

Экспериментальное исследование на СКИФ МГУ “Чебышев” комбинированной MPI+threads реализации алгоритма решения систем линейных уравнений, возникающих во FlowVision при моделировании задач вычислительной гидродинамики

Г.Б. Сушко, С.А. Харченко

При моделировании задач вычислительной гидродинамики во FlowVision с использованием неявных численных методов возникает необходимость решения систем линейных уравнений, описываемых неструктурированными разреженными матрицами большой размерности. Во FlowVision при решении таких систем уравнений сначала вычисляется неполное разложение высокого порядка точности (ICN2/ILU2) матрицы коэффициентов системы уравнений, а затем система уравнений решается с использованием предобусловленного итерационного алгоритма типа подпространства Крылова. Для уменьшения времени вычислений этот алгоритм а priori распараллеливается по распределенной памяти с использованием стандарта MPI, а по общей памяти с динамическим распределением нагрузки на основе технологии Intel® Threading Building Blocks (TBB). Для явного выделения параллелизма в зависимых вычислениях неполной факторизации и при решении систем уравнений с треугольными матрицами используется упорядочивание типа вложенных сечений Nested Dissection (ND). Приводятся результаты численных экспериментов по масштабируемости предложенных алгоритмов на СКИФ МГУ “Чебышев” для систем линейных уравнений, возникающих при моделировании задач вычислительной гидродинамики во FlowVision

1. Введение

Моделирование задач вычислительной аэро- и гидродинамики является актуальной проблемой многих отраслей промышленности. Параллельная версия программного комплекса **FlowVision**, разрабатываемого в ООО “ТЕСИС”, используется для моделирования широкого круга индустриальных задач, включая задачи с подвижными телами и свободными поверхностями для областей со сложной геометрией. Для адекватного воспроизведения тонких физических эффектов в геометрически сложных трехмерных областях 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 [1].

Использование неявных численных методов во **FlowVision** приводит к необходимости 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) используются предобусловленные итерационные методы. Во **FlowVision** в качестве предобуславливателя используется предобуславливатель ILU2, являющийся несимметричным обобщением предобуславливателя ICN2[1]. В работе [1] 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. На практике предобуславливатель ICN2 и ILU2 показали свою высокую надежность и эффективность при решении разнообразных систем линейных уравнений. 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 уменьшения времени счета проводимые вычисления требуется распараллелить в соответствии с архитектурой используемой вычислительной техники. Доступная в настоящий момент техника обладает как правило архитектурой с неоднородным доступом к памяти. А имен-

но, с одной стороны, имеется набор вычислительных узлов с распределенной памятью, обмен данными между которыми может быть осуществлен по быстрой обменной сетке. С другой стороны, каждый узел представляет собой многопроцессорный/многоядерный компьютер с общим доступом к оперативной памяти.

В предыдущих работах [7-10] авторами были предложены алгоритмы распараллеливания вычислений при решении систем уравнений для случая параллельных вычислений с распределенной [7-9] на основе MPI и с общей [10] памятью на основе OpenMP/TBB [5-6]. Алгоритмы, описанные в этих работах, основаны на единых подходах к распараллеливанию вычислений, базируются на рекурсивной геометрической декомпозиции расчетной сетки на основе упорядочиваний типа вложенных сечений Nested Dissection (ND) [4], используют специальные подходы к распараллеливанию вычислений с поверхностными межпроцессорными границами [9], и при этом учитывают адаптацию расчетной сетки [8].

Эффективная хорошо масштабируемая организация параллельных вычислений при большом общем числе доступных процессоров/ядер предполагает хорошую балансировку вычислений на различных процессорах между собой. Чем больше число задействованных процессоров/ядер, тем более точная балансировка вычислений требуется для поддержания заданного уровня масштабируемости. К сожалению, в случае использования распределенной памяти трудно заранее хорошо сбалансировать вычисления если объем вычислений заранее точно неизвестен, а принимать решение об организации вычислений (о распределении данных по процессорам и о взаимодействии процессоров/ядер) приходится до собственно самих вычислений. С другой стороны, если вычисления проводятся по общей памяти, то имеется возможность динамически определить какое ядро/процессор должно проводить текущее вычисление, так как можно организовать вычисления по принципу клиент-сервер с потенциально наилучшей возможной балансировкой нагрузки.

