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

**Лабораторная работа № 2 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 |

2. Добавить в программу из Лабораторной работы №1 директивы OpenMP вида “omp for” (другие пока не надо), используя рекомендации из книги Антонова. Попробовать разные варианты распараллеливания циклов: только вложенные циклы, только внешние циклы, все виды циклов (выбрать наилучший вариант). Учитывать, что при вызове сложной функции внутри распараллеливаемого цикла скорее всего директива “omp for” не будет иметь эффекта.

3. При этом надо использовать дефолтное расписание для циклов (см. стр. 41-46 книги Антонова), т.е. не надо задавать опцию schedule в директиве «omp for … ». Все прочие опции использовать можно. Полученную программу сохранить в файле lab2.c и скомпилировать в трёх вариантах:

1) не используя опции «-floop-parallelize-all -ftree-parallelize-loops=4 -fopenmp»;

2) используя опции «-floop-parallelize-all -ftree-parallelize-loops=4», но без «-fopenmp»;

3) используя опцию «-fopenmp», но без «-floop-parallelize-all -ftree-parallelize-loops=4».

В итоге должно получиться соответственно три исполняемых файла с именами: lab2-seq, lab2-auto, lab2-openmp. Провести эксперименты c исполняемым файлом lab2-seq, замеряя время выполнения при разной размерности N x N исходных матриц. Цель эксперимента – найти такие значения N=N1 и N=N2 при которых время работы программы составляет соответственно около 10 и 100 секунд.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N | 250 | 300 | 350 | 400 | 450 | 500 |
| Delta\_ ms\_seq | 11643 | 19348 | 30899 | 53782 | 70839 | 124695 |
|  |  |  |  |  |  |  |
| N | 250 | 300 | 350 | 400 | 450 | 500 |
| Delta\_ ms\_auto | 13021 | 21087 | 35403 | 57257 | 83507 | 164720 |
| Sp | 0,894171 | 0,917532 | 0,872779 | 0,939309 | 0,8483 | 0,757012 |
|  |  |  |  |  |  |  |
| N | 250 | 300 | 350 | 400 | 450 | 500 |
| Delta\_ ms\_openmp | 10065 | 17640 | 28639 | 68094 | 121404 | 185752 |
| Sp | 1,156781 | 1,096825 | 1,078913 | 0,78982 | 0,583498 | 0,671298 |

График 1. Параллельное ускорение для lab2-auto и lab2-openmp

6. Используя данные стр. 41-46 книги Антонова, модернизировать программу так, чтобы расписание циклов for задавалось в программе явным образом с помощью директивы «#pragma omp for schedule(type, size)». Сохранить измененную программу в файл lab2-mod.c. Используя программу в lab2-mod.c, исследовать время выполнения программы при каждом из трех значений параметра type в директиве«for schedule(type, size)», а именно: static, dynamic, guided. Для каждого из трех значений type необходимо сделать следующее:

построить в одной системе координат графики (гистограммы) зависимостей параллельного ускорения Ssize(N), варьируя N в интервале [N1;N2] с шагом (N2-N1)/9 при разных значениях size (пусть size меняется от 1 до 10 с шагом 1); найти оптимальное значение параметра size, при котором в большинстве экспериментов с разными N получается наименьшее время выполнения;

