

Влияние распараллеливания вычислений с поверхностными межпроцессорными границами на масштабируемость параллельного итерационного алгоритма решения систем линейных уравнений на примере уравнений вычислительной гидродинамики

С.А. Харченко

Работа посвящена разработке параллельных алгоритмов решения систем линейных уравнений, возникающих при аппроксимации задач вычислительной гидродинамики в пакете **FlowVision**. Основное распараллеливание в алгоритме для трехмерных задач основано на распределении по процессорам больших трехмерных объемов. С ростом числа процессоров относительная доля числа ячеек, принадлежащих двумерным межпроцессорным границам, возрастает. В силу закона Амдала для хорошей масштабируемости всего алгоритма необходимо эффективно распараллеливать также и вычисления, связанные с этими границами. В работе приводятся алгоритмы распараллеливания вычислений, связанных с межпроцессорными границами. Для набора тестовых задач показывается влияние соответствующей организации вычислений на масштабируемость итерационного алгоритма с ростом числа используемых процессоров.

1. Введение

Моделирование задач вычислительной аэро-гидродинамики является актуальной проблемой многих отраслей промышленности. Для воспроизведения тонких физических эффектов в геометрически сложных трехмерных объектах требуются подробные расчетные сетки, содержащие от сотен тысяч до десятков и даже сотен миллионов расчетных ячеек. Подобные задачи требуют огромных вычислительных ресурсов и могут быть решены только на самой современной мощной параллельной вычислительной технике. В связи с этим возникает необходимость обеспечить эффективность использования такой техники при решении аэро-гидродинамических задач.

При моделировании задач вычислительной гидродинамики в пакете “**FlowVision**” с использованием неявных численных методов возникает необходимость решения систем линейных уравнений

$$Ax = b \quad (1)$$

с неструктурированными сильно разреженными матрицами большой размерности. В настоящее время для эффективного решения таких систем линейных алгебраических уравнений (СЛАУ) традиционно используются предобусловленные итерационные методы. В данной работе в качестве предобуславливателя используется предобуславливатель ILU2, являющийся несимметричным обобщением предобуславливателя ICN2, описанного в работе [1]. В работе [1] описаны свойства ICN2: доказано существование и устойчивость этого неполного разложения для произвольных симметричных положительно-определенных матриц, а также получены оценки скорости сходимости метода сопряженных градиентов с этим предобуславливанием. В качестве итерационной схемы используются методы типа подпространства Крылова: метод сопряженных градиентов CG [2] в симметричном случае, алгоритм GMRES [3] и алгоритм Ланцоша [2] в несимметричном случае.

Для эффективного решения системы уравнений (1) на параллельной вычислительной технике необходимо некоторым образом эффективно распараллелить алгоритм решения этой системы. Основное распараллеливание больших трехмерных объемов осуществляется в данной работе на основе упорядочивания по вложенным сечениям Nested Dissection (ND) [4]. Это упорядочивание естественным образом порождает декомпозицию области на число подобластей – степень двойки, и выделяет соответствующие поверхностные межобластные границы. Каждая подобласть и поверхностные границы естественным образом приписываются

некоторым процессорам, которые и производят соответствующие параллельные вычисления. Как оказывается, подобная организация вычислений позволяет в том числе добиться практической независимости производимых вычислений от числа процессоров. Тем не менее, в организации параллельных вычислений в соответствии с упорядочиванием по вложенным сечениям есть одно слабое место с точки зрения масштабируемости – это размер получаемых межпроцессорных границ, а значит доли последовательных вычислений. Чтобы исправить эту ситуацию в данной работе предлагается некоторый способ распараллеливания вычислений с поверхностными границами. В работе проводится сравнение результатов численных экспериментов с распараллеливанием поверхностных границ и без их распараллеливания для некоторой тестовой задачи.

