

Теория параллелизма CS

Отчёт по заданию 2

“Уравнение теплопроводности”

Аксенов Валентин, группа 21933

https://github.com/AxeVal/teor_parall-cs.git

Задание:

Реализовать решение уравнения теплопроводности (пятиточечный шаблон) в двумерной области на равномерных сетках (128^2 , 256^2 , 512^2 , 1024^2). Граничные условия - линейная интерполяция между углами области. Значения в углах - 10, 20, 30, 20. Ограничить точность - 10^{-6} и максимальное число итераций - 10^6 .

GPU: pgc++ task2.cpp -o task2_gpu -fast -acc=gpu -O2

CPU(onecore): pgc++ task2.cpp -o task2_one -fast -O2

CPU(multicore): pgc++ task2.cpp -o task2_multi -fast -acc=multicore -O2
-Mconcur=allcores

128 x 128	time	error	iterations
Gpu	1.605 sec	9.99928e-07	29134
Cpu-onecore	2.320 sec	9.99928e-07	29134
Cpu-multicore	2.510 sec	9.99928e-07	29134

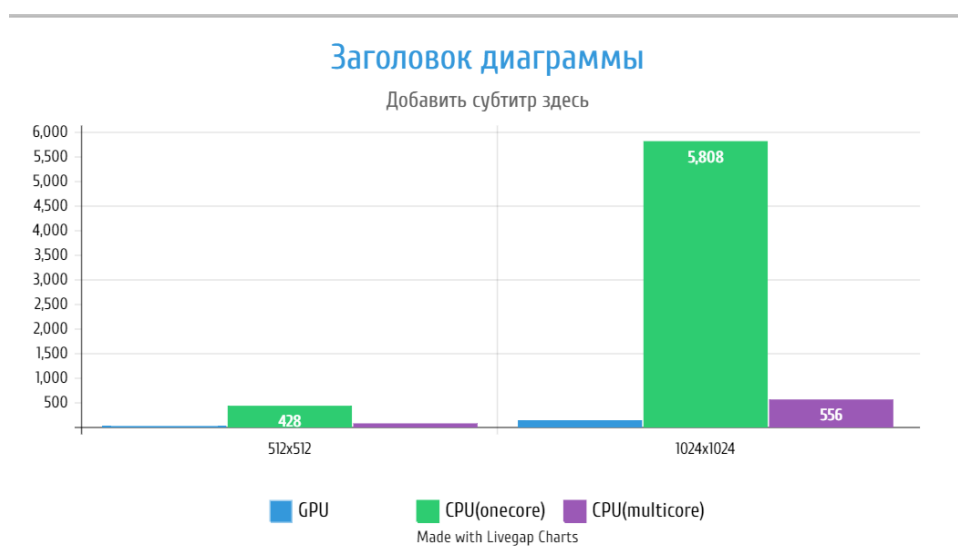
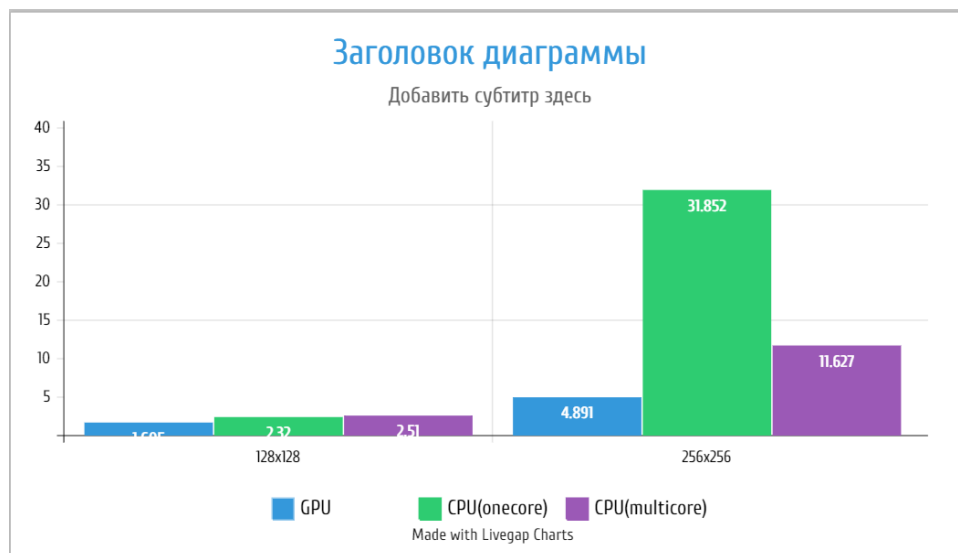
256 x 256	time	error	iterations
Gpu	4.891 sec	9.99958e-07	99094
Cpu-onecore	31.852 sec	9.99958e-07	99094
Cpu-multicore	11.627 sec	9.99958e-07	99094

