#include <stdio.h>

#include <time.h>

#include <math.h>

#define N 10000

double norma(double arr[]) {

double result = 0;

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

result += arr[i] \* arr[i];

result = sqrt(result);

return result;

}

double A[N \* N];

int main() {

int start = time(NULL);

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

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

A[i \* N + j] = 1;

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

A[i \* N] = 2;

double b[N];

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

b[i] = N + 1;

double x[N] = { 0 };

double estimate;

double normab = norma(b);

while (1) {

double u[N]={0};

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

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

u[i] += A[i \* N + j] \* x[j];

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

u[i] -= b[i];

estimate = norma(u) / normab;

if (estimate < 0.00001)

break;

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

u[i] \*= 0.00001;

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

x[i] -= u[i];

}

int end = time(NULL);

printf("%d\n", end - start);

return 0;

}

Пк 32

Серв 46

#include <stdio.h>

#include <time.h>

#include <math.h>

#include <omp.h>

#define N 10000

double norma(double arr[]) {

double result = 0;

int i;

#pragma omp parallel for

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

result += arr[i] \* arr[i];

result = sqrt(result);

return result;

}

double A[N \* N];

int main() {

int start = time(NULL);

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

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

A[i \* N + j] = 1;

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

A[i \* N] = 2;

double b[N];

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

b[i] = N + 1;

double x[N] = { 0 };

double estimate;

double normab = norma(b);

int i;

int j;

while(1) {

double u[N] = { 0 };

#pragma omp parallel for

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

#pragma omp parallel for // этот параллел игнорится(видно, ведь там нужен редьюс, но все работает норм)

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

u[i] += A[i \* N + j] \* x[j];

#pragma omp parallel for

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

u[i] -= b[i];

estimate = norma(u) / normab;

if (estimate < 0.00001) {

break;

}

#pragma omp parallel for

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

u[i] \*= 0.00001;

#pragma omp parallel for

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

x[i] -= u[i];

}

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

printf("%llf ", x[i]);

int end = time(NULL);

printf("%d\n", end - start);

return 0;

}

Пк 6

Серв 6

#include <stdio.h>

#include <time.h>

#include <math.h>

#include <omp.h>

#define N 10000

double norma(double arr[]) {

double result = 0;

int i;

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

result += arr[i] \* arr[i];

result = sqrt(result);

return result;

}

double A[N \* N];

int main() {

int start = time(NULL);

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

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

A[i \* N + j] = 1;

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

A[i \* N] = 2;

double b[N];

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

b[i] = N + 1;

double x[N] = { 0 };

double estimate = 1000000;

double normab = norma(b);

normab \*= normab;

int i;

int j;

while (estimate > 0.0000000001) {

double u[N] = { 0 };

double normau = 0;

#pragma omp parallel

{

#pragma omp for collapse(2) private(i,j)

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

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

u[i] += A[i \* N + j] \* x[j];

#pragma omp for

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

u[i] -= b[i];

#pragma omp for reduction(+:normau)

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

normau += u[i] \* u[i];

#pragma omp single

{

estimate = normau / normab;

}

#pragma omp for

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

u[i] \*= 0.00001;

#pragma omp for

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

x[i] -= u[i];

}

}

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

printf("%lf ", x[i]);

int end = time(NULL);

printf("%d\n", end - start);

return 0;

}

Пк 6

Серв 6