**Национальный Исследовательский Университет Информационных технологий, Механики и Оптики**

**Лабораторная работа № 1 по курсу «Параллельные вычисления»**

Выполнил:

Бедаш Д.С., гр. 5110

Проверил:

Соснин Владимир Валерьевич

Задание:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ФИО студента** | **Математическое ожидание** | **Дисперсия** | **Преобразование исходной матрицы** | **Способ формирования вектора с результатами** | **Метод сортировки вектора с результатами** |
| Бедаш Дмитрий Сергеевич | -45 | 65 | Умножение М1 на скаляр 5. | Выборочная дисперсия каждой строки | Shell sort |

1. На **языке Cи** написать консольную программу lab1.c, решающую задачу (исходный код в Приложении А).
2. Скомпилировать написанную программу без использования автоматического распараллеливания с помощью следующей команды:   
   ***/home/user/gcc -O2 -Wall -Werror -o lab1-seq lab1.c***
3. Скомпилировать написанную программу, используя средства автоматического распараллеливания с помощью следующей команды (переменную K следует изменять от 2 до 20 с шагом 10):   
   ***/home/user/gcc -O2 -Wall -Werror -floop-parallelize-all -ftree-parallelize-loops=K lab1.c -o lab1-par-K***
4. Запускать файл lab1-seq из командной строки, увеличивая значения N до значения N1, при котором время выполнения превысит 0.5 с. Продолжать запускать lab1-seq, увеличивая значения N до значения N2, при котором время выполнения превысит 100 с.

N1 = 0, 34 ms

N2 = 480 102088 ms

1. Используя найденные значения N1 и N2, выполнить следующие эксперименты (для автоматизации проведения экспериментов рекомендуется написать скрипт):
2. Запускать **lab1-seq** для значений N = {N1, N1+∆, N1+2∆, N1+3∆,…, N2} и записывать получающиеся значения времени *delta\_ms(N)* в функцию *seq(N), результат в таблице 1.*
3. Запускать **lab1-par-K** для значений N = {N1, N1+∆, N1+2∆, N1+3∆,…, N2} и записывать получающиеся значения времени *delta\_ms(N)* в функцию *par-K(N), результат в таблице 1.*
4. ∆ = (N2- N1)/10 =(480-0)/10≈50.

