Постановка задачи

В данной работе необходимо:

1. Реализовать параллельную программу, решающую СЛАУ методом Гаусса, с использованием библиотеки MPI.
2. Реализовать последовательную программу решения СЛАУ.
3. Провести эксперименты по сравнению скорости работы для СЛАУ разного размера и сделать выводы.

Схема последовательного алгоритма

Одним из самых распространенных методов решения систем линейных уравнений является метод Гаусса.
Вычисления с помощью метода Гаусса заключаются в последовательном исключении неизвестных из системы для преобразования ее к эквивалентной системе с верхней треугольной матрицей. Вычисления значений неизвестных производят на этапе обратного хода.
Метод Гаусса с выбором главного элемента по всей матрице (схема полного выбора).
Прямой ход
На 1-м шаге метода среди элементов aij определяют максимальный по модулю элемент. Первое уравнение системы и уравнение с номером i1 меняют местами. Далее стандартным образом производят исключение неизвестного xi1 из всех уравнений, кроме первого.Будем называть его главным элементом 1-го шага.
Найдем величины qi1 = ai1/a11 (i = 2, 3, …, n), называемые множителями 1-го шага. Вычтем последовательно из второго, третьего, …, n-го уравнений системы первое уравнение, умноженное соответственно на q21, q31, …, qn1. Это позволит обратить в нуль коэффициенты при x1 во всех уравнениях, кроме первого. В результате получим эквивалентную систему
a11x1 + a12x2 + a13x3 + … + a1nxn = b1

a22x2 + a23x3 + … + a2nxn = b2

a33x3 + … + a3nxn = b3

……………………………………

annxn = bn

Матрица A является верхней треугольной. На этом вычисления прямого хода заканчиваются.

Обратный ход.
Из последнего уравнения системы находим xn. Подставляя найденное значение xn в предпоследнее уравнение, получим xn−1. Осуществляя обратную подстановку, далее последовательно находим xn−1, xn−2, …, x1.

Алгоритм нахождения координаты решения по методу Гаусса заключается в следующем:

1. Находим главный элемент по всей матрице (Max). Строку содержащую данный элемент принимаем за главную на текущей итерации.
2. Приводим матрицу коэффициентов к треугольному виду.

3. Осуществляем обратный ход и находим координаты решения по методу Гаусса.

Реализация прямого хода требует $N \sim 2n^3/3$ арифметических операций, а обратного – $N \sim n^2$ арифметических операций.

Схема параллельного алгоритма

При внимательном рассмотрении метода Гаусса можно заметить, что все вычисления сводятся к однотипным вычислительным операциям над строками матрицы коэффициентов системы линейных уравнений. Как результат, в основу параллельной реализации алгоритма Гаусса может быть положен принцип распараллеливания по данным. В качестве базовой подзадачи можно принять тогда все вычисления, связанные с обработкой одной строки матрицы A и соответствующего элемента вектора b.

Определение подзадач

Рассмотрим общую схему параллельных вычислений и возникающие при этом информационные зависимости между базовыми подзадачами. Для выполнения прямого хода метода Гаусса необходимо осуществить (n-1) итерацию по исключению неизвестных для преобразования матрицы коэффициентов A к верхнему треугольному виду.
Выполнение итерации i, 0<=i Далее для продолжения вычислений ведущая подзадача должна разослать свою строку матрицы A и соответствующий элемент вектора b всем остальным подзадачам с номерами k, k При выполнении обратного хода метода Гаусса подзадачи выполняют необходимые вычисления для нахождения значения неизвестных. Как только какая-либо подзадача i, 0<=i

Масштабирование и распределение подзадач по процессорам

Выделенные базовые подзадачи характеризуются одинаковой вычислительной трудоемкостью и сбалансированным объемом передаваемых данных. В случае когда размер матрицы, описывающей систему линейных уравнений, оказывается большим, чем число доступных процессоров, базовые подзадачи можно укрупнить, объединив в рамках одной подзадачи несколько строк матрицы. Однако применение последовательной схемы разделения данных для параллельного решения систем линейных уравнений приведет к неравномерной вычислительной нагрузке процессоров: по мере исключения (на прямом ходе) или определения (на обратном ходе) неизвестных в методе Гаусса для все большей части процессоров все необходимые вычисления будут завершены и процессоры окажутся простаивающими. Возможное решение проблемы балансировки вычислений может состоять в использовании ленточной циклической схемы для распределения данных между укрупненными подзадачами. В этом случае матрица A делится на наборы (полосы) строк вида:

$$A = (A_0, A_2, \ldots, A_{p-1})^T, A_i = (a_{i_0}, a_{i_1}, \ldots, a_{i_{k-1}}),\ i_j = i+jp,\ 0 \le j < k,\ k = n/p$$

Сопоставив схему разделения данных и порядок выполнения вычислений в методе Гаусса, можно отметить, что использование циклического способа формирования полос позволяет обеспечить лучшую балансировку вычислительной нагрузки между подзадачами. Распределение подзадач между процессорами должно учитывать характер выполняемых в методе Гаусса коммуникационных операций. Основным видом информационного взаимодействия подзадач является операция передачи данных от одного процессора всем процессорам вычислительной системы. Как результат, для эффективной реализации требуемых информационных взаимодействий между базовыми подзадачами топология сети передачи данных должны иметь структуру гиперкуба или полного графа.