

Лабораторная работа №3 «Использование Message Passing
Interface для распараллеливания программ методом операций
двухточечного обмена»

Оглавление

1. Цель работы.....	3
2. Пример использования и схема метода.....	3
3. Ход работы	4
4. Выводы.....	8
Литература	8

1. Цель работы

Получить навык работы с платформой MPI в использовании операций двухточечного обмена.

2. Пример использования и схема метода

В качестве примера использования операций двухточечного обмена рассмотрим численное решение задачи Дирихле для уравнения Лапласа в одномерном и двумерном случаях:

$$\Delta u = 0$$

$$u|_{\Gamma} = \varphi$$

где φ - значение решения на границе области,

Δ - оператор Лапласа

$$\Delta = \frac{\partial^2}{\partial x^2} \text{ - одномерный; } \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \text{ - двумерный}$$

Численное решение основано на введении 1- или 2-мерной сетки в области, ограниченной границей Γ

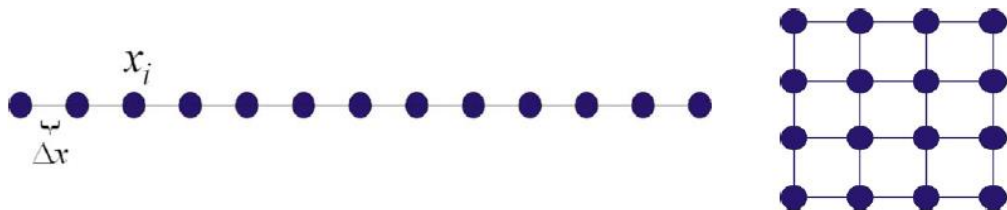


Рисунок 1 – Схема численного решения уравнения

Метод Якоби является итерационным методом. Сначала задаются произвольные значения функции u во внутренних узлах сетки, затем выполняются итерации (1-мерный случай):

$$u_i^{(k+1)} = \frac{1}{2} \left(u_{i-1}^{(k)} + u_{i+1}^{(k)} - \Delta x^2 u_i^{(k)} \right)$$

где $u_i^{(k)} = u^{(k)}(x_i)$ - значение u в i -м узле, полученное на k -й итерации.

Так как численный метод Якоби итерационный, он легко распараллеливается. Параллельный алгоритм основан на декомпозиции по

данным – разбиении одномерной сетки на одинаковые части. Каждая часть обрабатывается на отдельном процессоре. Обмен заключается в пересылке значений функции в граничных узлах. Он может быть организован с помощью операций двухточечного обмена:



Рисунок 2 – Схема двухточечного обмена

3. Ход работы

В эксперименте будет использоваться стандартный блокирующий двухточечный обмен. Блокирующие прием/передача - приостанавливают выполнение процесса на время приема или передачи сообщения.

В работе будет меняться число потоков до 15. Для изменения трудоёмкости задачи, будем менять число итераций (процессов) алгоритма, при распределении по 1000 порций данных (узлов).