Таблица 1 – затраченное время на выполнение программы:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
| Delta\_ ms\_seq | 0 | 121 | 822 | 2488 | 5750 | 11643 | 19348 | 30899 | 53782 | 70839 | 124695 |
| Delta\_ms\_par\_1 | 0 | 140 | 885 | 2918 | 6615 | 13016 | 22118 | 35666 | 57343 | 90408 | 132609 |
| Delta\_ ms\_par\_2 | 0 | 152 | 909 | 2770 | 6549 | 14109 | 21845 | 36970 | 58174 | 86687 | 127596 |
| Delta\_ ms\_par\_3 | 0 | 133 | 890 | 2776 | 6515 | 12871 | 21503 | 35636 | 56212 | 77831 | 123779 |
| Delta\_ ms\_par\_4 | 0 | 128 | 874 | 2742 | 6523 | 12874 | 21369 | 34794 | 54964 | 80745 | 120211 |
| Delta\_ ms\_par\_5 | 0 | 131 | 881 | 2739 | 6487 | 12853 | 21343 | 34616 | 54991 | 84413 | 119960 |
| Delta\_ ms\_par\_6 | 0 | 131 | 896 | 2733 | 6495 | 12908 | 21519 | 34573 | 54939 | 78403 | 122046 |
| Delta\_ ms\_par\_7 | 0 | 130 | 884 | 2727 | 6492 | 12813 | 21375 | 34987 | 54924 | 78269 | 130159 |
| Delta\_ ms\_par\_8 | 0 | 130 | 887 | 2770 | 6500 | 12826 | 21315 | 34598 | 54986 | 81388 | 123266 |
| Delta\_ ms\_par\_9 | 0 | 137 | 875 | 2809 | 6490 | 12958 | 21495 | 34746 | 58442 | 80095 | 129116 |
| Delta\_ ms\_par\_10 | 0 | 128 | 875 | 2809 | 6514 | 12863 | 21346 | 34505 | 59551 | 87879 | 132183 |
| Delta\_ ms\_par\_11 | 0 | 132 | 878 | 2761 | 6536 | 12840 | 21351 | 34954 | 56148 | 84464 | 125505 |
| Delta\_ ms\_par\_12 | 0 | 142 | 880 | 2746 | 6507 | 12821 | 21276 | 34559 | 57243 | 77992 | 124398 |
| Delta\_ ms\_par\_13 | 0 | 130 | 879 | 2748 | 6490 | 12842 | 21363 | 34546 | 54849 | 86117 | 129846 |
| Delta\_ ms\_par\_14 | 0 | 131 | 879 | 2743 | 6557 | 12890 | 21355 | 34581 | 57166 | 87138 | 125781 |
| Delta\_ ms\_par\_15 | 0 | 143 | 887 | 2756 | 6508 | 12816 | 21425 | 34980 | 55027 | 83128 | 124581` |
| Delta\_ ms\_par\_16 | 0 | 128 | 878 | 2758 | 6498 | 12888 | 21347 | 34669 | 55281 | 78140 | 125344 |
| Delta\_ ms\_par\_17 | 0 | 130 | 885 | 2850 | 6509 | 12854 | 21419 | 34865 | 54892 | 79181 | 117929 |
| Delta\_ ms\_par\_18 | 0 | 130 | 885 | 2780 | 6466 | 12808 | 21323 | 34543 | 55083 | 78575 | 122148 |
| Delta\_ ms\_par\_19 | 0 | 129 | 875 | 2777 | 6535 | 12881 | 21393 | 34724 | 55290 | 82591 | 118198 |
| Delta\_ ms\_par\_20 | 0 | 129 | 886 | 2764 | 6535 | 12812 | 21363 | 34801 | 56683 | 81931 | 122473 |