2. Основное объемное распараллеливание

Для явного выделения параллелизма вычислений используется подходящая нумерация переменных, при которой явным образом выделяются независимые и зависимые вычисления [5,6]. Пусть для некоторой нумерации неизвестных и некоторого целого $p > 1$ матрица системы уравнений имеет следующий блочный вид:

$$A = \begin{bmatrix} A_1 & & 0 & C_1 \\ & \ddots & & \vdots \\ 0 & & A_p & C_p \\ B_1 & \cdots & B_p & D \end{bmatrix}. \quad (2)$$

В таком случае мы будем говорить, что матрица системы уравнений имеет p -блочно окаймленный вид. Хорошо известной особенностью представления (2) является то, что в таком представлении явно выделяются независимые вычисления для p процессоров. Эти блочно независимые вычисления связаны с независимой обработкой диагональных блоков A_i .

Представление (2) будем использовать в частном случае $p=2$. Хорошо известное

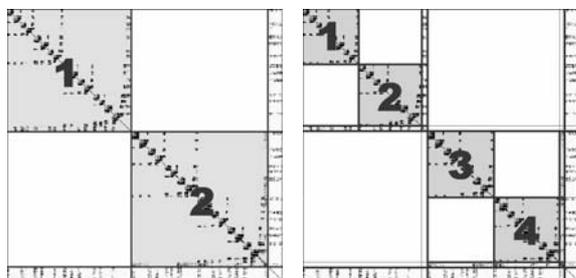


Рис. 1 ND разбиение на 1 и 2 уровня рекурсии

упорядочивание вложенных сечений Nested Dissection (ND) рекурсивным образом строит представление (2) для $p=2$ с минимальным размером разделителя – размером матрицы D , и близкими насколько возможно размерами блоков A_1 и A_2 . Далее это представление снова строится уже для диагональных блоков A_j , и т.д. В результате получается упорядочивание, показанное на Рис. 1, на рисунке показаны

первый и второй уровни рекурсии. Можно выделять и дальнейшие уровни рекурсии до тех пор, пока есть элементы в диагональных блоках. Уровням рекурсии упорядочивания ND соответствует некоторое бинарное дерево. На Рис. 2 показан пример бинарного дерева, которому соответствует разбиение матрицы до трех уровней рекуррентного биения. Корень дерева соответствует самому внешнему окаймлению, листья дерева на верхнем уровне – самым внутренним диагональным блокам, элементы дерева на промежуточных уровнях – соответствующим окаймлениям.

На Рис. 8 показан пример декомпозиции трехмерной области на 2, 4 и 8 процессоров на основе упорядочивания по ND. На этом рисунке хорошо видно, что каждое следующее биение строится как биение на две части всех предыдущих объемных подобластей.

Объемное распараллеливание вычислений осуществляется в соответствии с бинарным деревом [5]. Пусть L и U соответственно нижний и верхний треугольный фактор предобуславливателя ILU2.

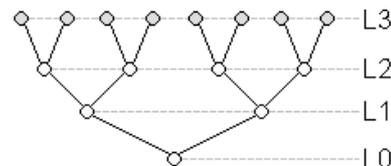


Рис. 2 Бинарное дерево зависимых вычислений

Каждому узлу бинарного дерева соответствует некоторый набор строк U/столбцов L. Параллельные вычисления, связанные с каждой такой группой строк/столбцов, проводятся в соответствии с бинарным деревом: неполное треугольное разложение – от листьев дерева к корням, решение системы уравнений с L – от листьев дерева к корням, при решении с U – от корней дерева к листьям. При этом возможность выполнять последующее вычисление с данными некоторого узла в параллельном режиме определяется во всех ситуациях только готовностью данных в узлах, от которых идут входные дуги по дереву к текущему узлу, и не зависит от состояния остальных узлов дерева.

3. Предсказание разреженности дополнения по Шуру

В описанном параллельном алгоритме решения системы линейных уравнений вычисления с каждой межобластной границей (с матрицей D в представлении (2)) проводятся на одном процессоре. В силу закона Амдала это изначально ограничивает масштабируемость такого алгоритма. Рассмотрим способ, при котором можно распараллелить также и эти вычисления.