В работе описывается параллельная реализация алгоритма решения СЛАУ с предобуславливателем ICN2/ILU2 и итерационной схемой типа подпространства Крылова для компьютеров с неоднородным доступом к памяти. Для уменьшения времени вычислений этот алгоритм заранее распараллеливается по распределенной памяти (по узлам вычислений) с использованием стандарта MPI, а по общей памяти узла распараллеливание по процессорам/ядрам осуществляется с динамическим распределением нагрузки на основе технологии Intel® Threading Building Blocks (TBB) [6]. Приводятся результаты численных экспериментов по масштабируемости предложенных алгоритмов на СКИФ МГУ “Чебышев” для систем линейных уравнений, возникающих при моделировании во **FlowVision** задач вычислительной гидродинамики.

2. Блочные биения и матрица зависимостей блочных вычислений при решении систем уравнений

Естественным способом явного выделения параллелизма вычислений является введение для основных структур данных подходящей крупно-блочной структуры. При этом явное указание зависимостей вычислений при построении объектов этой блочной структуры естественным образом приводит к понятию матрицы зависимостей блочных вычислений. В терминах такой матрицы для каждого объекта крупно-блочной структуры можно указать от каких других объектов крупно-блочной структуры этот объект зависит.

В данном разделе вводится понятие блочного биения и блочной структуры матрицы системы линейных уравнений, а также в терминах блочных операций описываются основные алгебраические вычисления алгоритма решения системы уравнений. Вводится понятие матрицы зависимостей блочных вычислений, описывается связь этой матрицы со структурой параллельных вычислений для основных вычислений алгоритма. Также в этом разделе приводится способ построения упорядочивания и построения блочного биения для системы линейных уравнений, для которого имеет место большой ресурс параллелизма вычислений, и для которого матрица зависимостей блочных вычислений обладает заданными свойствами.

2.1 Упорядочивание и блочное биение

Предположим, что все неизвестные системы уравнений занумерованы некоторым образом. Нумерация переменных может быть различной в различных случаях и для различных целей. Например, нумерация может быть специально подобрана таким образом, чтобы в этой нумерации явным образом выделялся параллелизм вычислений для заданного числа процессоров и/или чтобы вычисления были максимально эффективны при решении системы уравнений, и т.д. Для выбранной нумерации неизвестных системы уравнений вводим разбиение множества всех индексов неизвестных на некоторое количество подмножеств, например p , $p > 1$, каждое из которых содержит набор индексов неизвестных, расположенных подряд. Нумеруем выбранные множества индексами от 1 до p в соответствии с возрастанием индексов в множествах. Пусть матрица системы уравнений A переупорядочена в соответствии с новой нумерацией неизвестных. Обозначим через A_{ij} блок матрицы системы уравнений, отвечающий наборам индексов i по строке и j по столбцу. Матрица системы уравнений в терминах введенного блочного биения имеет вид:

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1p} \\ \vdots & \ddots & \vdots \\ A_{p1} & \cdots & A_{pp} \end{bmatrix}. \quad (2)$$

Для перенумерованного вектора правой части b обозначим через b_i блок этой правой части, соответствующий набору индексов i из (1). Для остальных векторных и матричных данных алгоритма решения системы уравнений (1) их блочные биения вводятся аналогичным образом.

2.2 Расширенная блочная разреженность как матрица зависимостей блочных вычислений

Алгоритм решения систем уравнений, используемый во **FlowVision**, предполагает неполную факторизацию матрицы системы уравнений и последующие предобусловленные итерации. Построим для этих вычислений описание максимально возможной зависимости по данным с учетом разреженности матрицы системы уравнений.

Неполная треугольная факторизация.

