

Задание 10

Отчёт

по Модифицировать уравнение Лапласа для 3D-области

Ши Хуэй shihuicollapsor@gmail.com

1. Постановка задачи

Построить сетку (на каждом процессе выделить массив размера, равного числу элементов в подобласти). Будем использовать блочные 3D-подобласти (каждый процесс работает над своей частью сетки $\{N_x, N_y, N_z\}$, причём $1 \leq N_x \leq N$, $1 \leq N_y \leq N$, $1 \leq N_z \leq N$, где N - размер всей сетки).

Рекомендуется использовать виртуальные топологии для организации обменов

Инициализировать начальное значение f случайным значением в каждой области сетки

До предустановленного числа итераций n_iter выполнять вычисления согласно методу Якоби

Запрещается хранить массив, соответствующий полной сетке, на одном процессе (за исключением запуска на 1 процессе)

Для коммуникации разрешается использовать любые виды обменов, главное требование - пользоваться производными типами данных для отправки сообщений. Для приёма (recv) сообщений разрешается воспринимать сообщение как набор из Q подряд идущих элементов

Можно предполагать, что размер сетки N - степень двойки. Сетка кубическая ($N*N*N$)

Произвести запуски на Polus (через mpisubmit.pl !)

Для фиксированного большого размера сетки произвести запуски при числе процессов $P = \{1, 2, 4, 6, 8, 12, 16\}$, нарисовать $T(P)$. На 8, 12, 16 процессах программа должна работать для случаев, когда число процессов в каждом измерении > 1 ($8 = 2*2*2$, $12 = 2*3*2$, $16 = 2*4*2$)

2. Формат командной строки

```
mpicc -g -Wl,-rpath=/home_edu/edu-cmc-sqi22/edu-cmc-sqi22-29/sem07/mpiP-3.5 -L /home_edu/edu-cmc-sqi22/edu-cmc-sqi22-29/sem07/mpiP-3.5 -lmpiP work10.c -o a
mpisubmit.pl -p P ./a
```

3. Спецификация системы

- Operating system : polus

4. Записи экспериментов и результаты

Мы проверили корректность работы программы. В одноступенчатом выводе ошибки мы видим, что от итерации 0 до итерации 99 норма разницы после каждой итерации постоянно уменьшается. Это показывает, что по мере продвижения итерации Якобиана разрыв между двумя последовательными решениями с временным шагом становится все меньше и меньше. Норма медленно снижается примерно с $1,2 \cdot 10^{-2}$ до $7,4 \cdot 10^{-3}$, что отражает процесс постепенного приближения к стабильному раствору.

```
collapsor@CollapsordeMacBook-Pro sem10 % mpirun -np 16 --map-by :OVERSUBSCRIBE
```

```
./a
```

```
Iteration 0 difference norm = 1.204185e-02
Iteration 1 difference norm = 1.055645e-02
Iteration 2 difference norm = 1.027465e-02
Iteration 3 difference norm = 1.012869e-02
Iteration 4 difference norm = 1.002356e-02
Iteration 5 difference norm = 9.937469e-03
Iteration 6 difference norm = 9.862584e-03
Iteration 7 difference norm = 9.795245e-03
Iteration 8 difference norm = 9.733431e-03
Iteration 9 difference norm = 9.675896e-03
Iteration 10 difference norm = 9.621811e-03
Iteration 11 difference norm = 9.570588e-03
Iteration 12 difference norm = 9.521795e-03
Iteration 13 difference norm = 9.475099e-03
Iteration 14 difference norm = 9.430241e-03
Iteration 15 difference norm = 9.387011e-03
Iteration 16 difference norm = 9.345238e-03
Iteration 17 difference norm = 9.304779e-03
Iteration 18 difference norm = 9.265514e-03
Iteration 19 difference norm = 9.227340e-03
Iteration 20 difference norm = 9.190168e-03
Iteration 21 difference norm = 9.153921e-03
Iteration 22 difference norm = 9.118533e-03
Iteration 23 difference norm = 9.083944e-03
Iteration 24 difference norm = 9.050099e-03
Iteration 25 difference norm = 9.016954e-03
Iteration 26 difference norm = 8.984464e-03
Iteration 27 difference norm = 8.952592e-03
```

Iteration 28 difference norm = 8.921304e-03
Iteration 29 difference norm = 8.890566e-03
Iteration 30 difference norm = 8.860352e-03
Iteration 31 difference norm = 8.830635e-03
Iteration 32 difference norm = 8.801390e-03
Iteration 33 difference norm = 8.772595e-03
Iteration 34 difference norm = 8.744231e-03
Iteration 35 difference norm = 8.716277e-03
Iteration 36 difference norm = 8.688716e-03
Iteration 37 difference norm = 8.661532e-03
Iteration 38 difference norm = 8.634709e-03
Iteration 39 difference norm = 8.608234e-03
Iteration 40 difference norm = 8.582093e-03
Iteration 41 difference norm = 8.556273e-03
Iteration 42 difference norm = 8.530763e-03
Iteration 43 difference norm = 8.505552e-03
Iteration 44 difference norm = 8.480629e-03
Iteration 45 difference norm = 8.455986e-03
Iteration 46 difference norm = 8.431611e-03
Iteration 47 difference norm = 8.407498e-03
Iteration 48 difference norm = 8.383637e-03
Iteration 49 difference norm = 8.360021e-03
Iteration 50 difference norm = 8.336642e-03
Iteration 51 difference norm = 8.313494e-03
Iteration 52 difference norm = 8.290569e-03
Iteration 53 difference norm = 8.267862e-03
Iteration 54 difference norm = 8.245366e-03
Iteration 55 difference norm = 8.223075e-03
Iteration 56 difference norm = 8.200985e-03
Iteration 57 difference norm = 8.179089e-03
Iteration 58 difference norm = 8.157383e-03
Iteration 59 difference norm = 8.135863e-03
Iteration 60 difference norm = 8.114522e-03
Iteration 61 difference norm = 8.093358e-03
Iteration 62 difference norm = 8.072365e-03
Iteration 63 difference norm = 8.051540e-03