Основой подобного распараллеливания может являться собственно *неполнота* треугольного разложения. Действительно, при неполных вычислениях неполные треугольные факторы L и U сильно разрежены, а значит можно ожидать, что подматрица, которая факторизуется при работе с окаймлением, будет также сильно разрежена.

Хорошо известно, что при точном треугольном разложении в представлении (2) матрица, которую необходимо факторизовать при работе с окаймлением, имеет вид матрицы дополнения по Шуру:

$$S = D - \sum_{i=1}^{i \leq p} B_i A_i^{-1} C_i. \quad (3)$$

При точном треугольном разложении эта достаточно сильно заполнена – из-за присутствующих в этой формуле матриц, обратных к матрицам диагональных блоков. При неполном разложении реально вычисленная по неполным факторам приближенная матрица дополнения по Шуру сильно разрежена. Более того, оказывается, что с некоторой степенью точности можно предсказать ее структуру разреженности используя только представление (2).

Пусть Ncycle – некоторое целое положительное число: Ncycle >= 1. В терминах блочного представления (2) Ncycle раз произведем структурное умножение разреженности матрицы C_i на A_i. Пусть Z_i структурно описывает результат умножения, т.е., Sp[Z_i] = Sp[(A_i)^{Ncycle} C_i], где символ Sp используется для обозначения равенства по структуре. Формально точная матрица дополнения по Шуру есть матрица вида $D - \sum_{i=1}^{i \leq p} B_i A_i^{-1} C_i$. Заменяя структурно пару

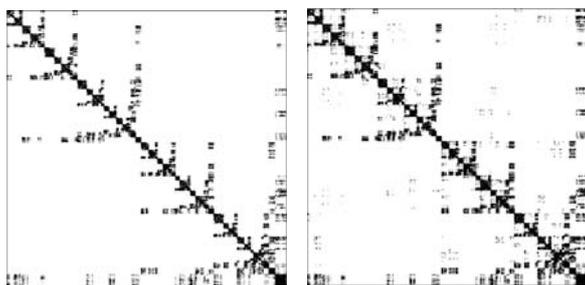


Рис. 3 Разреженность дополнения по Шуру структурно оцененная (слева) и явно вычисленная (справа)

$A_i^{-1} C_i$ на Z_i , аппроксимируем разреженность дополнения матрицы по Шуру разреженностью выражения

$$D - \sum_{i=1}^{i \leq p} B_i Z_i, \quad \text{что соответствует}$$

полиномиальной аппроксимации разреженности матрицы A_i^{-1} . На Рис. 3 в одном и том же упорядочивании для сравнения показаны предсказанная разреженность матрицы дополнения по Шуру с Ncycle=2 и реально вычисленная

неполная разреженность. Из сравнения рисунков видно, что подобный способ вычисления приближенной разреженности достаточно хорошо предсказывает основные области заполнения матрицы дополнения по Шуру.

4. Декомпозиция поверхностной границы и параллельные алгоритмы

Предсказание разреженности матрицы дополнения по Шуру позволяет предложить естественные механизмы распараллеливания вычислений при работе с поверхностными межобластными границами. Для начала рассмотрим границу между двумя подобластями. На Рис. 4 нарисована поверхностная граница раздела между двумя объемными подобластями для некоторой тестовой задачи.

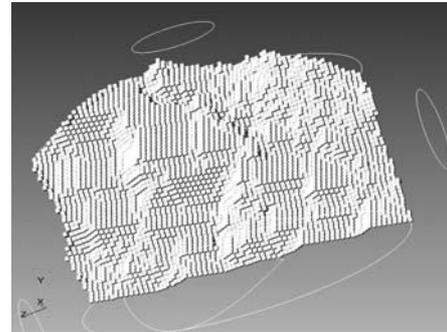


Рис. 4 Поверхностная граница для двух областей

Распределить эту границу по процессорам можно следующим образом. Вычисляем предсказание структуры разреженности матрицы дополнения по Шуру для этого множества ячеек сетки. Производим декомпозицию этого множества по расширенной структуре разреженности на две части, снова с использованием упорядочивания ND. Получаем результат, показанный на Рис. 5.

