Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра Програмної інженерії

Контрольна робота

з дисципліни «Паралельне програмування»

Варіант 26

Студент

гр. ПІз-11-1 Нікулін А.В.

Викладач

К.т.н., доцент Мельнікова Р. В.

Харків, 2014 р.

Завдання на контрольну роботу

1. За допомогою мови програмування С розробити алгоритм, що вирішує поставлене завдання, при послідовному виконанні операцій. Перевірити правильність роботи алгоритму.
2. Визначити теоретичну обчислювальну складність алгоритму.
3. Визначити час виконання алгоритму для масиву розміром 2048 та 4096 елементів. Для виміру часу виконання обрати одну з функцій: \_\_rdtsc() або GetTickCount.
4. Розробити алгоритм, що вирішує поставлене завдання, з використанням SIMD операцій. Перевірити правильність роботи алгоритму.
5. Визначити час виконання алгоритму з використанням SIMD операцій.
6. Побудувати графіки залежності часу обчислень від розміру масиву для пунктів 2, 3 та 5.
7. Зробити висновки по ефективності використання SIMD операцій.
8. Дано 2 масиву чисел типу double однакової розмірності. Розробити алгоритм для обчислення z[i]= x[i]-y[i].

#include "stdafx.h"

#include <iostream>

#include <emmintrin.h>

#include <ctime>

#include <Windows.h>

#include <intrin.h>

/\* 26. Дано 2 масиву чисел типу double однакової розмірності. Розробити алгоритм для обчислення z[i]= x[i]-y[i]. \*/

#define N 32768

int \_tmain(int argc, \_TCHAR\* argv[])

{

srand(std::time(0));

\_\_m128d a;

double x[N], y[N], z[N];

int i;

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

x[i]=rand();

y[i]=rand();

}

\_\_int64 start\_s=\_\_rdtsc();

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

z[i] = x[i] - y[i];

}

\_\_int64 stop\_s=\_\_rdtsc();

std::cout<<(stop\_s -start\_s)<<std::endl;

\_\_m128d\* X = (\_\_m128d\*)x;

\_\_m128d\* Y = (\_\_m128d\*)y;

\_\_m128d\* Z = (\_\_m128d\*)z;

\_\_int64 start\_s2=\_\_rdtsc();

for (i=0; i<N/2; ++i){

Z[i] = \_mm\_sub\_pd(X[i], Y[i]);

}

\_\_int64 stop\_s2=\_\_rdtsc();

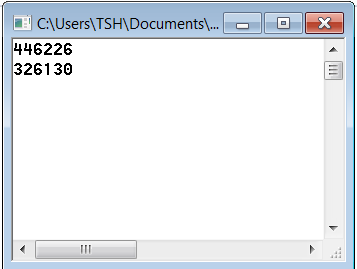
std::cout<<stop\_s2 -start\_s2;

getchar();

return 0;

}

Результат работы программы для массива 32768 элементов



Теоритическая сложность алгоритма O(n) = n.

Время выполнения для 2048, 4096 и 32768 без использования SIMD.

|  |  |
| --- | --- |
| 2048 | 28110 |
| 4096 | 38112 |
| 32768 | 446226 |

Время выполнения для 2048, 4096 и 32768 используя SIMD

|  |  |
| --- | --- |
| 2048 | 20340 |
| 4096 | 34120 |
| 32768 | 326130 |

График зависимости времени вычисления:

Вывод: Как видно из таблиц и графика скорости выполнения, спользование SIMD команд ускоряет выполнения кода примерно в 1.37. Наиболее заметен прирост на достаточно больших размерах исходного массива.