Следуя изложению работы [10], рассмотрим алгоритм (точной) треугольной факторизации – разложения матрицы A в произведение нижней L и верхней U треугольных матриц. В блочном виде рекуррентное соотношение этой операции можно записать следующим образом:

$$\begin{bmatrix} A_{11} & a_{12} & A_{13} \\ a_{21} & a_{22} & a_{23} \\ A_{31} & a_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ L_{31} & l_{32} & L_{33} \end{bmatrix} \cdot \begin{bmatrix} U_{11} & u_{12} & U_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \quad (3)$$

Здесь блок a_{22} имеет размер 1, блоки $a_{21}, a_{23}, l_{21}, u_{23}$ – строки, а $a_{12}, a_{32}, l_{32}, u_{12}$ – столбцы. Рекуррентный обход матрицы производится последовательно по строкам U и столбцам L сверху вниз, т.е. блоки, L_{i1}, U_{1j} , $i, j = 1, \dots, 3$, уже вычислены, блоки $l_{22}, l_{32}, u_{22}, u_{23}$ рассчитываются на данном шаге, а блоки L_{33} и U_{33} пока не известны. На каждом шаге вычислений необходимо:

1. Из уравнения $l_{22}u_{22} = a_{22} - l_{21}u_{12}$ вычислить значения l_{22}, u_{22} .
2. Из уравнений $u_{23} = (a_{23} - l_{21}U_{13})l_{22}^{-1}$ и $l_{32} = (a_{32} - L_{31}u_{12})u_{22}^{-1}$ найти значения u_{23} и l_{32} .

Различные варианты неполных разложений, в том числе и ISH2/ILU2 [1], получаются из описанного выше алгоритма путем некоторой модификации шагов 1 и 2.

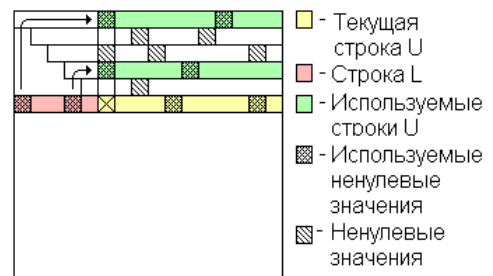


Рис. 1 Зависимость по данным при факторизации строки U

Зависимость по данным в этом алгоритме для текущей строки U показана на Рис. 1, для столбца L зависимость по данным аналогична. Таким образом, при вычислении текущей строки U /столбца L используются некоторые ненулевые значения из уже посчитанных строк U /столбцов L .

Неполная треугольная факторизация в терминах блочных вычислений. Матрица зависимостей блочных вычислений.

В терминах блочного биения (2) неполные вычисления неполной факторизации могут быть снова записаны в виде шагов 1) и 2) с той лишь разницей, что количество шагов теперь будет равным числу блоков блочного биения матрицы, и что в новых формулах вычислений операция умножения элементов матриц заменяется на операцию умножения разреженных подматриц, а операция умножения на обратное диагональное значение заменяется в зависимости от варианта неполного разложения соответствующим приближенным блочным решением систем уравнений с диагональным блоком с учетом разреженности участвующих подматриц. Зависимости по данным при этих вычислениях в терминах блоков снова описывается диаграммой, показанной на Рис. 1, при этом зависимости по данным имеют место только если соответствующие блоки матриц не равны тождественно нулю.

Рассмотрим вопрос о блочной структуре разреженности матриц предобуславливания L и U . С точки зрения качества предобуславливания очевидно блочное заполнение матриц L и U должно быть не меньшим, чем блочное заполнение исходной матрицы системы уравнений. С другой стороны, как это показано выше, максимальное блочное заполнение матриц L и U предобуславливания соответствует так называемой *символьной факторизации* блочной разреженности матрицы системы уравнений. На практике, с учетом неполноты вычислений блочная разреженность матриц L и U занимает некоторое промежуточное положение между этими двумя естественными пределами. Алгоритм распараллеливания вычислений с поверхностными межпроцессорными границами, описанный в работе [9], также предполагает некоторые дополнительное уменьшение заполнения блочной разреженности матриц L и U по сравнению со случаем символьной факторизации блочной структуры разреженности матрицы системы уравнений.