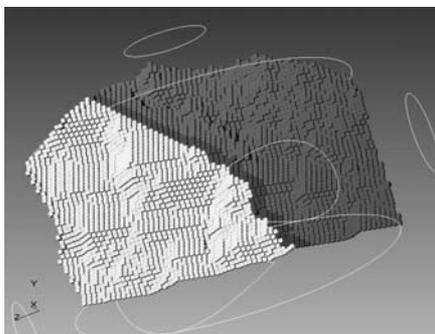


Рис. 5 Декомпозиция поверхностной границы для двух областей

При этом одна часть поверхностной границы отдается одному процессору, вторая – другому, а “линия раздела” между ними – любому из двух процессоров. “Линия раздела” на этом рисунке трехслойная поскольку для оценки разреженности матрицы дополнения по Шуру использовалось значение $N_{cycle} = 2$.

В случае большего числа процессоров эта процедура повторяется с некоторыми изменениями. Прежде всего, каждой поверхностной границе соответствует ее уровень в бинарном дереве зависимостей. Поверхностная граница должна быть распределена только по тем процессорам, к объемам которых она примыкает, иначе неизбежны лишние

синхронизации и обмены. Ясно также, что границу-узел дерева между двумя верхними листьями дерева зависимостей можно распределять так, как было описано выше для двух подобластей.

Границы, соответствующие узлам дерева, более близких к корню, распределяем из соображений попарного примыкания к границе. А именно, вся текущая поверхностная граница разбивается на наборы несвязных множеств, к которым примыкают объемы ровно двух процессоров. Каждое такое множество разбивается на два примыкающих процессора как описано выше, и границу. В результате получается некоторый набор кусочков границы, разделенных линиями раздела. На Рис. 6 для той же задачи показано разбиение той же внешней границы на 4 процессора, в другом ракурсе показана декомпозиция остальных границ.

По существу, разбиение граничных поверхностей на подмножества соответствует следам объемных распределений на поверхностях. После декомпозиции каждого подмножества получившиеся его части распределяются именно тем процессорам, объемные распределения которых примыкают к этому множеству. Это сделано для того, чтобы при вычислениях минимизировать дополнительные обмены и вычисления с чужими данными.

Блочная разреженность матрицы при подобной декомпозиции задачи обладает некоторыми особенностями. На Рис. 7 приведен пример блочной структуры разреженности матрицы для тестовой задачи в случае 2-х, 4-х и 8-ми подобластей.

Распараллеливание вычислений с поверхностными границами предлагается осуществлять на основе блочной структуры разреженности матрицы исходя из естественного распределения блоков по процессорам. Различные вычисления с частями поверхностной границы считаются зависимыми только если между ними есть связь в этой части блочной структуры разреженности матрицы. Это означает, что в процессе вычислений фильтруются те элементы

неполных разложений, которые выходят за рамки заданной блочной разреженности матрицы. В терминах блоков вычисления ведутся “по позициям”, внутри блоков вычисления ведутся “по значениям”.

