Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Отчет по заданию

«Численное решение трехмерного гиперболического уравнения в частных производных на ПВС IBM Blue Gene/P и IBM Polus»

Выполнил студент 624 группы
Пучкин Данила Андреевич
Вариант 5

Москва
2018

Математическая постановка задачи

Целью задания является исследование производительности алгоритма получения численного решения гиперболического уравнения с использованием технологий MPI и MPI/OpenMP на ПВС IBM Blue Gene/P и IBM Polus. В варианте 5 подразумевалось использование граничных условий 1-го рода для переменных y и z и использование периодических условий для переменной x.

Рассмотрим математическую постановку задачи:

В трехмерной замкнутой области

$$\Omega = [0 \leq x \leq L_x] \times [0 \leq y \leq L_y] \times [0 \leq z \leq L_z],$$

для $(0 \leq t \leq T]$ требуется найти решение $u(x, y, z, t)$ уравнения в частных производных

$$\frac{\delta^2 u}{\delta t^2} = \Delta u \tag{1}$$

с начальными условиями

$$u|_{t=0} = \phi(x, y, z) \tag{2}$$

$$\frac{\delta u}{\delta t} \; |_{t=0} = 0, \tag{3}$$

при условии, что на границах области заданы граничные условия

$$u(0, y, z, t) = u(L_x, y, z, t) \qquad u_x(0, y, z, t) = u_x(L_x, y, z, t) \tag{4}$$

$$u(x, 0, z, t) = 0) \qquad u(x, L_y, z, t) = 0 \tag{5}$$

$$u(x, y, 0, t) = 0 \qquad u(x, y, L_z, t) = 0 \tag{6}$$

Численный метод решения

Для численного решения задачи введем на Ω сетку $\omega_{h\tau} = \overline{\omega_h} \times \omega_\tau$, где

$$T = T_0, \quad L_x = L_{x_0}, \quad L_y = L_{y_0}, \quad L_z = L_{z_0}$$

$$\overline{\omega_h} = \{(x_i = ih_x, y_j = jh_y, z_k = kh_k), i, j, k = 0, 1, ..., N, h_x N = L_x, h_y N = L_y, h_z N = L_z\},$$

$$\omega_\tau = \{t_n = n\tau, n = 0, 1, ..., K, \tau K = T\}.$$

Для аппроксимации уравнения воспользуемся следующей системой уравнений:

$$\frac{u_{ijk}^{n+1} - 2u_{ijk}^{n} + u_{ijk}^{n-1}}{\tau^2} = \Delta_h u^n, \qquad (x_i, y_j, z_k) \in \omega_h, \quad n = 1, 2, ..., K-1,$$

где $\Delta_h u^n$ - семиточечный разностный аналог оператора Лапласа:

$$\Delta_h u^n = \frac{u_{i-1,j,k}^{n} - 2u_{i,j,k}^{n} + u_{i+1,j,k}^{n}}{h^2} + \frac{u_{i,j-1,k}^{n} - 2u_{i,j,k}^{n} + u_{i,j+1,k}^{n}}{h^2} +$$

$$+ \frac{u_{i,j,k+1}^{n} - 2u_{i,j,k}^{n} + u_{i,j,k+1}^{n}}{h^2}.$$

Для начала счета должны быть заданы:

$$u_{ijk}^{0} = \phi(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in \omega_h$$

$$u_{ijk}^{1} = u_{ijk}^{0} + \frac{\tau^2}{2}\Delta_h \phi(x_i, y_j, z_k)$$

Функция начального условия была выбрана следущей:

$$f(x, y, z) = \cos(x)\sin(y)\sin(z).$$

Тогда аналитическое решение задачи представимо в следующем виде:

$$f(x, y, z, t) = \cos(x)\sin(y)\sin(z)\cos(\sqrt{3}t).$$

Невязка вычислений находилась по следующим формулам:

$$\delta_n = \frac{\sum_{i,j,k} |u_{i,j,k}^{n} - u(x_i, y_j, z_k)|}{N^3}, \qquad \delta = \frac{\sum_{0}^{K-1} \delta_n}{K}.$$

Программная реализация

В качестве программной реализации был построен алгоритм на языке C++, осуществляющий блочное трехмерное разбиение сетки $\overline{\omega}_h$ между процессами (см. код в Приложении 1). Для параллельной обработки полученного блочного разбиения алгоритм использует технологии MPI и OpenMP: каждый блок разбиения обрабатывается своим процессом, при этом внутри процесса операторы цикла выполняются различными потоками. На вход алгоритм принимает 4 параметра: линейный размер

сетки $\overline{\omega}_h$ и 3 линейных размера блочного разбиения (размеры указывают число блоков по соответствующей оси). В качестве результата алгоритм выводит на экран время работы алгоритма и полученную при решении невязку. Вычисления производились на ПВС IBM Blue Gene/P и ПВС IBM Polus.