С учетом вышесказанного, для обеспечения фиксированной блочной зависимости параллельных вычислений и для обеспечения гарантированного достаточного ресурса параллелизма при вычислениях для каждого блочного биения матрицы будем рассматривать *матрицу зависимостей блочных вычислений*. Это матрица, структура разреженности которой есть сумма допустимых с учетом всех необходимых требований блочных разреженностей матриц L и U предобуславливания. По определению будем предполагать, что структура разреженности матрицы зависимостей блочных вычислений не меньше, чем блочная структура разреженности матрицы системы уравнений, и не больше, чем сумма разреженностей факторов символьной факторизации матрицы блочной разреженности системы уравнений.

В терминах матрицы зависимостей блочных вычислений легко описываются все зависимости по данным, которые имеют место при построении неполных факторов L и U . С другой стороны, соответственно нижняя и верхняя треугольные части этой же матрицы могут быть использованы для описания зависимостей блочных вычислений при решении системы уравнений с нижней L и верхней U треугольной матрицей. В силу вложения блочной структуры разреженности матрицы системы в структуру разреженности матрицы зависимостей блочных вычислений, эта же матрица может быть использована при описании зависимостей блочных вычислений при умножении матрицы системы уравнений на вектор. Получается, что структура разреженности матрицы зависимостей блочных вычислений тем или иным образом описывает зависимости по данным на всех этапах алгебраического алгоритма решения системы линейных уравнений.

2.3 Адаптация расчетной сетки. Нумерация ячеек, блочные биения и декомпозиция для параллельных вычислений

Эффективная организация параллельных вычислений требует выделения независимых вычислений, которые можно было бы производить одновременно, а также зависимых вычислений, которые можно было бы выполнять максимально сбалансированно и с минимальными задержками. Для того, чтобы выделить подобные вычисления при решении систем уравнений во

FlowVision в случае использования адаптированных расчетных сеток используются подходы, описанные в работах [7-9].

Сначала производится объемная декомпозиция расчетной сетки в терминах так называемых начальных ячеек [8]. Каждая начальная ячейка представляет собой элемент регулярной декартовой начальной сетки, который может содержать некоторое множество реальных ячеек. Количество реальных ячеек в начальных ячейках увеличивается при адаптациях расчетной сетки в соответствии с заданными пользователем критериями. Объемная декомпозиция осуществляется на число процессоров/ядер – степень двойки, и производится последовательным разбиением на две подобласти каждого предыдущего разбиения. Объемная декомпозиция на две подобласти осуществляется с помощью программы ParMeTiS [12] с учетом условий минимального возможного числа реальных ячеек на границе между подобластями и с учетом условия наилучшего возможного совпадения полного числа расчетных ячеек в обеих подобластях.

На основе объемной декомпозиции задачи строятся однослойные поверхностные границы между подобластями [9], эти границы по построению проходят по поверхностным границам начальных ячеек. Внутри каждой полученной объемной подобласти ячейки нумеруются в соответствии с упорядочиванием вложенных сечений ND [4]. Для ячеек на границе между подобластями при помощи специального алгоритма предсказывается разреженность дополнения по Шуру для этой границы [9]. На основе предсказанной разреженности дополнения по Шуру производится декомпозиция поверхностных границ на близкие по числу ячеек поверхностные подобласти по принципу непосредственного примыкания этих поверхностных подобластей к исходным объемным подобластям. Это сделано для выделения максимального параллелизма при работе с поверхностными границами и для минимизации обменов и зависимостей в вычислениях при их обработке. Возникающая при каждой поверхностной декомпозиции одномерная граница между поверхностными подобластями ассоциируется с одной (любой из примыкающей пары) объемной подобластью. После перенумерации матрицы в соответствии с таким образом построенным объемным и поверхностным разбиением возникают структуры разреженности, показанные для тестовой задачи на Рис. 4. На Рис. 5 для той же тестовой задачи показаны соответствующие матрицы зависимостей блочных вычислений.