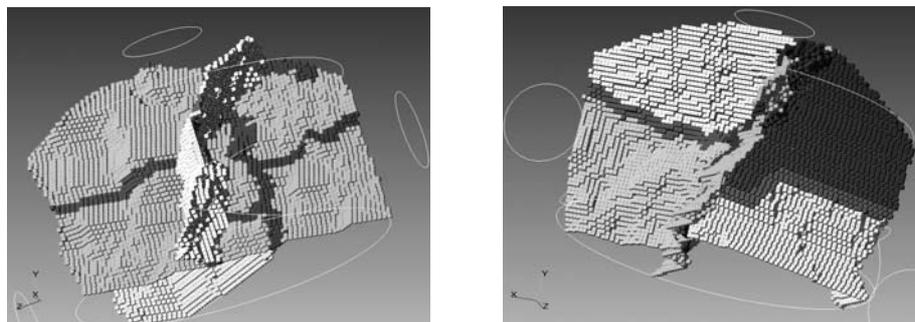


Рис. 6 Декомпозиция поверхностных границ для четырех областей в двух ракурсах

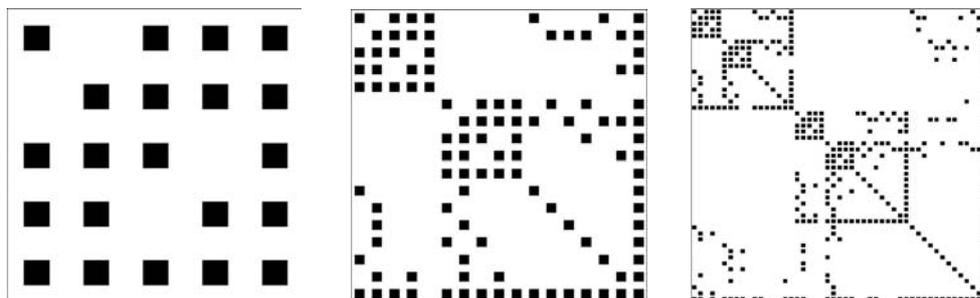


Рис. 7 Блочная структура разреженности матрицы для двух, четырех и восьми подобластей

5. Результаты

Для исследования свойств масштабируемости предложенных параллельных алгоритмов решения систем уравнений была выбрана существенно трехмерная тестовая задача о течении воздуха в смесителе. Для этой задачи была построена расчетная сетка с количеством расчетных ячеек $N = 405600$. Геометрия задачи и ее декомпозиция на 2, 4 и 8 процессоров на основе упорядочивания вложенных сечений показана на Рис.8.

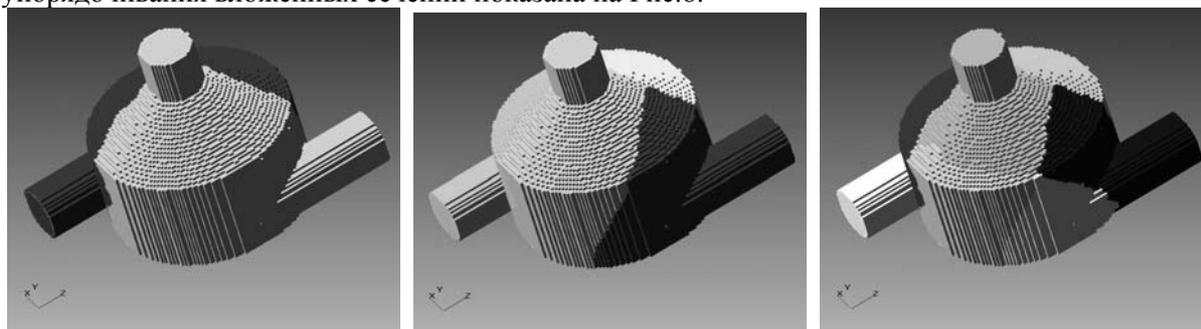


Рис. 8. Декомпозиция тестовой задачи на 2, 4 и 8 процессоров

Для тестовой задачи рассматривалась система линейных уравнений для давления на некотором шаге по времени. Результаты экспериментов для тестовой системы линейных уравнений при распараллеливании по MPI представлены на Рис. 9. Эксперименты проводились на кластере МВС-6000, составленном из двухпроцессорных Intel Itanium 2, соединенных транспортной сетью Myrinet 2000 с коммутатором Myricom M3-128. Эксперименты проводились в режиме “один процессор на узел”, чтобы исключить влияние на результаты вычислений шины общей памяти. Во всех экспериментах число итераций не зависело от числа процессоров, и было равно 54 без распараллеливания поверхностной границы и 63 с распараллеливанием поверхностной границы. Во всех экспериментах измерялось астрономическое время выполнения параллельных вычислений.