512 x 512	time	error	iterations
Gpu	16.245 sec	9.99996e-07	324376
Cpu-onecore	7 min 8 sec	9.99996e-07	324376
Cpu-multicore	1 min 10 sec	9.99996e-07	324376

1024 x 1024	time (sec)	error	iterations
Gpu	2min 14 sec	1.02696e-06	1000000
Cpu-onecore	96 min 48 sec	1.02696e-06	1000000

Cpu-multicore	9 min 16 sec	1.02696e-06	1000000
---------------	--------------	-------------	---------

Гистограммы затрат времени (в секундах)



Оптимизация:

Тесты проводились на гри с размером сетки 128x128

1. Распараллеливание цикла копирования

Ускорение в ~50 раз

2. Распараллеливание цикла удаления

Ускорение на ~10%

Код:

```
1  int main(int argc, char *argv[])
2  {
3      for(int arg = 0; arg < argc; arg++)
4      {
5          std::stringstream stream;
6          if(strcmp(argv[arg], "-error") == 0)
7          {
8              stream << argv[arg+1];
9              stream >> eps;
10         }
11         else if(strcmp(argv[arg], "-iter") == 0)
12         {
13             stream << argv[arg + 1];
14             stream >> iter_max;
15         }
16         else if(strcmp(argv[arg], "-size") == 0)
17         {
18             stream << argv[arg+1];
19             stream >> size;
20         }
21     }
22
23     int S = size - 1;
24
25     std::cout << "Settings: " << std::endl;
26     std::cout << "\tEPS: " << eps << std::endl;
27     std::cout << "\ttotal iteration: " << iter_max << std::endl;
28     std::cout << "\tsize: " << size << std::endl;
29
30     double** A = new double*[size];
31     for(int i = 0; i < size; i++)
32         A[i] = new double[size];
33
34     double** Anew = new double*[size];
35     for(int i = 0; i < size; i++)
36         Anew[i] = new double[size];
37
38     initArr(A);
39
40     double error = 1.0;
41     int iteration = 0;
42
43     #pragma acc data copy(A[:size][:size]) create(Anew[:size][:size])
44     while ( (error > eps) && (iteration < iter_max) )
45     {
46         error = 0.0;
47
48         #pragma acc parallel loop reduction(max:error)
49         for (int i = 1; i < S; i++)
50         {
51             #pragma acc loop reduction(max:error)
52             for (int j = 1; j < S; j++)
53             {
54                 Anew[i][j] = 0.25 * ( A[i][i-1] + A[i][i+1] + A[i-1][j] + A[i+1][j] + A[i-1][j-1] );
55                 error = fmax( error, fabs(Anew[i][j] - A[i][j]) );
56             }
57         }
58
59         #pragma acc parallel loop collapse (2)
60         for (int i = 1; i < S; i++)
61         {
62             for (int j = 1; j < S; j++)
63             {
64                 A[i][j] = Anew[i][j];
65             }
66         }
67
68         iteration++;
69     }
70
71     std::cout << "Result: " << std::endl;
72     std::cout << "\tIterations: " << iteration << std::endl;
73     std::cout << "\tError: " << error << std::endl;
74
75     #pragma acc loop
76     for (size_t i = 0; i < size; i++)
77     {
78         delete[] A[i];
79         delete[] Anew[i];
80     }
81     delete[] A;
82     delete[] Anew;
83 }
```