3. MPI+threads реализация алгоритма решения систем уравнений на компьютерах с неоднородным доступом к памяти

На компьютерах с неоднородным доступом к памяти возможны различные способы организации вычислений. В данной работе при распараллеливании вычислений между узлами для организации обменов и синхронизаций используется стандарт обмена сообщениями MPI. В каждом узле с общей памятью при распараллеливании по процессорам/ядрам для организации синхронизаций по общей памяти используется технология Intel[®] Threading Building Blocks (ТВВ). При этом предполагается, что на каждом узле имеется только один MPI процесс, который затем порождает на этом узле нужное количество одновременно работающих потоков вычислений.

При работе с распределенной и с общей памятью естественными являются различные подходы к распараллеливанию вычислений. С одной стороны, на распределенной памяти для минимизации обменов естественным является заранее распределить данные по процессорам, и при этом заранее по возможности сбалансировать вычисления и минимизировать обмены. С другой стороны, при работе по общей памяти нет необходимости заранее распределять данные по процессорам/ядрам. Для наилучшей возможной балансировки вычислений можно динамически выбирать ядро, которое будет проводить текущие вычисления.

На компьютерах с неоднородным доступом к памяти для достижения по возможности наилучших результатов естественно использовать одновременно оба подхода к распараллеливанию вычислений. А именно, сначала все вычисления заранее распределяются по узлам таким же образом, как это делалось при работе только с распределенной памятью. Затем, каждое вычисление в узле динамически распараллеливается по процессорам/ядрам с общей памятью таким же образом, как это делается в случае только общей памяти. При этом для достижения эффекта самобалансировки вычислений внутри узла полное число объектов, с которыми необходимо

проводить вычисления внутри одного узла, должно быть достаточно большим для обеспечения возможности организации конвейера вычислений.

Описанная выше методика организации параллельных вычислений на компьютерах с неоднородным доступом к памяти в применении к алгоритмам решения систем уравнений реализована следующим образом. Пусть имеется s узлов и t процессоров/ядер в каждом узле, итого всего $p = s * t$ процессоров/ядер. Производится параллельная декомпозиция задачи (в том числе построение блочного биения матрицы системы уравнений и матрицы зависимостей параллельных вычислений) на число процессоров/ядер, равное $p * q$, где $q > 1$ - целое число, обозначающее запас параллелизма для автоматической балансировки вычислений внутри узла. В данной работе использовалось значение $q = 4$. Далее, на каждый узел распределяются все данные, соответствующие $t * q$ условным процессорам/ядрам этого узла. Все параллельные вычисления проводятся в соответствии с блочными структурами соответствующих матриц на основе матрицы зависимостей блочных вычислений, как это описано в разделе 2.2.

4. Результаты численных экспериментов

Для проведения численных экспериментов была выбрана задача об обтекании спортивного автомобиля. Адаптивная расчетная сетка для этой задачи содержит 2.994.497 расчетных ячеек. Структура адаптации сетки в различных ракурсах показана на Рисунке 2.

Объемная декомпозиция расчетной сетки на 16 процессоров в терминах начальных ячеек [8] показана на Рисунке 3. На Рисунке 4 показана разреженность матрицы коэффициентов системы с учетом блочного биения для случаев 16 и 64 процессоров/ядер, построенная в соответствии с описанием Раздела 2.3. На Рисунке 5 представлены структуры разреженности матрицы зависимостей блочных вычислений для случаев 16 и 64 процессоров/ядер.

Численные эксперименты проводились на СКИФ МГУ “Чебышев”. В Таблице 1 представлены результаты численных экспериментов.

Из результатов численных экспериментов следует, что комбинированная MPI+threads версия алгоритма решения систем уравнений позволяет ускорить вычисления до 2 раз по сравнению с чисто MPI версией. Это связано с сильным дисбалансом вычислений, который естественным образом возникает при использовании неполных факторизаций по значениям в условиях существенной адаптации расчетной сетки. Этот дисбаланс удается компенсировать за счет динамической балансировки вычислений внутри узла. При этом положительный эффект наблюдается начиная с 4-х ядер в узле, при использовании 2-х ядер на узел в итерационной схеме наблюдается даже небольшой проигрыш по времени по сравнению с чисто MPI версией. Из таблицы также видно, что увеличение числа используемых ядер на узел до 8 привело к дальнейшему ускорению неполной факторизации и к небольшому замедлению итерационной схемы. Этот эффект по-видимому связан с особенностями переиспользования памяти в этих алгоритмах. В неполной факторизации каждое данное переиспользуется множество раз, в отличие от итерационной схемы, а потому неполная факторизация слабее подвержена влиянию эффекта недостаточной масштабируемости узла с точки зрения работы с оперативной памятью при большом числе используемых ядер.