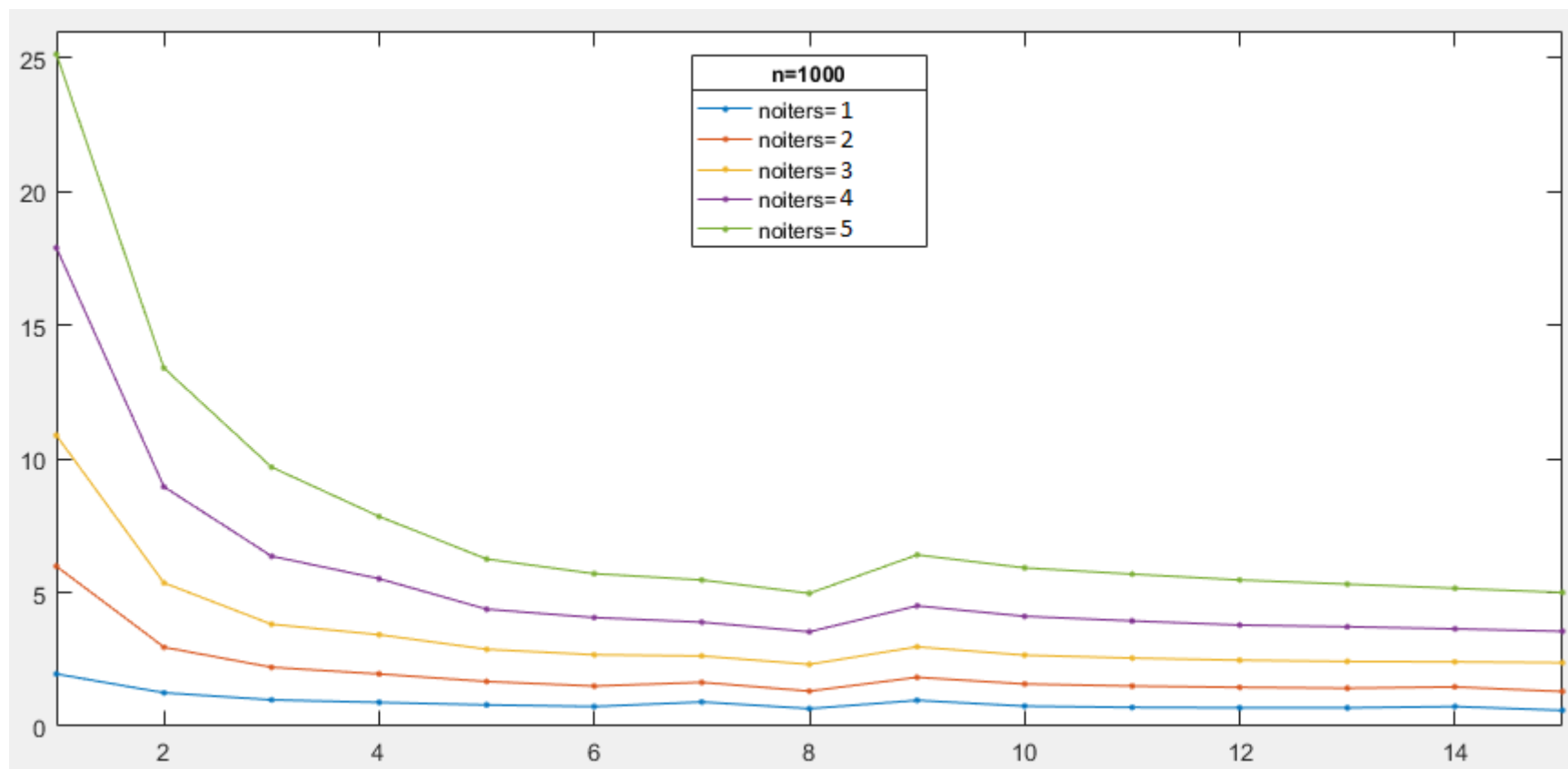


Рисунок 3 – Время выполнения программы при фиксированном значении порций данных

Заметно, что масштабируемость полностью соответствует закону Амдала. При увеличении количества процессов, время выполнения увеличивается, из-за затрат на время передачи данных.

