

# **ПРИМЕНЕНИЕ МЕТОДА ГАУССА ДЛЯ РЕШЕНИЯ СИСТЕМЫ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ БОЛЬШИХ РАЗМЕРНОСТЕЙ В КЛАСТЕРНЫХ СИСТЕМАХ**

**П. П. Аниховский**

*Гомельский государственный технический университет  
имени П. О. Сухого, Беларусь*

Научный руководитель К. С. Курочка

Во многих областях человеческой деятельности существуют инженерные и научные задачи, решение которых невозможно получить в аналитическом виде, а численное решение требует значительных вычислительных затрат. Компьютерное моделирование таких задач зачастую приводит к решению системы линейных алгебраических уравнений (СЛАУ) большой размерности (порядка нескольких десятков и сотен тысяч неизвестных) [1]. Эффективное решение таких систем возможно только в распределенных вычислительных системах.

Большинство существующих методов организации параллельных вычислений реализуется в виде самостоятельных программных интерфейсов (API). Одним из наиболее распространенных интерфейсов является MPI [2].

Для распределенного решения СЛАУ в кластерных системах часто используют модифицированный метод Гаусса с различными вариантами размещения данных в кластере и, как следствие, с различной эффективностью и скоростью решения. Самыми распространенными вариантами размещения являются строчно-, столбцово-циклический и блочный.

Был произведен теоретический и практический анализ приведенных способов размещения данных для выбора наиболее эффективного варианта. Наилучшие результаты были получены при строчно-циклическом размещении матрицы системы в памяти вычислительных узлов.

В данном варианте метода исходная матрица коэффициентов распределяется по компьютерам циклическими горизонтальными полосами с шириной полосы в одну строку. Первая строка матрицы помещается в компьютер 0, вторая строка - в компьютер 1 и т. д.,  $n$ -я строка в узел  $n - 1$  (где  $n$  количество узлов в системе). Затем  $n + 1$ -я строка снова помещается в узел 0,  $n + 2$ -я строка - в узел 1 и т. д.

Сначала текущей строкой является строка с индексом 0 в компьютере 0, затем строка с индексом 0 в компьютере 1 и т. д., и наконец, строка с индексом 0 в последнем по номеру компьютере. После чего цикл по компьютерам повторяется и текущей строкой становится строка с индексом 1 в компьютере 0, затем строка с индексом 1 в компьютере 1 и т. д. Аналогично, последовательно по узлам, начиная с последнего по номеру компьютера, осуществляется обратный ход. Разработанный алгоритм метода представлен на рис. 1.

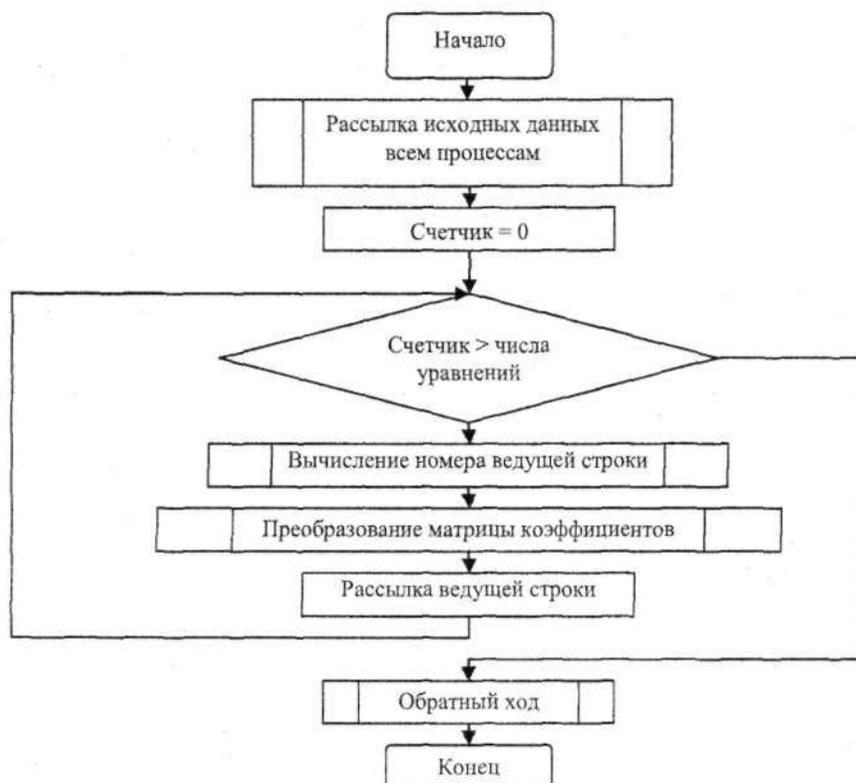


Рис. 1. Алгоритм метода Гаусса со строчно-циклическим размещением данных в кластере