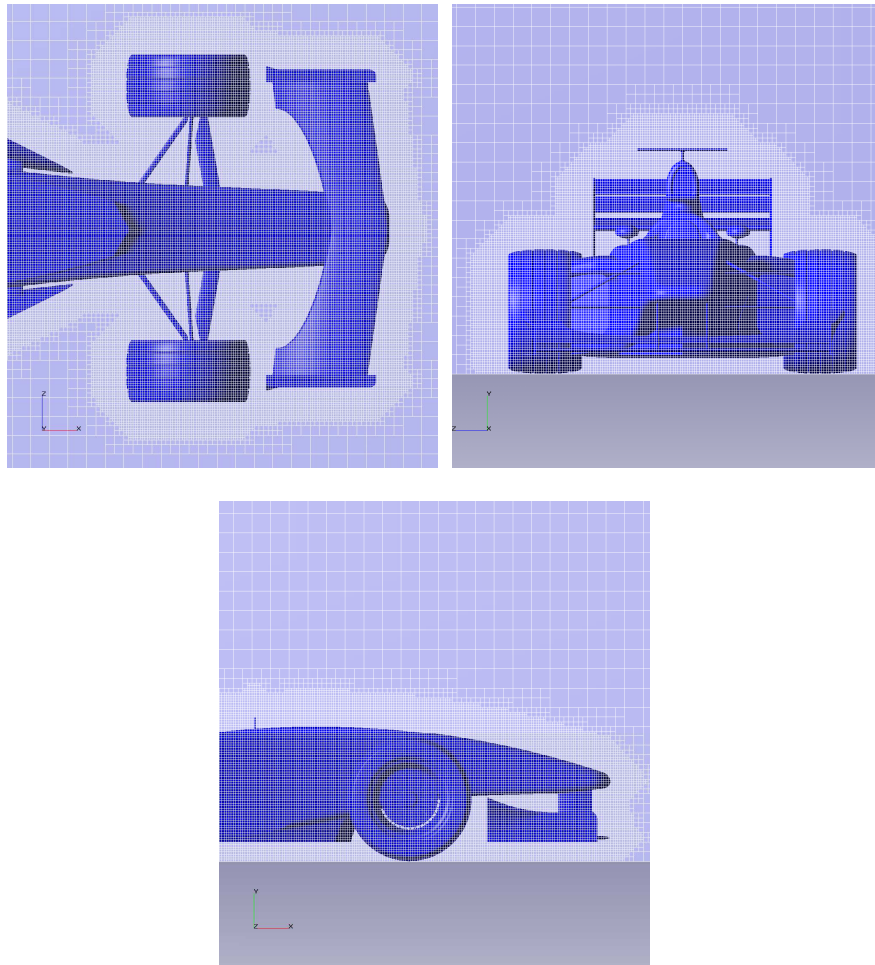


Рис. 2. Адаптации сетки вокруг спортивного автомобиля в различных ракурсах.