Основная программа хранится в файле main.cpp. Компиляция и запуск программы производится по разному на ПВС IBM Blue Gene/P и IMB Polus. В дальнейшем будем использовать следующие обозначения:

X0 - число процессов;

X1 - линейный размер сетки $\overline{\omega}_h$;

X2, X3, X4 - линейные размеры блочного разбиения.

Компиляция MPI-программы на IBM Polus:
mpicxx main.cpp

Запуск MPI-программы на IBM Polus:
bsub -n <X0> -W 00:30 mpirun -n <X1> a.out <X1> <X2> <X3> <X4>

Компиляция MPI-программы на IBM Blue Gene/P:
mpicxx main.cpp

Запуск MPI-программы на IBM Blue Gene/P:
mpisubmit.bg -n <X0> -m smp -e $"OMP\_NUM\_THREADS = 4"$
a.out <X1> <X2> <X3> <X4>

Компиляция MPI/OpenMP-программы на IBM Blue Gene/P:
mpixlcxx_r -qsmp=omp main.cpp

Запуск MPI/OpenMP-программы на IBM Blue Gene/P:
mpisubmit.bg -n <X0> -m smp -e $"OMP\_NUM\_THREADS = 3"$
a.out <X1> <X2> <X3> <X4>

Дополнительные возможности реализации:

В качестве дополнительного задания была реализована возможность разбиения сетки на произвольное число областей, в том числе и на нечетное. Единственным требованием является кратность размера сетки числу блоков по каждой оси.

Также в качестве дополнительного задания реализована аппроксимация периодических условий, не выходящих за границы области, что достигается путем передачи и хранения каждым процессом соседних для блока граней.

Результаты расчетов

Далее приводятся таблицы с полученными результатами запусков программы на Polus и на Blue Gene P.

Polus				
Число процессоров N_p	Число точек сетки N^3	Время решения T	Ускорение S	Невязка δ
1	128^3	17,09	1,0	1,40E-06
2	128^3	9,82	1,9	1,40E-06
4	128^3	5,85	3,9	1,40E-06
8	128^3	2,70	7,4	1,40E-06
1	256^3	135,88	1,0	6,87E-07
2	256^3	103,38	1,7	6,87E-07
4	256^3	45,23	3,9	6,87E-07
8	256^3	18,34	8,6	6,87E-07
1	512^3	1071,39	1,0	3,41E-07
2	512^3	673,84	1,7	3,41E-07
4	512^3	363,41	3,9	3,41E-07
8	512^3	140,32	8,4	3,41E-07

Blue Gene/P							
Число процессоров N_p	Число точек сетки N^3	Невязка δ	Время решения T_{MPI}	Ускорение S_{MPI}	Время решения $T_{MPI/OpenMP}$	Ускорение $S_{MPI/OpenMP}$	$\frac{T_{MPI}}{T_{MPI/OpenMP}}$
128	128^3	0,0059	3,3	1,00	0,5	6,60	6,94
256	128^3	0,0059	1,7	0,52	0,2	16,50	7,10
512	128^3	0,0059	0,9	0,27	0,1	33,00	6,26
128	256^3	0,0092	26,1	1,00	3,4	7,68	7,72
256	256^3	0,0092	13,1	0,50	1,7	15,35	7,85
512	256^3	0,0092	6,6	0,25	0,9	29,00	7,57
128	512^3	0,0102	207,6	1,00	26,4	7,85	7,85
256	512^3	0,0102	103,8	2,00	13,0	15,97	7,98
512	512^3	0,0102	52,1	3,99	6,7	31,17	7,82
128	1024^3	0,0051			208,6		
256	1024^3	0,0051			103,3		
512	1024^3	0,0051			52,4		
128	1536^3	0,0026			697,0		
256	1536^3	0,0026			350,9		
512	1536^3	0,0026			174,7		