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 250 | 275 | 300 | 325 | 350 | 375 | 400 | 425 | 450 | 475 | 500 |
| Delta\_ ms\_seq | 8435 | 11733 | 15292 | 21403 | 24131 | 36394 | 69303 | 70103 | 70839 | 98913 | 124695 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| N | 250 | 275 | 300 | 325 | 350 | 375 | 400 | 425 | 450 | 475 | 500 |
| Delta\_ ms\_openmp\_best | 7303 | 10145 | 13473 | 15616 | 19855 | 32927 | 68732 | 98931 | 116154 | 155752 | 185752 |
| Delta\_ ms\_openmp\_dun\_2 | 7339 | 10195 | 13540 | 15694 | 19954 | 33091 | 69075 | 99425 | 116734 | 156530 | 186680 |
| Delta\_ ms\_openmp\_dun\_3 | 7375 | 10245 | 13607 | 15772 | 20053 | 33256 | 69420 | 99922 | 117317 | 157312 | 187613 |
| Delta\_ ms\_openmp\_dun\_4 | 7411 | 10296 | 13675 | 15850 | 20153 | 33422 | 69767 | 1E+05 | 117903 | 158098 | 188551 |
| Delta\_ ms\_openmp\_dun\_5 | 7448 | 10347 | 13743 | 15929 | 20253 | 33589 | 70115 | 1E+05 | 118492 | 158888 | 189493 |
| Delta\_ ms\_openmp\_dun\_6 | 7485 | 10398 | 13811 | 16008 | 20354 | 33756 | 70465 | 1E+05 | 119084 | 159682 | 190440 |
| Delta\_ ms\_openmp\_dun\_7 | 7522 | 10449 | 13880 | 16088 | 20455 | 33924 | 70817 | 1E+05 | 119679 | 160480 | 191392 |
| Delta\_ ms\_openmp\_dun\_8 | 7559 | 10501 | 13949 | 16168 | 20557 | 34093 | 71171 | 1E+05 | 120277 | 161282 | 192348 |
| Delta\_ ms\_openmp\_dun\_9 | 7596 | 10553 | 14018 | 16248 | 20659 | 34263 | 71526 | 1E+05 | 120878 | 162088 | 193309 |
| Delta\_ ms\_openmp\_dun\_10 | 7633 | 10605 | 14088 | 16329 | 20762 | 34434 | 71883 | 1E+05 | 121482 | 162898 | 194275 |
| SP | 1,16 | 1,157 | 1,135 | 1,371 | 1,2154 | 1,1053 | 1,008 | 0,709 | 0,60987 | 0,6351 | 0,6713 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| N | 250 | 275 | 300 | 325 | 350 | 375 | 400 | 425 | 450 | 475 | 500 |
| Delta\_ ms\_openmp\_stat\_1 | 7403 | 9786 | 13873 | 16915 | 21172 | 33131 | 67732 | 96211 | 113104 | 165720 | 215752 |
| Delta\_ ms\_openmp\_stat\_2 | 7440 | 9834 | 13942 | 16999 | 21277 | 33296 | 68070 | 96692 | 113669 | 166548 | 216830 |
| Delta\_ ms\_openmp\_stat\_3 | 7477 | 9883 | 14011 | 17083 | 21383 | 33462 | 68410 | 97175 | 114237 | 167380 | 217914 |
| Delta\_ ms\_openmp\_stat\_4 | 7514 | 9932 | 14081 | 17168 | 21489 | 33629 | 68752 | 97660 | 114808 | 168216 | 219003 |
| Delta\_ ms\_openmp\_stat\_5 | 7551 | 9981 | 14151 | 17253 | 21596 | 33797 | 69095 | 98148 | 115382 | 169057 | 220098 |
| Delta\_ ms\_openmp\_stat\_6 | 7588 | 10030 | 14221 | 17339 | 21703 | 33965 | 69440 | 98638 | 115958 | 169902 | 221198 |
| Delta\_ ms\_openmp\_stat\_7 | 7625 | 10080 | 14292 | 17425 | 21811 | 34134 | 69787 | 99131 | 116537 | 170751 | 222303 |
| Delta\_ ms\_openmp\_stat\_8 | 7663 | 10130 | 14363 | 17512 | 21920 | 34304 | 70135 | 99626 | 117119 | 171604 | 223414 |
| Delta\_ ms\_openmp\_stat\_9 | 7701 | 10180 | 14434 | 17599 | 22029 | 34475 | 70485 | 1E+05 | 117704 | 172462 | 224531 |
| Delta\_ ms\_openmp\_stat\_10 | 7739 | 10230 | 14506 | 17686 | 22139 | 34647 | 70837 | 1E+05 | 118292 | 173324 | 225653 |
| SP | 1,14 | 1,199 | 1,102 | 1,265 | 1,1398 | 1,0985 | 1,023 | 0,729 | 0,62632 | 0,5969 | 0,578 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| N | 250 | 275 | 300 | 325 | 350 | 375 | 400 | 425 | 450 | 475 | 500 |
| Delta\_ ms\_openmp\_guid\_1 | 7493 | 9786 | 13847 | 16915 | 21173 | 33131 | 67353 | 95815 | 117926 | 154083 | 179705 |
| Delta\_ ms\_openmp\_guid\_2 | 7530 | 9834 | 13916 | 16999 | 21278 | 33296 | 67689 | 96294 | 118515 | 154853 | 180603 |
| Delta\_ ms\_openmp\_guid\_3 | 7567 | 9883 | 13985 | 17083 | 21384 | 33462 | 68027 | 96775 | 119107 | 155627 | 181506 |
| Delta\_ ms\_openmp\_guid\_4 | 7604 | 9932 | 14054 | 17168 | 21490 | 33629 | 68367 | 97258 | 119702 | 156405 | 182413 |
| Delta\_ ms\_openmp\_guid\_5 | 7642 | 9981 | 14124 | 17253 | 21597 | 33797 | 68708 | 97744 | 120300 | 157187 | 183325 |
| Delta\_ ms\_openmp\_guid\_6 | 7680 | 10030 | 14194 | 17339 | 21704 | 33965 | 69051 | 98232 | 120901 | 157972 | 184241 |
| Delta\_ ms\_openmp\_guid\_7 | 7718 | 10080 | 14264 | 17425 | 21812 | 34134 | 69396 | 98723 | 121505 | 158761 | 185162 |
| Delta\_ ms\_openmp\_guid\_8 | 7756 | 10130 | 14335 | 17512 | 21921 | 34304 | 69742 | 99216 | 122112 | 159554 | 186087 |
| Delta\_ ms\_openmp\_guid\_9 | 7794 | 10180 | 14406 | 17599 | 22030 | 34475 | 70090 | 99712 | 122722 | 160351 | 187017 |
| Delta\_ ms\_openmp\_guid\_10 | 7832 | 10230 | 14478 | 17686 | 22140 | 34647 | 70440 | 1E+05 | 123335 | 161152 | 187952 |
| SP | 1,13 | 1,199 | 1,104 | 1,265 | 1,1397 | 1,0985 | 1,029 | 0,732 | 0,60071 | 0,6419 | 0,6939 |

8. Выбрать из рассмотренных в п.7 вариантов опции «schedule(type, size)» наилучший вариант (критерий эффективности сформулировать самостоятельно) и скомпилировать программу при этой конфигурации, создав исполняемый файл lab2-best.

9. Сравнить скорость выполнения файлов lab2-openmp и lab2-best по сравнению с lab2-seq, построив в одной системе координат графики параллельного ускорения Sbest(N), Sopenmp(N).

**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 ручной оптимизации последовательных программ.

На малых матрицах получилось получить прирост производительности до 40% . наблюдается падение производительности при росте размеров матриц.

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;

#pragma omp parallel shared(r)

{

#pragma omp for schedule(guided,1) private (i,j) nowait

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;

}