Iteration 64 difference norm = 8.030879e-03
Iteration 65 difference norm = 8.010379e-03
Iteration 66 difference norm = 7.990034e-03
Iteration 67 difference norm = 7.969843e-03
Iteration 68 difference norm = 7.949802e-03
Iteration 69 difference norm = 7.929907e-03
Iteration 70 difference norm = 7.910156e-03
Iteration 71 difference norm = 7.890545e-03
Iteration 72 difference norm = 7.871072e-03
Iteration 73 difference norm = 7.851733e-03
Iteration 74 difference norm = 7.832528e-03
Iteration 75 difference norm = 7.813451e-03
Iteration 76 difference norm = 7.794502e-03
Iteration 77 difference norm = 7.775678e-03
Iteration 78 difference norm = 7.756976e-03
Iteration 79 difference norm = 7.738394e-03
Iteration 80 difference norm = 7.719930e-03
Iteration 81 difference norm = 7.701582e-03
Iteration 82 difference norm = 7.683348e-03
Iteration 83 difference norm = 7.665225e-03
Iteration 84 difference norm = 7.647213e-03
Iteration 85 difference norm = 7.629308e-03
Iteration 86 difference norm = 7.611510e-03
Iteration 87 difference norm = 7.593816e-03
Iteration 88 difference norm = 7.576225e-03
Iteration 89 difference norm = 7.558735e-03
Iteration 90 difference norm = 7.541344e-03
Iteration 91 difference norm = 7.524051e-03
Iteration 92 difference norm = 7.506855e-03
Iteration 93 difference norm = 7.489753e-03
Iteration 94 difference norm = 7.472744e-03
Iteration 95 difference norm = 7.455828e-03
Iteration 96 difference norm = 7.439002e-03
Iteration 97 difference norm = 7.422266e-03
Iteration 98 difference norm = 7.405617e-03
Iteration 99 difference norm = 7.389055e-03

В последующей программе мы используем норму только на первой и последней итерациях и выводим такие параметры, как время выполнения.

P=1

Iterations = 100

N = 240

Sub-three-dimensional grid: 240 x 240 x 240

Initial norm = 6.211261e-03

Final norm = 4.981183e-03

Total execution time (T): 52.735847 seconds

P=2

Iterations = 100

N = 240

Sub-three-dimensional grid: 120 x 240 x 240

Initial norm = 6.211891e-03

Final norm = 4.981242e-03

Total execution time (T): 36.685190 seconds

P=4

Iterations = 100

N = 240

Sub-three-dimensional grid: 120 x 120 x 240

Initial norm = 6.211704e-03

Final norm = 4.979460e-03

Total execution time (T): 24.966244 seconds

P=6

Iterations = 100

N = 240

Sub-three-dimensional grid: 80 x 120 x 240

Initial norm = 6.211779e-03

Final norm = 4.978109e-03

Total execution time (T): 17.483712 seconds

P=8

Iterations = 100

N = 240

Sub-three-dimensional grid: 120 x 120 x 120

Initinal norm = 6.210973e-03

Final norm = 4.977695e-03

Total execution time (T): 13.720810 seconds

P=12

Iterations = 100

N = 240

Sub-three-dimensional grid: 80 x 120 x 120

Initinal norm = 6.211883e-03

Final norm = 4.975330e-03

Total execution time (T): 8.324230 seconds

P=16

Iterations = 100

N = 240

Sub-three-dimensional grid: 60 x 120 x 120

Initinal norm = 6.211704e-03

Final norm = 4.972345e-03

Total execution time (T): 6.896592 seconds

Iterations = 100

N = 360

Sub-three-dimensional grid: 360 x 360 x 360

Initinal norm = 5.071432e-03

Final norm = 4.174393e-03

Total execution time (T): 183.556714 seconds

Iterations = 100

N = 360

Sub-three-dimensional grid: 180 x 360 x 360

Initinal norm = 5.071610e-03

Final norm = 4.174457e-03

Total execution time (T): 153.094413 seconds

Iterations = 100

N = 360

Sub-three-dimensional grid: 180 x 180 x 360

Initinal norm = 5.071683e-03

Final norm = 4.174025e-03

Total execution time (T): 84.132910 seconds

Iterations = 100

N = 360

Sub-three-dimensional grid: 120 x 180 x 360

Initinal norm = 5.071768e-03

Final norm = 4.173468e-03

Total execution time (T): 66.413442 seconds

Iterations = 100

N = 360

Sub-three-dimensional grid: 180 x 180 x 180

Initinal norm = 5.071690e-03

Final norm = 4.173523e-03

Total execution time (T): 29.042248 seconds

Iterations = 100

N = 360

Sub-three-dimensional grid: 120 x 180 x 180

Initinal norm = 5.071571e-03

Final norm = 4.172465e-03

Total execution time (T): 20.062424 seconds

Iterations = 100

N = 360

Sub-three-dimensional grid: 90 x 180 x 180

Initinal norm = 5.071626e-03

Final norm = 4.171404e-03

Total execution time (T): 24.674909 seconds

6. Заключение

Нарисуйте $T(P)$ в соответствии со временем, и вы увидите, что время выполнения уменьшается в зависимости от увеличения количества потоков.

