Федеральное государственное автономное образовательное учреждение

высшего образования

"Российский университет транспорта" (МИИТ)

Кафедра «Цифровые технологии управления транспортными процессами»

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

**ПО ДИСЦИПЛИНЕ**

**«Высокопроизводительные вычислительные системы на транспорте»**

Выполнил ст. г. УВА-411

Журавлёв К. Д.

Принял:

Варфоломеев В. А.

Москва  
2022

**Характеристики компьютера:**

Процессор - Intel® Core™ 3i-10100 CPU @ 3.60GHz 3.60GHz;

Количество ядер – 4, количество логических ядер – 8;

Объём оперативной памяти – 8Гб;

**Параметры вычислительной сети:**

Тип сети – локальная сеть;

Номинальная пропускная способность – 100Mbps

Адреса и имена узлов:

IPv4-адрес 10.242.120.185/24 – Desktop-L4VP1GT

IPv4-адрес 10.242.120.186/24 – Desktop-CEI7FJ5

**Текст программы MPI:**

#include "mpi.h"

#include

#include

// Главная функция программы (выполняется всеми процессами)

int main(int argc, char \*\*argv) {

// Объявление переменных

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];

// Инициализация подсистемы MPI MPI\_Init(&argc, &argv);

// Получить общее число процессов numprocs в рамках задачи) MPI\_Comm\_size(MPI\_COMM\_WORLD,&numprocs);

// Получить номер myid текущего процесса MPI\_Comm\_rank(MPI\_COMM\_WORLD,&myid);

MPI\_Get\_processor\_name(processor\_name,&namelen);

// Вывод номера процесса в общем пуле

fprintf(stdout, "Process %d of %d is on %s\n", myid,numprocs,processor\_name);

fflush(stdout); while(!done) { // количество 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();

}

// Рассылка количества интервалов всем процессам (в том числе и себе)

MPI\_Bcast(&n, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

if(n==0) done = 1; else { h = 1.0 / (double) n; sum = 0.0;

// Расчет площадей прямоугольников, закрепленных за процессом

for(i = myid + 1 ; (i <= n) ; i += numprocs) {

x = h \* ((double)i - 0.5);

sum += (4.0/(1.0 + x\*x)); }

mypi = h \* sum;

// Передача 0-му процессу и итоговое суммирование частных сумм

// (mypi)процессов в переменной pi

MPI\_Reduce(&mypi, &pi, 1, MPI\_DOUBLE, MPI\_SUM, 0, MPI\_COMM\_WORLD);

// Если это главный процесс, вывод полученного результата

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", endwtimestartwtime);

fflush(stdout);

} } }

// Освобождение подсистемы MPI

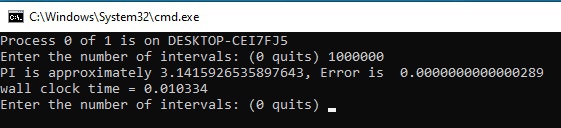
MPI\_Finalize();

return 0;