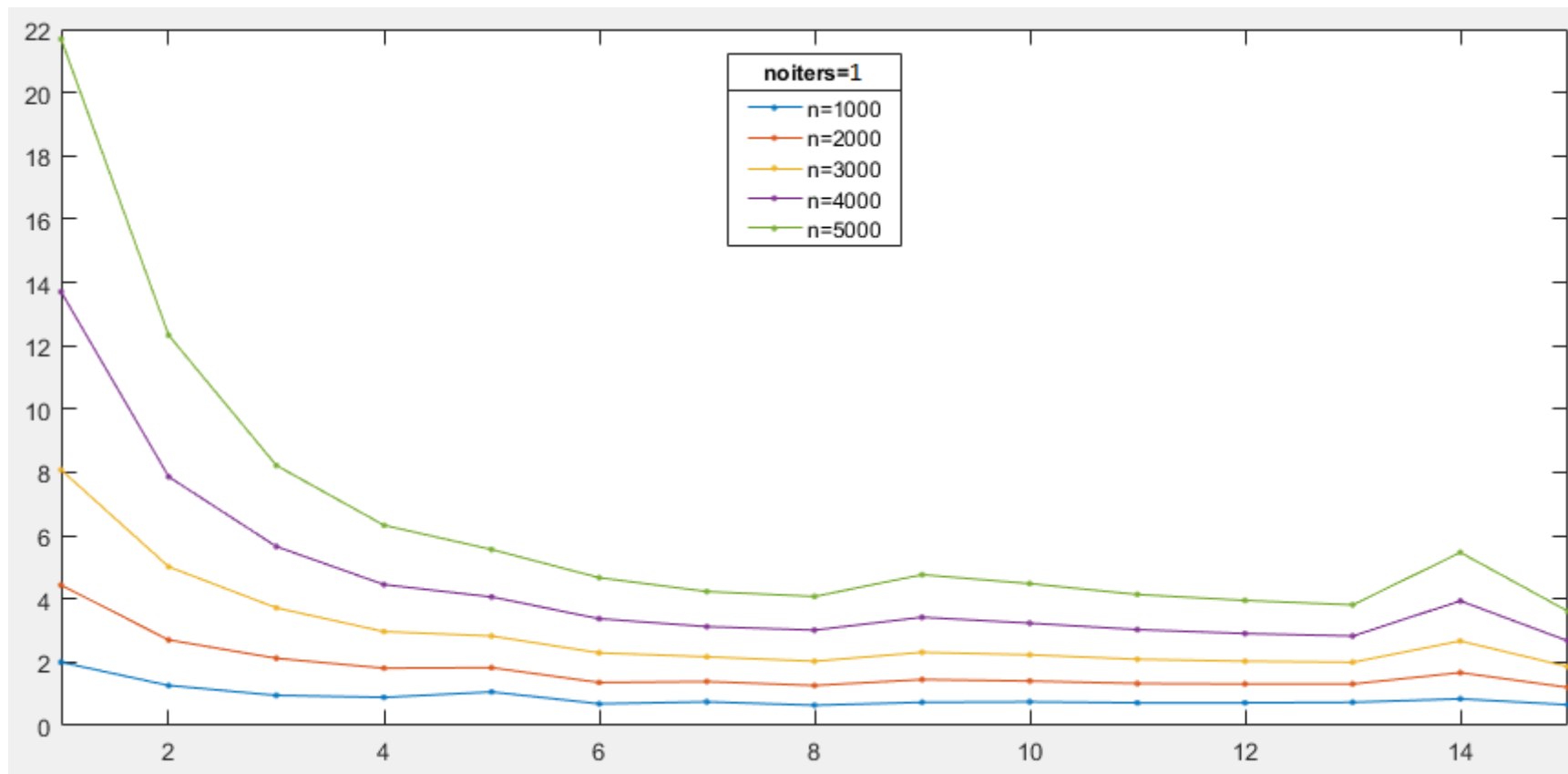


Рисунок 3 – Время выполнения программы при фиксированном числе итераций алгоритма

Уменьшение порций данных влияет на время выполнения в лучшую сторону, но при этом масштабируемость программы снижается.

4. Выводы

В работе был получен навык работы с платформой MPI в использовании операций двухточечного обмена. Было выявлено, что с увеличением числа процессов и размера порций данных, время выполнения программы увеличивается. Связано это с затратами на передачу данных между процессами.

Литература

[1] Основы параллельного программирования с использованием MPI. Лекция 5 – Немнюгин С.А.:

https://www.intuit.ru/EDI/22_05_15_7/1432246798-28527/tutorial/450/objects/5/files/05.pdf

[2] Средства программирования для многопроцессорных вычислительных систем - Немнюгин С.А. (2007)