необходимо в качестве знакомства с технологиями распараллеливания изучить код программы, известной как «Hello World».

## ЗАПУСК ПРОГРАММ MPI И OPENMP

Доступ к серверам был получен из командной строки через протокол SSH. Для обмена файлами между компьютером и сервером использовался протокол SFTP. Запуск программы для MPI производится с помощью системы пакетного запуска PBS Torque.

### Технология OpenMP

Для компиляции файлов с кодом подгружаем модуль, используя следующую команду `module load intelicc16`. Для компиляции программы используются следующая команда `icc -qopenmp -o <название исполняемого файла> omp.c`. Запуск программы для OpenMP осуществляется запуском скомпилированного файла из командной строки `./omp` [1]. На рисунке 1 приведен результат работы программы с использованием данной технологии.



```
0.000978 ms OpenMP thread ?0 from 4 threads
0.001070 ms OpenMP thread ?1 from 4 threads
0.001400 ms OpenMP thread ?2 from 4 threads
0.001412 ms OpenMP thread ?3 from 4 threads
```

Рисунок 1 - Результат выполнения программы OpenMP

### Технология MPI

Для компиляции файлов с кодом подгружаем модуль, используя следующую команду `module load impi/4`. Для компиляции программы используются следующая команда `mpicc -o <название исполняемого файла> mpi.c`. Далее был создан скрипт с указанием исполняемого файла. В командной строке запускаем задачу следующим образом `qsub <имя PBS скрипта>`. На рисунке 2 представлены результаты MPI на нескольких процессах.

```
===== Begin prologue script =====  
End prologue script =====  
Hello world from process 0 from total number of 20.000011 ms  
Hello world from process 1 from total number of 20.000012 ms
```

Рисунок 2 - Результат выполнения программы MPI на двух нитях



Результаты измерений показывают, что на выполнение задачи вывода текстового сообщения в стандартный поток тратится очень мало времени. Результаты полученные с помощью технологии OpenMP несколько отличаются. Вероятно, такое отличие обусловлено тем что потоки выполняются асинхронно.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы мы получили представления о работе на многопроцессорном устройстве, познакомились с технологиями MPI и OpenMP, запустили программу «Hello world» на кластере с использованием данных технологий. Также оценили время выполнения программы на разном количестве процессоров. Сравнив процессоры на данных технологиях, можно сказать, что использование технологии OpenMP работает быстрее, чем MPI.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Козлова Е.С. Методические указания к лабораторной работе №1 по курсу «Параллельное программирование» Методические указания к лабораторным работам / Сост. Козлова Е.С. – Самара, 2017. 12 с.

## ПРИЛОЖЕНИЕ А

### Код программ на OpenMP и MPI

Код OpenMP:

```
#include <math.h>
#include <omp.h>
#include <stdlib.h>
#include <locale.h>
#include <stdio.h>
int main(int argc, char* argv[])
{
    omp_set_num_threads(4);
    double start, end;
    start = omp_get_wtime();
    int nThreads, theadNum;
    #pragma omp parallel private(nThreads, theadNum)
    {
        nThreads = omp_get_num_threads();
        theadNum = omp_get_thread_num();
        end = omp_get_wtime();
        printf("%lf ms ", end - start);
        printf("OpenMP thread ?%d from %d threads \n", theadNum, nThreads);
    }
    return 0;
}
```

Код MPI:

```
#include "mpi.h"
#include "stdio.h"
#include <time.h>
#include <math.h>
#include <omp.h>
int main(int argc, char* argv[])
{
    int rank, ranksize, i;
    double start, end;
    MPI_Init(&argc, &argv);
    start = MPI_Wtime();
    unsigned int st_time = clock();
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &ranksize);
    printf("Hello world from process %d from total number of %d\n", rank, ranksize);
    end = MPI_Wtime();
    unsigned int end_time = clock();
}
```

```
printf("%lf ms ", end - start);
MPI_Finalize();
return 0;
}
PBS крипт:
#!/bin/bash
#PBS -N MPI_mapo_8
#PBS -l walltime=00:01:10
#PBS -l nodes=1:ppn=8
#PBS -j oe
#PBS -A tk
cd $PBS_O_WORKDIR
module load impi/4
export I_MPI_DEVICE=rdma
export I_MPI_DEBUG=0
export I_MPI_FALLBACK_DEVICE=disable
mpirun -r ssh -machinefile $PBS_NODEFILE -np $PBS_NP ./MPI
```