Особенностью этого алгоритма является то, что как при прямом, так и при обратном ходе компьютеры более равномерно загружены, чем в остальных методах. Вычислительная нагрузка распределяется по компьютерам более равномерно. Например, нулевой компьютер, завершив обработку своих строк при прямом ходе, ожидает, пока другие компьютеры обработают только по одной, оставшейся у них не обработанной строке, а не полностью обработают полосы, как в блочном алгоритме. При более равномерной загрузке компьютеров при вычислении одного алгоритма по сравнению с другим алгоритмом следует предположить и большую эффективность алгоритма с более равномерной загрузкой компьютеров. Загрузка компьютеров в циклическом алгоритме является более равномерной. Но большая его эффективность, по сравнению с блочным вариантом, может проявиться только на исходных матрицах большого размера (например, начиная с исходных матриц 400 x 400 и более, но это зависит от конкретной системы).

Для решения СЛАУ методом Гаусса было разработано и верифицировано соответствующее программное обеспечение. Был проведен вычислительный эксперимент в кластере, состоящем из 15-ти узлов, соединенных коммуникационной сетью Ethernet со скоростью 100 Мбит/с. Каждый узел кластера снабжен двухядерным процессором Intel Core 2 Duo E2160 с частотой 1,8 GHz (при решении задействовалось одно ядро), 2 Gb оперативной памяти и 160 Gb винчестером.

Результаты решения СЛАУ различных размерностей приведены в таблице.

Размер матрицы коэффициентов	Время решения, с				
	1 процессор	3 процессора	5 процессоров	10 процессоров	15 процессоров
2883 x 2883	83,4	33,3	36,5	38,9	36,0
5043 x 5043	444,6	136,7	121,9	120,1	116,2
7803 x 7803	1639,2	446,4	360,5	289,6	276,9
11123 x 11123	4508,8	1225,7	931,4	649,8	573,3

#### Выводы:

1. Если время решения СЛАУ сопоставимо со временем передачи данных, то в этом случае нецелесообразно организовывать параллельные вычисления. В нашем случае оказалось, что системы размерностью до 2000 элементов распараллеливать неэффективно, поскольку последовательный вариант для этих систем работает быстрее параллельного.

2. При решении систем больших размерностей (от 5000 и выше) увеличение числа узлов позволяет значительно сократить время решения. В этом случае чем больше размерность системы, тем больший выигрыш во времени решения при работе параллельного варианта по сравнению с последовательным. Но при значительном увеличении числа процессоров выигрыш в скорости нахождения решения снижается. Для системы размерностью порядка 11000 элементов время решения на 3-х узлах почти в четыре раза меньше, чем на одном, а время решения на 15-ти узлах почти в восемь раз меньше, чем в последовательном варианте. Суперлинейное ускорение для систем таких размерностей при реализации на 3-х узлах можно объяснить большим количеством кэш-попаданий процессора.

3. Для каждой из рассмотренных систем уравнений можно найти оптимальное число вычислительных узлов, обеспечивающее наиболее эффективное решение с точки зрения временных затрат.

#### Литература

1. Быховцев, В. Е. Интегральный метод построения математической модели и алгоритма исследования вязкоупругих деформаций грунтовых оснований / В. Е. Быховцев, В. Е. Сеськов, К. С. Курочка // Вестн. Белорус, нац. техн. ун-та. - 2008. - № 4. - С. 17-24.
2. MPI-The Complete Reference: The MPI Core, 2nd edition / M. Snir, S. Otto, Cambridge: MIT Press, USA, ISBN: 0262692155, 1998. - 426 p.