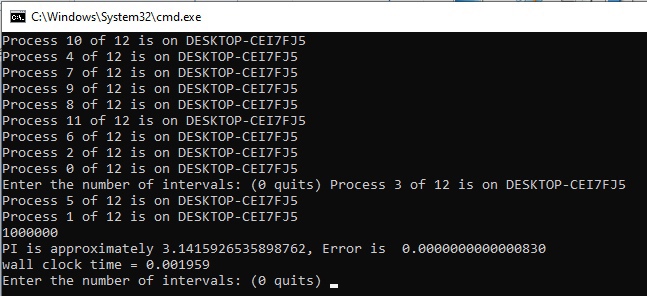
}

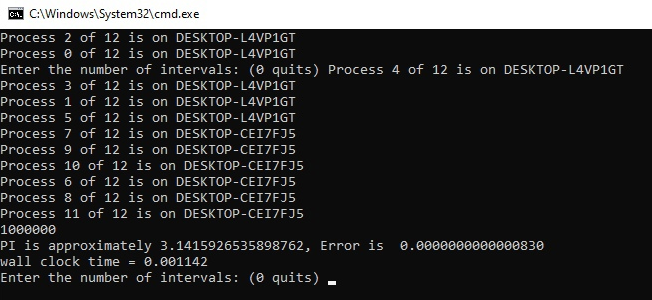
**Результаты работы:**

**Режим А**



**Режим В**



**Режим С**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сложность задачи (млн. итераций) | Время выполнения, с | | | | | | | | | |
| Режим А | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0,010374 | 0,0103 | 0,010295 | 0,010649 | 0,010329 | 0,010511 | 0,010376 | 0,01042 | 0,010376 | 0,010361 |
| 5 | 0,051172 | 0,051963 | 0,052031 | 0,051771 | 0,051704 | 0,051926 | 0,051798 | 0,051271 | 0,051879 | 0,05197 |
| 10 | 0,103007 | 0,103617 | 0,103499 | 0,103875 | 0,103072 | 0,103608 | 0,103279 | 0,103221 | 0,103419 | 0,103264 |
| 50 | 0,515774 | 0,518019 | 0,517717 | 0,513507 | 0,517348 | 0,517243 | 0,511136 | 0,512922 | 0,51458 | 0,512467 |
| 100 | 1,023828 | 1,031882 | 1,022064 | 1,045871 | 1,041623 | 1,026131 | 1,02218 | 1,023498 | 1,033654 | 1,030317 |
| 500 | 5,119205 | 5,108023 | 5,187529 | 5,119368 | 5,12654 | 5,141263 | 5,149236 | 5,142218 | 5,135718 | 5,165574 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сложность задачи (млн. итераций) | Время выполнения, с | | | | | | | | | |
| Режим B | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0,001938 | 0,001931 | 0,001846 | 0,00184 | 0,00187 | 0,001895 | 0,00188 | 0,001849 | 0,00186 | 0,001855 |
| 5 | 0,008993 | 0,009169 | 0,009134 | 0,009019 | 0,009043 | 0,008945 | 0,009017 | 0,00923 | 0,008978 | 0,00897 |
| 10 | 0,018015 | 0,017987 | 0,017968 | 0,01813 | 0,017832 | 0,017911 | 0,017847 | 0,018055 | 0,01809 | 0,018033 |
| 50 | 0,084014 | 0,084778 | 0,089484 | 0,083866 | 0,083573 | 0,083872 | 0,083135 | 0,08374 | 0,083804 | 0,084447 |
| 100 | 0,171923 | 0,153354 | 0,172205 | 0,172338 | 0,172366 | 0,147585 | 0,14893 | 0,147602 | 0,17243 | 0,171895 |
| 500 | 0,800279 | 0,79821 | 0,80351 | 0,797819 | 0,767931 | 0,799373 | 0,802894 | 0,804131 | 0,799525 | 0,798548 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сложность задачи (млн. итераций) | Время выполнения, с | | | | | | | | | |
| Режим C | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0,001142 | 0,001192 | 0,00111 | 0,001119 | 0,001076 | 0,001095 | 0,001135 | 0,001074 | 0,001151 | 0,001094 |
| 5 | 0,004619 | 0,004682 | 0,004627 | 0,004708 | 0,004679 | 0,004682 | 0,004574 | 0,004627 | 0,004695 | 0,004631 |
| 10 | 0,009153 | 0,009166 | 0,009049 | 0,009091 | 0,009182 | 0,009019 | 0,0091 | 0,009159 | 0,009059 | 0,009129 |
| 50 | 0,045687 | 0,045137 | 0,044949 | 0,044646 | 0,044941 | 0,044842 | 0,04474 | 0,044718 | 0,04455 | 0,044604 |
| 100 | 0,089858 | 0,089747 | 0,089211 | 0,089559 | 0,089167 | 0,089345 | 0,0888 | 0,088993 | 0,089055 | 0,088928 |
| 500 | 0,44729 | 0,444738 | 0,446336 | 0,44464 | 0,444755 | 0,443478 | 0,44363 | 0,445318 | 0,444488 | 0,443626 |

**Эксперимент 1 – изменение количества итераций при заданном числе процессов**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Сложность задачи (млн. итераций) | Среднее время выполнения, с Заданное число процессов = 12 | | | Показатели эффективности | | |
| Один узел | | 2 узла | S | E | C |
| Режим А | Режим В | Режим С |
| 1 | 0,0103991 | 0,0018764 | 0,0011188 | 9,294869503 | 0,580929344 | 0,0179008 |
| 5 | 0,0517485 | 0,0090498 | 0,0046524 | 11,12296879 | 0,695185549 | 0,0744384 |
| 10 | 0,1033861 | 0,0179868 | 0,0091107 | 11,34776691 | 0,709235432 | 0,1457712 |
| 50 | 0,5150713 | 0,0844713 | 0,0448814 | 11,47627525 | 0,717267203 | 0,7181024 |
| 100 | 1,0301048 | 0,1630628 | 0,0892663 | 11,53968295 | 0,721230184 | 1,4282608 |
| 500 | 5,1394674 | 0,797222 | 0,4448299 | 11,55378134 | 0,722111334 | 7,1172784 |

**Эксперимент 2 – изменение количества процессов при заданном количестве итераций**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Число параллельных процессов | Среднее время выполнения, с Заданное число итераций = 30 млн. | | Показатели эффективности | | |
| Один узел | 2 Узла | S | E | C |
| Режим B | Режим C |
| 1 | 0,306294 | 0,307886 | 0,994829255 | 0,062176828 | 4,926176 |
| 2 | 0,156195 | 0,155552 | 1,969077865 | 0,123067367 | 2,488832 |
| 4 | 0,079447 | 0,079315 | 3,861741159 | 0,241358822 | 1,26904 |
| 8 | 0,040763 | 0,040751 | 7,51623273 | 0,469764546 | 0,652016 |
| 16 | 0,060643 | 0,021522 | 14,23166992 | 0,88947937 | 0,344352 |
| 32 | 0,072377 | 0,030604 | 10,00829957 | 0,625518723 | 0,489664 |

**Вывод:**

По результатам эксперимента 1 можно увидеть, что время выполнения и стоимость вычислений при заданном количестве процессов увеличивается во столько же раз, во сколько увеличилось количество итераций. Коэффициент эффективности и коэффициент ускорения вычислений при увеличении количества итераций постепенно выходят на плато.

По результатам эксперимента 2 можно увидеть, что самым эффективным и оптимальным выбором количества параллельных процессов будет сумма количества логических ядер на всех узлах кластера, выставлять значения количества параллельных процессов больше, чем эта сумма, не имеет смысла, так как по результатам время только увеличилось, так же, как и стоимость вычислений. Коэффициент эффективности и коэффициент ускорения вычислений уменьшаются во столько же раз, во сколько увеличилось количество параллельных процессов, но до тех пор, пока их количество не достигнет суммы логических ядер на всех узлах.