Таблица 2 – значение параллельного ускорения:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
| Delta\_S\_par\_1 | 0 | 0,9 | 0,93 | 0,85 | 0,869 | 0,895 | 0,8748 | 0,866 | 0,9379 | 0,7835 | 0,94032 |
| Delta\_S\_par\_2 | 0 | 0,8 | 0,9 | 0,9 | 0,878 | 0,825 | 0,8857 | 0,836 | 0,9245 | 0,8172 | 0,97726 |
| Delta\_S\_par\_3 | 0 | 0,9 | 0,92 | 0,9 | 0,883 | 0,905 | 0,8998 | 0,867 | 0,9568 | 0,9102 | 1,0074 |
| Delta\_S\_par\_4 | 0 | 0,9 | 0,94 | 0,91 | 0,881 | 0,904 | 0,9054 | 0,888 | 0,9785 | 0,8773 | 1,0373 |
| Delta\_S\_par\_5 | 0 | 0,9 | 0,93 | 0,91 | 0,886 | 0,906 | 0,9065 | 0,893 | 0,978 | 0,8392 | 1,03947 |
| Delta\_S\_par\_6 | 0 | 0,9 | 0,92 | 0,91 | 0,885 | 0,902 | 0,8991 | 0,894 | 0,9789 | 0,9035 | 1,0217 |
| Delta\_S\_par\_7 | 0 | 0,9 | 0,93 | 0,91 | 0,886 | 0,909 | 0,9052 | 0,883 | 0,9792 | 0,9051 | 0,95802 |
| Delta\_S\_par\_8 | 0 | 0,9 | 0,93 | 0,9 | 0,885 | 0,908 | 0,9077 | 0,893 | 0,9781 | 0,8704 | 1,01159 |
| Delta\_S\_par\_9 | 0 | 0,9 | 0,94 | 0,89 | 0,886 | 0,899 | 0,9001 | 0,889 | 0,9203 | 0,8844 | 0,96576 |
| Delta\_S\_par\_10 | 0 | 0,9 | 0,94 | 0,89 | 0,883 | 0,905 | 0,9064 | 0,895 | 0,9031 | 0,8061 | 0,94335 |
| Delta\_S\_par\_11 | 0 | 0,9 | 0,94 | 0,9 | 0,88 | 0,907 | 0,9062 | 0,884 | 0,9579 | 0,8387 | 0,99355 |
| Delta\_S\_par\_12 | 0 | 0,9 | 0,93 | 0,91 | 0,884 | 0,908 | 0,9094 | 0,894 | 0,9395 | 0,9083 | 1,00239 |
| Delta\_S\_par\_13 | 0 | 0,9 | 0,94 | 0,91 | 0,886 | 0,907 | 0,9057 | 0,894 | 0,9805 | 0,8226 | 0,96033 |
| Delta\_S\_par\_14 | 0 | 0,9 | 0,94 | 0,91 | 0,877 | 0,903 | 0,906 | 0,894 | 0,9408 | 0,813 | 0,99137 |
| Delta\_S\_par\_15 | 0 | 0,8 | 0,93 | 0,9 | 0,884 | 0,908 | 0,9031 | 0,883 | 0,9774 | 0,8522 | 1,00092 |
| Delta\_S\_par\_16 | 0 | 0,9 | 0,94 | 0,9 | 0,885 | 0,903 | 0,9064 | 0,891 | 0,9729 | 0,9066 | 0,99482 |
| Delta\_S\_par\_17 | 0 | 0,9 | 0,93 | 0,87 | 0,883 | 0,906 | 0,9033 | 0,886 | 0,9798 | 0,8946 | 1,05737 |
| Delta\_S\_par\_18 | 0 | 0,9 | 0,93 | 0,89 | 0,889 | 0,909 | 0,9074 | 0,895 | 0,9764 | 0,9015 | 1,02085 |
| Delta\_S\_par\_19 | 0 | 0,9 | 0,94 | 0,9 | 0,88 | 0,904 | 0,9044 | 0,89 | 0,9727 | 0,8577 | 1,05497 |
| Delta\_S\_par\_20 | 0 | 0,9 | 0,93 | 0,9 | 0,88 | 0,909 | 0,9057 | 0,888 | 0,9488 | 0,8646 | 1,01814 |

**График величины паралельного ускорения**

**Характеристика системы**

**Процессор:**

model name : Intel(R) Core(TM) i5 CPU M 480 @ 2.67GHz

cpu MHz : 1197.000

cache size : 3072 KB

cpu cores : 2

clflush size : 64

cache\_alignment : 64

address sizes : 36 bits physical, 48 bits virtual

**ОЗУ – два модуля с одинаковыми параметрами:**

Memory Device

Array Handle: 0x001D

Error Information Handle: Not Provided

Total Width: 64 bits

Data Width: 64 bits

Size: 2048 MB

Form Factor: DIMM

Set: None

Locator: DIMM\_A

Bank Locator: DIMM\_A

Type: DDR3

Type Detail: Synchronous

Speed: 1333 MHz

Manufacturer: CE80

Serial Number: 9249EC3A

Asset Tag: 1051

Part Number: M471B5773DH0-CH9

Rank: 1

**OS:**

Description: Ubuntu 13.04

Codename: raring

Linux core version 3.8.0-35-generic

**GCC:**

COLLECT\_GCC=gcc

COLLECT\_LTO\_WRAPPER=/usr/lib/gcc/x86\_64-linux-gnu/4.7/lto-wrapper

Целевая архитектура: x86\_64-linux-gnu

Параметры конфигурации: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 4.7.3-1ubuntu1' --with-bugurl=file:///usr/share/doc/gcc-4.7/README.Bugs --enable-languages=c,c++,go,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-4.7 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --with-gxx-include-dir=/usr/include/c++/4.7 --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-gnu-unique-object --enable-plugin --with-system-zlib --enable-objc-gc --with-cloog --enable-cloog-backend=ppl --disable-cloog-version-check --disable-ppl-version-check --enable-multiarch --disable-werror --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --with-tune=generic --enable-checking=release --build=x86\_64-linux-gnu --host=x86\_64-linux-gnu --target=x86\_64-linux-gnu