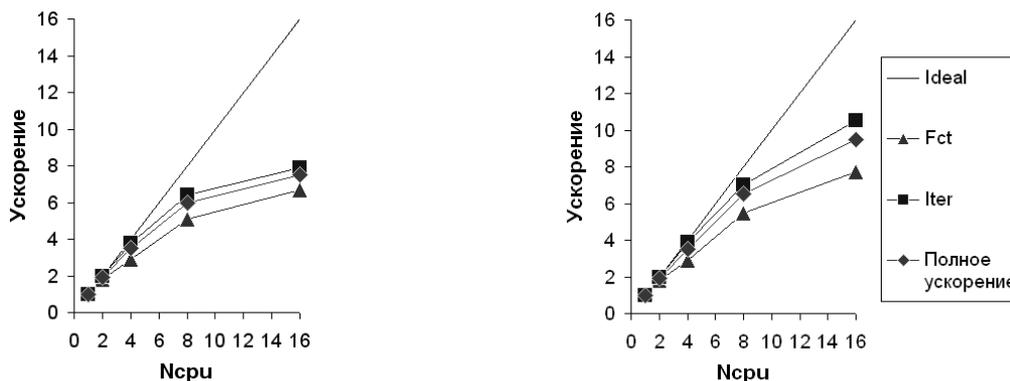


Рис. 9. Результаты численных экспериментов без (слева) и с учетом (справа) распараллеливания поверхностных границ

Результаты, показанные на Рис. 9 показывают, что даже на очень маленькой тестовой задаче предложенный алгоритм решения системы линейных уравнений может показывать отличное ускорение – до 10 раз на 16 процессорах. При этом распараллеливание поверхностных межпроцессорных границ несколько увеличивает число итераций – с 54 до 63, однако при этом настолько сильно улучшает масштабируемость для 16 процессоров, что суммарное время вычислений может оказаться меньшим (6.2 сек. против 7.1 сек.) даже при большем числе итераций.

6. Заключение

Основной целью данного исследования является разработка масштабируемого параллельного алгоритма решения систем уравнений на кластерных вычислительных системах. Представленные результаты численных экспериментов показывают, что предложенные в работе алгоритмы распараллеливания поверхностных межпроцессорных границ позволяют улучшить масштабируемость параллельного алгоритма решения системы уравнений ценой некоторого ухудшения качества предобуславливания. Предложенный в работе способ организации параллельных вычислений позволяет добиться независимости вычислений от числа процессоров на некотором выбранном интервале чисел процессоров, что подтверждается численными экспериментами.

Литература

1. I.E.Kaporin. "High Quality Preconditioning of a General Symmetric Positive Definite Matrix Based on its $U^T U + U^T R + R^T U$ decomposition". Numer. Linear Algebra Appl., 5, 483-509 (1998).
2. Тыртышников Е.Е. Краткий курс численного анализа. Москва: ВИНТИ. 1994.
3. Saad Y., Schultz M.H., "GMRES: A generalized minimum residual algorithm for solving non-symmetric linear systems". SIAM J. Sci. Comput. 1986. 7. 856-869.
4. George A., Liu J.W., "Computer Solution of Large Sparse Positive Definite Systems", Prentice Hall, 1981.
5. Харченко С.А. "Параллельная реализация алгоритма решения систем линейных уравнений в пакете **FlowVision**", в сборнике "Прикладные исследования в механике", труды конференции "Инженерные системы -2007", 135-144, 2007.
6. А.А. Дядькин, С.А. Харченко. "Алгоритмы декомпозиции области и нумерации ячеек с учетом локальных адаптаций расчетной сетки при параллельном решении систем уравнений в пакете **FlowVision**". Труды Всероссийской научной конференции "Научный сервис в сети Internet: многоядерный компьютерный мир", 201-206, 2007.