НИЯУ “МИФИ”

Индивидуальное задание №7

Группа: М19-117
Студент: Кайгородов Александр
Преподаватель: Поляков С.В.

Москва, 2020

Задание 7:

На базе примеров ex11a, ex11b решить смешанную краевую задачу

$$\frac{d}{dx}\left(k(x)\frac{du}{dx}\right)-q(x)u=-f(x),\quad a<x<b,$$

$$u(a)=u_a,\quad u'(b)=u_b,$$

с коэффициентами

$$k(x)=1+\exp\left[-5\left(\frac{x-a}{b-a}\right)\right],\quad q(x)=1-0.5\sin\left[10\left(\frac{x-a}{b-a}\right)\right],$$

$$f(x)=q(x)u(x)-k'(x)u'(x)-k(x)u''(x),$$

с помощью алгоритма параллельной прогонки.

1. Решение реализовать в виде файла ex11c.c.
2. Оценить зависимость погрешности разностного решения от размера сетки.
3. Оценить эффективность распараллеливания MPI-программы.
4. Добавить распараллеливание по трэдам (технология MPI+OpenMP).
5. Оценить эффективность распараллеливания MPI+OpenMP-программы.

Решение:

1. Решение прикреплено в сообщении в виде двух файлов. ex11c.c и ex11c_th.c

2. Оценка зависимости погрешности решения от размера сетки

Размер сетки, ед. узлов	*Погрешность расчета, dmax*
10	`dmax=1.074532e-02`
100	`dmax=1.077703e-04`
500	`dmax=4.310928e-06`
1000	dmax=1.077732e-06
5000	dmax=4.307108e-08
10_000	dmax=1.081586e-08
50_000	dmax=7.626534e-10
100_000	dmax=3.741924e-09

500_000	dmax=1.413485e-09
1_000_000	dmax=1.264308e-07
5_000_000	dmax=9.727325e-06
10_000_000	dmax=1.474486e-04
50_000_000	dmax=3.553959e-03

Как видно из таблицы значений погрешности полученных при работе программы ex11c.c при одном работающем процессе, оптимальным значением разбиения сетки является величина 50_000 узлов. Расчетная схема активно наращивает ошибку вычислений при величине разбиения от 1_000_000.

3. Оценка эффективности распараллеливания MPI-программы.

В алгоритме используется метод прогонки теоретическая эффективность которого составляет от 33 до 100%. Оценить эффективность можно следующим образом.

$$E = Sp \div p = (t1 \div tp) \div p = t1 \div (tp \times p)$$

Где p - число запущенных процессов, tp - время решения исходной задачи по параллельному алгоритму на p процессорах, t1 - время решения исходной задачи на одном процессоре.

$$S_p = \frac{p}{3C_2 / C_1 + 2p^2 / N},\ E_p = \frac{1}{3C_2 / C_1 + 2p^2 / N} \cdot 100\%.$$

Проведем натурный эксперимент на конкретной задаче:

t1 = 3.599710 с на сетке 20_000_000 узлов.

Число процессов	Время на выполнение, с	Эффективность
2	1.970207	0.914
4	0.9028028	0.997
7	0.5690598	0.904

10	0.4073280	0.884
20	0.2443905	0.737
40	0.1260414	0.714
80	0.188364	0.234

Как и ожидалось, время выполнения программы уменьшается, эффективность падает, но все в разумных пределах. В момент когда колличество процессов превышает ~50-60 начинается возрастающий рост времени на обмен сообщениями между процессами и как следствие резко падает эффективность распараллеливания. На опытных данных видно что время выполнения на 80 процессах получается даже больше чем на 40, что явно нельзя назвать положительным моментом.

4. Добавить распараллеливание по трэдам (технология MPI+OpenMP).

Файл прикреплен в сообщении.

5. Оценить эффективность распараллеливания MPI+OpenMP-программы.

Идея распараллеливания по тредам заключалась в том, что-бы выбрать отдельную часть программы с использованием метода прогонки, выделить эту часть в отдельную функцию, определить ее выполнение на каждом отдельном процессе, но уже с применением распараллеливания по тредам. Главный поток (работающий с первым узлом процесса) занимается передачей информации остальным процессам, что собственно раньше выполнялось процессом самостоятельно.

Использование многопоточности приводит к увеличению используемой памяти при вычислении на процессоре вследствие введения дополнительных переменных и контейнеров, но чисто вычислений остается прежним. Как итог, для оценки сложности данного алгоритма с использованием многопоточности сокращает число вычислений на n-тредов раз, но время работы все же слегка ограничено периодическим ожиданием отработки первого потока процесса остальными, перед сведением результатов в единый результат и временем работы этого потока на задачу выдачи или сбора результатов от остальных процессов.

Полученные результаты времени исполнения программы:

P / T	t1	t2	t3
1/1	2.51064	0	2.51064
1/2	2.763222	0	2.763222
1/4	2.895089	0	2.895089
2/2	2.041161	0.19059	2.31758
2/4	1.714603	0.011418	1.72602
2/6	1.533181	1.557536	1.548757

По полученным результатам сложно говорить об ускорении четко. Видно увеличение времени исполнения программы при распараллеливании на треды при одном процессе, но с другой стороны получаем уже относительно заметное ускорение при распараллеливании на треды при выполнении на двух процессах. При выполнении на двух процессах, общее число выполняемых нитей равняется np*nt.