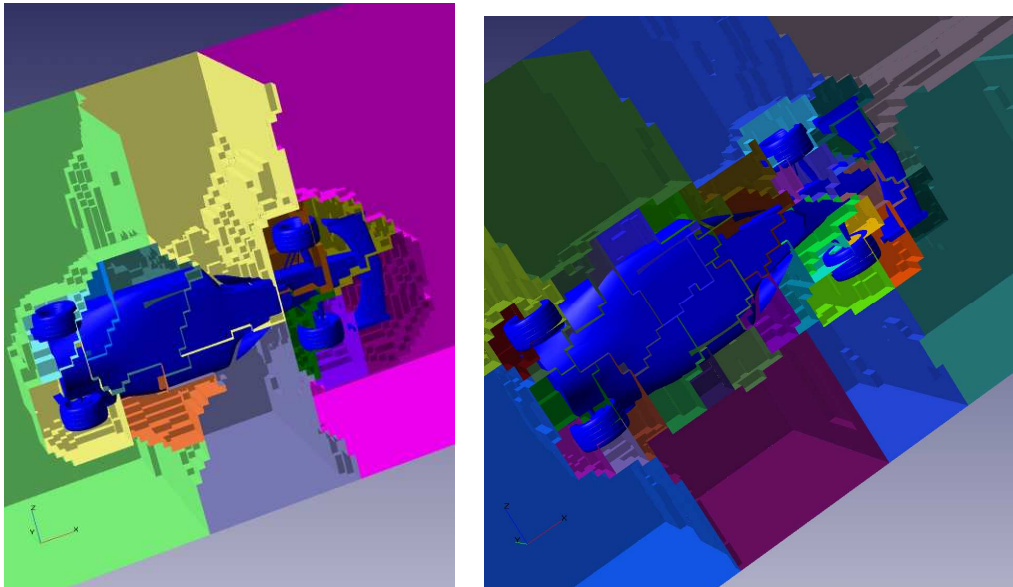


Рис. 3. Объемная декомпозиция расчетной сетки на 16 и 64 процессора

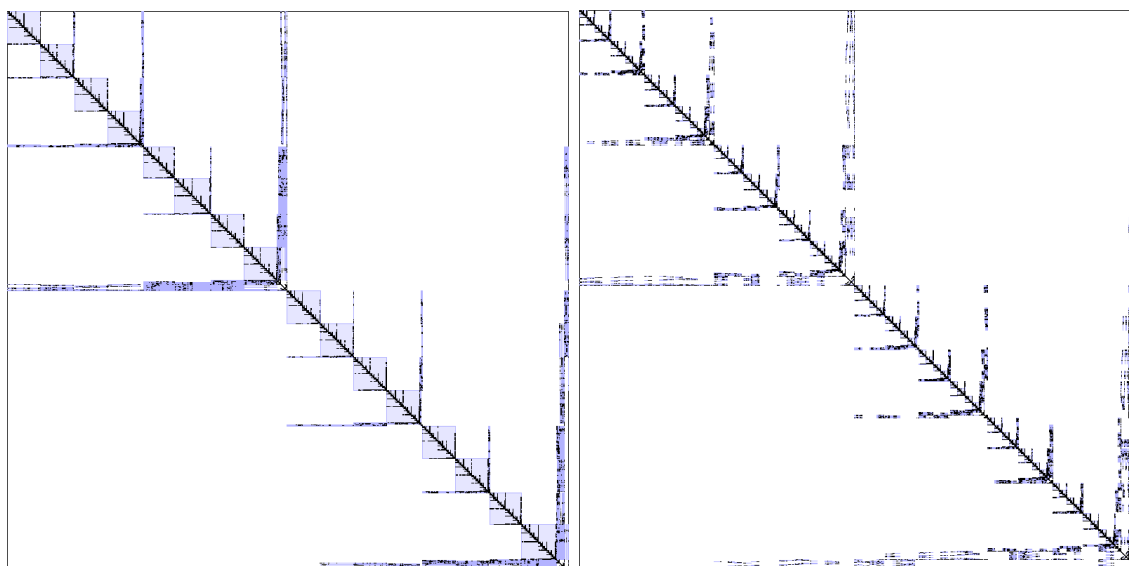


Рис. 4. Разреженность матрицы системы уравнений с учетом блочного биения для 16-ти и 64-ех процессоров/ядер