Модель многопоточности: posix

gcc версия 4.7.3 (Ubuntu/Linaro 4.7.3-1ubuntu1)

Вывод : В результате проведенной лабораторной работы был получен опыт автоматической оптимизации последовательных программ. Выяснено, что GCC позволяет распараллеливать участки кода, не имеющие зависимости по данным , путём построения графа программы и поиска независимых циклов, однако жизнь тлен и на практике получилось получить S>1 только на операциях с матрицами с большой размерностью.

Код Программы:

#include <stdio.h>

#include <stdlib.h>

#include <gsl/gsl\_rng.h>

#include <gsl/gsl\_randist.h>

#include <math.h>

#include <sys/time.h>

#define C\_PARAM = 5

void shellsort(double a[],int n)

{

int j,i,m;

double mid;

for(m = n/2;m>0;m/=2){

for(j = m;j< n;j++){

for(i=j-m;i>=0;i-=m){

if(a[i+m]>=a[i])

break;

else{

mid = a[i];

a[i] = a[i+m];

a[i+m] = mid;

}

}

}

}

}

void printmatrix (double a[],int n)

{

int i;

for(i = 0; i < n; i++){

printf("|%0.1f|", a[i]);

}

}

int main(int argc, char\* argv[])

{

int i,j,k, N,r;

N = atoi(argv[1]);

struct timeval T1, T2;

long delta\_ms;

double vector = 5;

double random\_number;

double M1[N][N];

double M2[N][N];

double M3[N][N];

double V[N];

double sred;

double num = 0;

double Nt=N;

double a,b;

gettimeofday(&T1, NULL);

printf("Started\n");

gsl\_rng \* rgen = gsl\_rng\_alloc(gsl\_rng\_taus);

a=31,15;

b=-1,444445;

for (r=0; r<100; r++)

{

//printf("\n\nMatrix1\n");

for (i = 0; i < N; i++ ){

for (j = 0; j < N; j++ ){

random\_number= gsl\_ran\_gamma(rgen,a,b);

M1[i][j] = random\_number;

//printf("|%f|", M1[i][j]);

}

//printf("\n");

}

//printf("Matrix2\n");

for (i = 0; i < N; i++ ){

for (j = 0; j < N; j++ ){

M2[i][j] = M1[i][j]\*vector;

//printf("|%f|", M2[i][j]);

}

//printf("\n");

}

//printf("Matrix3 resoult\n");

for (i = 0; i < N; i++ ){

for (j = 0; j < N; j++ ){

M3[i][j] =0;

for (k = 0; k < N; k++ ){

M3[i][j] = M3[i][j] + M1[i][k]\*M2[k][j];

}

//printf("|%f|", M3[i][j]);

}

//printf("\n");

}

for (i = 0; i < N; i++ ){

V[i] =0;

sred=0;

for (j = 0; j < N; j++ ){

sred += M3[i][j];

}

sred = sred/Nt;

for (j = 0; j < N; j++ ){

num = M3[i][j];

num=num-sred;

V[i] = num\*num;

}

V[i]= V[i]/Nt;

}

//printf("\nBefore sort:\n");

//printmatrix(V,N);

//printf("\nShell sort:\n");

//shellsort(V,N);

//printmatrix(V,N);

//printf("\n\n");

}

printf("\nStopt\n");

gettimeofday(&T2, NULL);

delta\_ms = 1000\*(T2.tv\_sec - T1.tv\_sec) + (T2.tv\_usec - T1.tv\_usec)/1000;

printf("\nN=%d. Milliseconds passed: %ld\n", N, delta\_ms);

return 0;

}