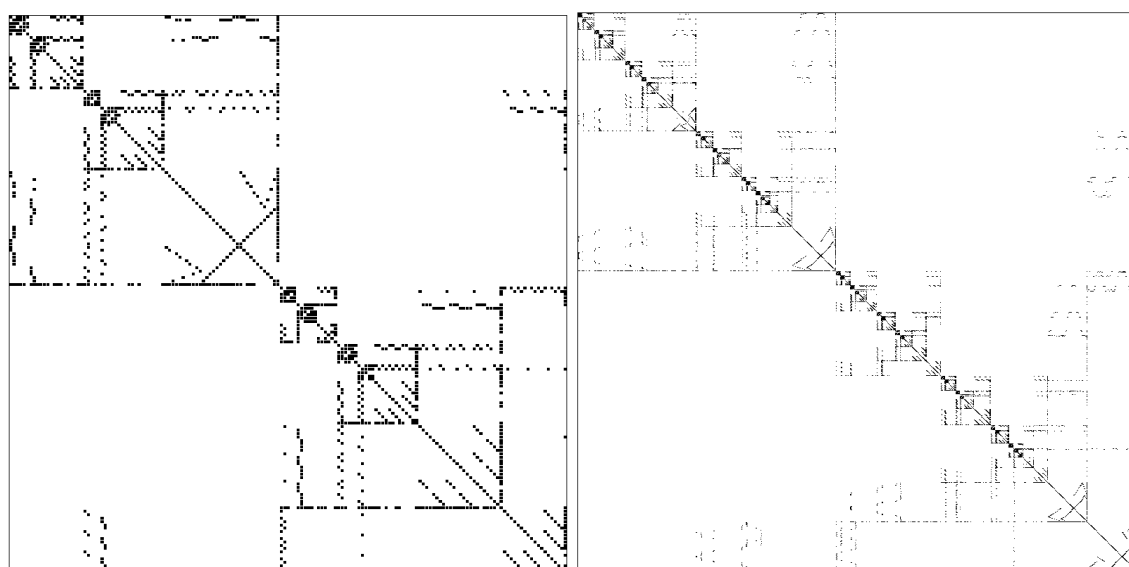


Рис. 5. Матрица зависимостей блочных вычислений для 16-ти и 64-ех процессоров/ядер

Таблица 1. Результаты численных экспериментов.

	2 узла по 4 ядра		4 узла по 2 ядра		4 узла по 4 ядра		4 узла по 8 ядер	
	MPI	MPI+TBB	MPI	MPI+TBB	MPI	MPI+TBB	MPI	MPI+TBB
Факторизация	14,4	9,32	9,71	9,16	7,63	5,01	---	3,98
Итерации	102,27	54,13	51,69	54,39	53,08	29,14	---	31,74

5. Заключение

В работе представлен комбинированный MPI+threads параллельный алгоритм решения системы линейных уравнений для компьютеров с неоднородным доступом памяти. На примере

системы линейных уравнений, возникающей во FlowVision при моделировании тестовой задачи вычислительной гидродинамики, показана его эффективность по сравнению с чисто MPI реализацией.

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. OpenMP Application Program Interface - 2.5 / OpenMP Architecture Review Board – 2005.
6. Intel Threading Building Blocks Tutorial – 1.6 / Intel Corp. 2007.
7. 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.
8. 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.
9. С.А.Харченко. "Влияние распараллеливания вычислений с поверхностными межпроцессорными границами на масштабируемость параллельного итерационного алгоритма решения систем линейных уравнений на примере уравнений вычислительной гидродинамики", Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2008), Санкт-Петербург, 28 января – 1 февраля 2008 г. Челябинск, Изд. ЮУрГУ, 2008, с.494-499.
10. Г.Б. Сушко, С.А. Харченко. "Многопоточная параллельная реализация итерационного алгоритма решения систем линейных уравнений с динамическим распределением нагрузки по нитям вычислений", Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2008), Санкт-Петербург, 28 января – 1 февраля 2008 г. Челябинск, Изд. ЮУрГУ, 2008, с.452-457.
11. А.А.Аксенов, А.А.Дядькин, В.А.Кутин, И.В.Москалёв, Г.Б.Сушко, С.А.Харченко. "Решение больших задач вычислительной гидродинамики на СКИФ МГУ с помощью FlowVision", Материалы Всероссийской научной конференции "Научный сервис в сети ИНТЕРНЕТ: решение больших задач", Новороссийск, 22-27 сентября 2008 г., Изд-во Московского Университета, 2008, с. 69-73.
12. <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview>