Параллельная реализация алгоритма разреженного QR разложения для прямоугольных верхних квази-треугольных матриц со структурой типа вложенных сечений

С.А. Харченко[1,2], А.А. Ющенко[1]

ООО "ТЕСИС"[1], Вычислительный Центр РАН[2]

В работе рассматривается параллельная MPI+threads+SIMD реализация алгоритма вычисления разреженного QR разложения специальным образом упорядоченной прямоугольной матрицы на основе разреженных блочных преобразований Хаусхолдера. В алгоритме производится предварительное независимое параллельное вычисление QR разложений для наборов строк матрицы. Затем в соответствии с деревом зависимостей производится вычисление QR разложения матриц, составленных из окаймлений R- факторов строчных разложений. Приводятся результаты экспериментов, подтверждающие эффективность предложенной параллельной реализации для тестовых задач. Алгоритм также может быть эффективно реализован на гетерогенных кластерных архитектурах с ускорителями типа GPGPU.

1. Введение

QR разложение прямоугольной матрицы является одним из базовых вычислительных алгоритмов для многих задач вычислительной математики. В частности, подобные вычисления возникают при решении СЛАУ, при решении задач наименьших квадратов и задач на собственные значения [1], и т.д. Возможность эффективно параллельным образом вычислять QR разложение разреженной матрицы в некоторых случаях означает возможность использования новых классов вычислительных алгоритмов, и поэтому подобные разработки представляют практический интерес.

В работе описывается реализация на гибридной MPI+threads+SIMD архитектуре представленного в работе [2] параллельного алгоритма вычисления разреженного QR разложения для многоуровневой верхней квази- треугольной разреженной матрицы со структурой типа вложенных сечений. Алгоритм в работе [2] во многом аналогичен представленному в работах [3] и [4] Тима Дэвиса с соавторами мульти- фронтальному алгоритму построения разреженного QR разложения. Основные отличия состоят в том, что:

- используются блочные преобразования Хаусхолдера [5];
- профильное разреженное QR разложение заменено на расширенное профильное разреженное QR разложение, которое во многих практически важных случаях дает заметно меньшее заполнение Q- фактора;
- введено дополнительное строчное упорядочивание и биение, которое позволяет дополнительно уменьшить связность вычислений и заполнение Q факторов;
- предложен алгоритм построения представления матрицы, удобного для параллельного вычисления разреженного QR разложения, на основе декомпозиции расчетной области.

Данная работа, также как и работа [2], является базовой для планируемой серии работ по новым параллельным итерационным алгоритмам решения СЛАУ и задач наименьших квадратов на основе композиции подпространств, порождаемых разреженными базисами. Параллельная реализация, представленная в работе, может быть взята за основу при реализации алгоритма вычисления разреженного QR разложения на гетерогенных кластерных архитектурах с ускорителями типа GPGPU.

Работа построена следующим образом. В Разделе 2 приводится краткое описание параллельного алгоритма из работы [2] для построения разреженного QR разложения прямоугольной многоуровневой верхней квази- треугольной матрицы типа вложенных сечений. В Разделе 3 описывается гетерогенная MPI+threads+SIMD архитектура, для которой указанный параллельный алгоритм был реализован. В Разделе 4 описываются подробности реализации при отобра-

жении алгоритма на параллельную архитектуру компьютера. В Разделе 5 приводится описание тестовой задачи и представлены результаты численных экспериментов.

2. Параллельный алгоритм построения разреженного QR разложения

В этом разделе приводится краткое описание параллельного алгоритма из работы [2] для построения разреженного QR разложения прямоугольной матрицы.

Последовательный алгоритм построения QR разложения основан на блочном преобразовании Хаусхолдера вида

$$\Omega = I_M + F\,T\,F^T, \tag{2.1}$$

где $\Omega \in \mathbb{R}^{M\times M}$, $\Omega^T\Omega = I_M$, $F \in \mathbb{R}^{M\times s}$, $T \in \mathbb{R}^{s\times s}$. Блочное преобразование (2.1) строится через известный набор из s обычных векторных преобразований Хаусхолдера следующим образом: матрица F составляется из набора векторов направлений векторных преобразований Хаусхолдера, а верхняя треугольная матрица T может вычислена рекуррентным образом с использованием коэффициентов векторных преобразований Хаусхолдера если известна матрица $\Psi = F^TF$.

При обсуждении разреженных вычислений будут рассматриваться вопросы вычисления QR разложения для прямоугольных так называемых мелко блочно разреженных матриц. Это означает, что разреженность понимается в смысле блоков малого размера, каждый из которых является плотной в общем случае прямоугольной матрицей. Для простоты будем предполагать, что все мелкие блоки квадратные малого размера s. При этом все алгоритмы могут быть обобщены на случай переменного столбцевого и строчного мелко блочного биения. В противовес мелким плотным $s \times s$ блокам будем говорить также о крупно- блочном или просто блочном биении, строчном и столбцевом. Это будет означать, что соответствующие подматрицы составлены из некоторого количества мелко блочных строк и столбцов. При этом под блочным преобразованием Хаусхолдера имеется в виду преобразование вида (2.1) для одного мелко блочного столбца.

При последовательном построении профильного разреженного QR разложения мелко блочной разреженной матрицы C действуем по аналогии со случаем плотной матрицы. По первому мелко блочному столбцу матрицы строим разреженное блочное преобразование Хаусхолдера с разреженностью столбца такой, чтобы обнулить мелкие блоки матрицы под первой блочной диагональю. Применяем транспонированное блочное преобразование Хаусхолдера ко второму мелко блочному столбцу, для полученного результата строим следующее разреженное блочное преобразование Хаусхолдера для обнуления элементов под второй мелко блочной диагональю, и т.д.

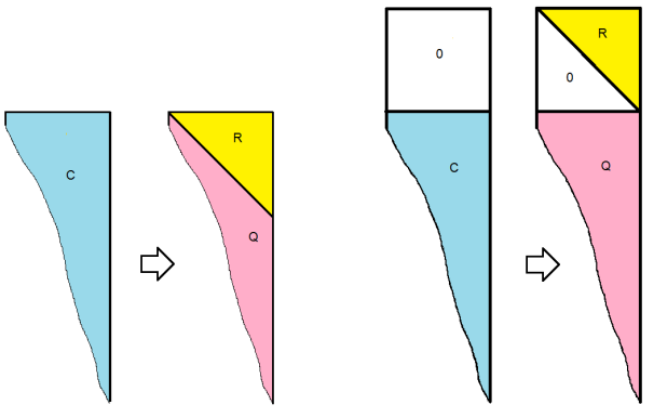

Рис. 1. Профильное (слева) и расширенное профильное (справа) QR разложения

Наравне с профильным разреженным QR разложением рассмотрим также расширенное профильное QR разложение матрицы. Схематически профильное и расширенное профильное QR изображены на Рисунке 1. Фактически расширенное профильное QR разложение - это профильное QR разложение, примененное к матрице, расширенной сверху нулевым квадратным блоком. При этом структура разреженности Q фактора пополняется возможными дополнительными элементами на месте бывшего фактора R, и отсоединенными диагональными элементами, примыкающими к новому R фактору.

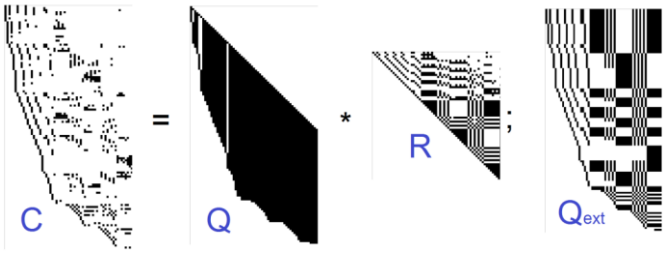

Рис. 2. Профильное (слева) и Q фактор расширенного профильного (справа) QR

В работе [2] показывается математическая эквивалентность профильного и расширенного профильного QR разложений в случае матрицы C полного столбцевого ранга. Также приводятся примеры, когда заполнение Q- фактора QR разложения в расширенном профильном QR разложении значительно больше заполнения Q- фактора в профильном за счет дополнительного заполнения в бывшем R- факторе разложения, а также обратный пример, представленный на Рисунке 2. В интересующих авторов приложениях основным является случай, когда число столбцов много меньше числа строк. В этих случаях более предпочтительным является использование варианта с расширенным профильным разреженным QR разложением. Кроме того, использование расширенного профильного QR удобно при проведении вычислений с мелко блочными матрицами, в которых число строк меньше числа столбцов.

Для прямоугольной разреженной матрицы $C \in \mathbb{R}^{M\times Ns}$ введем ее строчное биение в виде:

$$C = \begin{bmatrix} C_1 \\ \vdots \\ C_k \end{bmatrix}, \tag{2.2}$$

где $C_t \in \mathbb{R}^{M_t\times Ns}$, $t = 1,\cdots,k$, и $\sum_{t=1}^{k} M_t = M$. Пусть для каждой из матриц имеет место QR разложение с блочными преобразованиями Хаусхолдера:

$$C_t = \left(\prod_{j=1}^{N} \Omega_j{}^{(t)}\right)\begin{bmatrix} R^{(t)} \\ 0 \end{bmatrix}, \tag{2.3}$$

$t = 1,\cdots,k$. Для разреженной матрицы C ее строчные блоки C_t могут содержать много нулевых мелко блочных столбцов, в подобных случаях матрицы $R^{(t)}$ в (2.3) не обязательно верхние треугольные и имеют много нулевых столбцов, а среди блочных преобразований Хаусхолдера имеется много тождественных преобразований с единичной матрицей [2].

Рассмотрим задачу построения QR разложения всей матрицы из (2.2) на основе разложений (2.3). Для этого рассмотрим разреженную матрицу

$$\hat{C} = \begin{bmatrix} R^{(1)} \\ \vdots \\ R^{(k)} \end{bmatrix} \tag{2.4}$$

и ее QR разложение

$$\hat{C} = \left(\prod_{j=1}^{N} \hat{\Omega}_j\right)\begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}. \tag{2.5}$$

Обозначим $\hat{\Phi} = \left(\prod_{j=1}^{N} \hat{\Omega}_j\right)$. Имеют место соотношения

$$C = \begin{bmatrix} C_1 \\ \vdots \\ C_k \end{bmatrix} = \begin{bmatrix} \Phi_1 R^{(1)} \\ \vdots \\ \Phi_k R^{(k)} \end{bmatrix} = \begin{bmatrix} \Phi_1 & & 0 \\ & \ddots & \\ 0 & & \Phi_k \end{bmatrix}\hat{\Phi}\begin{bmatrix} \hat{R} \\ 0 \end{bmatrix} = \Phi\begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}, \tag{2.6}$$

где

$$\Phi = \begin{bmatrix} \Phi_1 & & 0 \\ & \ddots & \\ 0 & & \Phi_k \end{bmatrix}\hat{\Phi}, \tag{2.7}$$

причем $\Phi^T\Phi = I_M$. Отсюда следует, что неявное представление (2.7) совместно с равенством $C = \Phi\begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}$ из (2.6) есть QR разложение матрицы C, при этом матрица Φ представляет собой Q-фактор QR разложения, а квадратная верхняя треугольная матрица $\hat{R}$ есть R- фактор QR разложения.

Описанная конструкция очевидно позволяет параллельным образом вычислять QR разложение матрицы за счет введения строчного биения, поскольку строчные QR разложения (2.3) можно считать независимо. Синхронизация вычислений происходит только при вычислении

объединяющего QR разложения (2.5). С другой стороны понятно, что подобный подход к основному распараллеливанию вычислений может быть эффективен только если число столбцов в матрице существенно меньше числа строк, иначе затраты на объединяющее QR разложение могут доминировать в вычислениях.

Описанный параллельный алгоритм вычисления QR разложения по блочным строкам можно сделать эффективным за счет использования дополнительной столбцевой разреженности матрицы. Для этого рассмотрим двух- уровневую организацию вычислений для прямоугольной матрицы C со структурой разреженности, изображенной на Рисунке 3. Пусть число мелко блочных столбцов в матрицах C_1, C_2 и C_S равны соответственно N_1, N_2 и N_S соответственно, $N_1 + N_2 + N_S = N$. Как показано в [2], для матрицы C с такой структурой разреженности для соответствующей матрицы $\hat{C}$ типа (2.4) в объединяющем QR разложении задачу вычисления ее QR разложения можно перестановкой блочных строк свести к задаче вычисления QR разложения с разреженной матрицей мелко блочного размера $(n_1 + n_2 + N_S) \times N_S$, здесь n_1 и n_2 соответственно число ненулевых мелко блочных столбцов в матрицах c_1 и c_2. Обобщая этот подход на общий случай введем следующее определение [2].

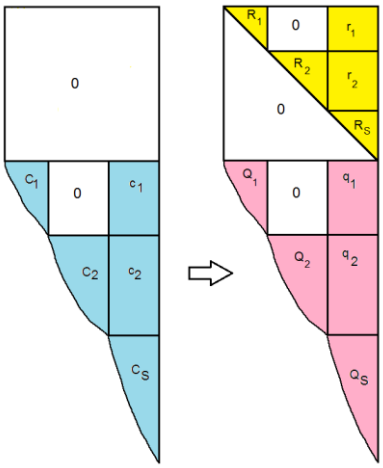

Рис. 3. Двух- уровневая организация параллельного вычисления QR

Определение 1. Прямоугольную мелко блочную матрицу с введенным на ней блочным строчным и столбцевым биениями будем называть *верхней квази- треугольной L -уровневой матрицей со структурой разреженности типа вложенных сечений*, если в терминах крупных блоков матрица является квадратной верхней треугольной и имеет структуру разреженности типа вложенных сечений, описываемой L -уровневым деревом зависимостей вычислений.

В частности, матрица на Рисунке 3 в терминах Определения 1 является двух уровневой верхней квази- треугольной с двух уровневым бинарным деревом зависимостей вычислений.

Как следует из предыдущего изложения, для эффективного вычисления QR разложения верхних квази- треугольных матриц с разреженностью типа вложенных сечений можно использовать следующий параллельный алгоритм:

1. Параллельно и независимо для каждой блочной строки матрицы строим расширенное профильное разреженное QR разложение на основе блочных преобразований Хаусхолдера;
2. Параллельно в порядке, определяемом деревом зависимостей вычислений, достраиваем QR разложения для объединяющих подматриц для вычисления QR разложений соответствующих мелко блочных столбцевых окаймлений.

3. Архитектура гибридной вычислительной системы

Большинство современных суперкомпьютерных вычислительных систем как правило имеют неоднородную архитектуру. С одной стороны, имеется набор вычислительных узлов с распределенной памятью, обмен данными между которыми может быть осуществлен по быстрой обменной сети. С другой стороны, каждый узел представляет собой многопроцессорный/многоядерный компьютер с общим доступом к оперативной памяти. Программная реализация вычислительных алгоритмов (включая алгоритм вычисления разреженного QR разложения) на компьютерах подобной архитектуры предполагает использование стандарта MPI при распараллеливании по распределенной памяти (по узлам вычислений), а по общей памяти узла распараллеливание по процессорам/ядрам естественно осуществлять на основе стандартов работы с потоками с встроенными механизмами динамической балансировки нагрузки, таких как OpenMP или Intel® Threading Building Blocks (TBB). При этом предполагается (Рисунок 4), что

на каждом узле имеется только один MPI процесс, который порождает на этом узле нужное количество одновременно работающих потоков вычислений.

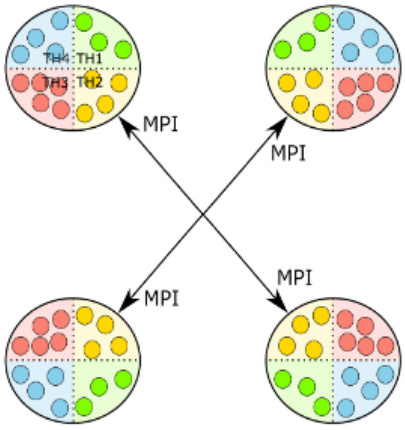

Рис. 4. Двухуровневая MPI+TBB организация параллельных вычислений

Большинство современных процессоров для оптимизации времени выполнения кода поддерживают так называемые векторные расширения систем команд - SIMD (Single Instruction Multiple Data) инструкции. Подобные вычисления можно проводить на любом ядре центрального процессора. В этих расширениях вычисления осуществляются не со стандартными данными целого и вещественного типа, а сразу с группой, вектором таких данных фиксированной длины. Любое вычисление на векторных регистрах осуществляется в следующей последовательности: данные из памяти загружаются в регистровые переменные, производится вызов аппаратно поддерживаемой функции работы с регистрами, затем данные обратно выгружаются в обычную память. Выпускаемые сейчас процессоры обычно поддерживают системы команд SSE и AVX, работающие соответственно со 128- XMM и 256- битными YMM регистрами. Это позволяет при использовании 256- битных регистров YMM, например, за один такт сложить 8 чисел с плавающей точкой одинарной точности или перемножить 4 числа с двойной точностью. На некоторых новейших процессорах поддерживается система команд AVX2, в этой системе команд дополнительно по отношению к AVX имеются FMA команды, совмещенного сложение+умножение векторов. В ускорителях Intel Xeon Phi имеется поддержка 512- битных ZMM регистров. В следующем поколении ускорителей Knights Landing появится аппаратная поддержка новой системы команд AVX512, совместимой с серверными процессорами Xeon.

4. Отображение алгоритма разреженного QR разложения на архитектуру вычислительной системы

Приведенный во втором разделе параллельный алгоритм вычисления разреженного QR разложения для верхней квази- треугольной матрицы типа вложенных сечений был реализован на кластерной MPI+threads архитектуре с использованием SIMD инструкций. Распараллеливание алгоритма на гетерогенной MPI+threads+SIMD архитектуре осуществлено следующим образом.

Распараллеливание верхнего уровня по MPI осуществлялось как распараллеливание по распределенной памяти. Для этих целей в дереве зависимостей вычислений каждому MPI процессу было выделено целиком под- дерево зависимых вычислений по возможности с близкой для всех поддеревьев вычислительной работой. Дополнительная динамическая балансировка вычислений в какой-либо форме не проводилась. Обработка каждого из вышестоящих узлов дерева зависимостей передавалась одному (например первому) из тех процессоров, который обрабатывал один из узлов сыновей данного узла. На каждый MPI процесс перераспределялись те блочные строки матрицы, которые нужны для окончательной обработки своих узлов поддеревьев зависимостей вычислений.

Распараллеливание среднего уровня по нитям осуществлялось с помощью технологии Intel TBB либо как независимые, либо как зависимые вычисления. Зависимости описываются в виде под- графа зависимых вычислений для узлов поддеревьев своего MPI процессора, независимые вычисления проводились при начальном вычислении QR разложений для блочных строк. При проведении вычислений с узлом дерева зависимых вычислений, не входящим в MPI- поддеревья, вычисление объединяющих QR разложений проводилось только при поступлении необходимых данных с других MPI процессов.

Распараллеливание нижнего уровня параллельных вычислений - SIMD векторизация - проводилось за счет использования блочных преобразований Хаусхолдера. Рассмотрим этот вопрос подробнее.

Основными операциями при работе с блочными преобразованиями Хаусхолдера являются:

1. Вычисление векторного QR разложения с векторными преобразованиями Хаусхолдера для мелко блочного столбца;

2. Преобразование набора векторных преобразований Хаусхолдера в единое блочное преобразование Хаусхолдера для мелко блочного столбца;

3. Применение с учетом разреженности блочных преобразований Хаусхолдера к последующим мелко блочным столбцам матрицы.

С учетом сказанного в Разделе 2 про способ трансформации векторных преобразований Хаусхолдера в блочное можно выделить следующие 4 основные вычислительные операции:

- векторный dot: $z = x^T y$ - скалярное произведение векторов;
- векторный axpy: $y += ax$;
- блочный dot: $Z = X^T Y$ - блочное обобщение скалярного произведения для блоков ;
- блочный axpy: $Y += X * A$.

Векторные операции встречаются при векторном вычислении QR разложения. В операциях dot и axpy длины векторов недостаточны для покрытия накладных расходов вызова оптимизированных BLAS функций из MKL. По этой причине в этих операциях осуществлялась ручная SIMD векторизация прямым вызовом соответствующих векторных инструкций с помощью интринсик функций.

Для максимальной локализации работы с памятью при обработке блочных преобразований Хаусхолдера естественно использовать формат хранения данных "по строкам" вместо традиционно используемого для блока векторов формата "по столбцам".

При проведении SIMD векторизации для блоков векторов в силу особенностей векторных инструкций реализовывалась поддержка только значений $s = 2; 4; 8; 16$ для двух типов данных float и double. Как уже отмечалось, разреженные QR разложения будут использоваться в контексте построения разреженных базисов в алгоритмах решения СЛАУ и задач наименьших квадратов, а в этом случае параметр s - это выбранный размер блока итерационной схемы. Для длинных блоков векторов задача вычисления блочного dot и блочного axpy сводилась к циклу вызовов для подматриц размера $s \times s$. Подробности реализации операций блочного dot и axpy в терминах SIMD инструкций для подматриц стандартного размера можно найти в работе [6]. Как показали численные эксперименты, ручная векторизация для таких маленьких размеров блока s оказалась значительно эффективней вызовов библиотечных реализаций из MKL и IPP.

5. Результаты численных экспериментов

Для тестирования предложенных в работе алгоритмов был выбран искусственный тест, в котором по возможности отражены основные особенности будущего их использования.

Для регулярной $N_x \times N_y \times N_z$ прямоугольной сетки была построена регулярная разреженная мелко блочная матрица с размерами блоков s, $s = 8$, структура которой отвечает шаблону уравнения Лапласа. Выбор размера блока был обусловлен тем, что для такого размера возможно для данных типов float и double продемонстрировать возможности SIMD векторных инструкций вплоть до набора AVX2.

Для полученной матрицы с помощью алгоритма вложенных сечений, как описано в работе [2], было построено упорядочивания и биения, приводящие матрицу к виду L- уровневой верхней квази- треугольной матрицы типа вложенных сечений. Обозначим эту мелко блочную матрицу A. Для этой матрицы был построен блочно разреженный ортонормированный базис со столбцевым размером блока s, совпадающим с размером блока матрицы, разреженность каждого блока базиса не выходит за блочный столбцевой размер матрицы. Обозначим матрицу ортонормированного базиса P. Матрица P представляет собой блочно-диагональную матрицу вида

$$P = \begin{bmatrix} P_1 & & 0 \\ & \ddots & \\ 0 & & P_l \end{bmatrix},$$

где l - число столбцевых блоков в матрице A. Очевидно, что матрица C такая, что

$$C = A * P ,$$

также как и A, представляет собой мелко блочную с размером блока s -уровневую верхнюю квази- треугольную матрицу типа вложенных сечений, и число столбцов в ней есть полное число столбцов в базисе P.

Эксперименты по вычислению разреженного QR разложения проводились для построенной таким образом матрицы C. В частности, если число векторов в базисе P много меньше мелко- блочного размера матрицы A, то в основных подматрицах число столбцов много меньше числа строк, что оправдывает использование расширенного разреженного QR разложения в основных вычислениях. Кроме того, каждый блок P_j базиса обладает разреженностью, а потому результат произведения также есть разреженная матрица.

Для тестирования алгоритма была сгенерирована матрица с числом строк равным порядка 1,5 млн. и размером мелкого блока $s = 8$ и количеством столбцевых блоков на два порядка меньше количества строк в матрице. Тестирование алгоритма проводилось на 18-ядерном процессоре Intel Xeon E5-2699v3 с архитектурой Haswell под управлением CentOS 6.6. Тестовое приложение компилировалось с помощью оптимизирующего компилятора ICC-15.0.3. В таблицах 1 и 2 представлены времена работы алгоритма для матрицы с одинарной и двойной точностью соответственно для различных наборов векторных инструкций и количества потоков.

Таблица 1 Времена работы алгоритма с числами одинарной точности

arch\threads	1	2	4	8	12	16	18
no-vec	2,197	1,111	0,609	0,356	0,254	0,197	0,178
SSE	1,122	0,570	0,312	0,184	0,131	0,104	0,092
AVX	0,887	0,453	0,245	0,143	0,103	0,082	0,076
AVX2	0,711	0,367	0,196	0,119	0,086	0,068	0,062

Таблица 2 Времена работы алгоритма с числами двойной точности

arch\threads	1	2	4	8	12	16	18
no-vec	2,447	1,235	0,677	0,386	0,276	0,216	0,198
SSE	1,985	1,01	0,555	0,316	0,224	0,176	0,162
AVX	1,466	0,743	0,399	0,231	0,165	0,134	0,12
AVX2	1,043	0,528	0,282	0,166	0,12	0,0962	0,0904

Из результатов видно, что использование самых современных векторных инструкций позволяет получить ускорение до 3 раз по сравнению с оптимизирующим компилятором ICC. Ускорение по сравнению с бесплатным компилятором GCC получается еще более значительным. Использование всех 18 ядер процессора ускоряет работу алгоритма в обоих случаях в более чем 11 раз. На Рисунке 5 изображен профиль загрузки потоков в тестовом приложении, полученном с помощью программы Intel VTune Amplifier, оранжевым цветом отмечены регионы синхронизации потоков. Первая оранжевая область на временной линии отвечает окончанию обработки

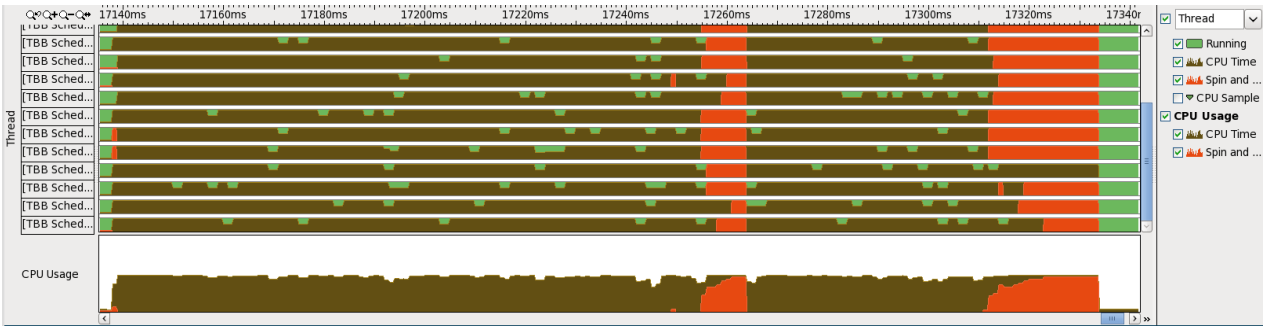

Рис. 5. Профиль загрузки потоков в тестовом приложении.

независимых блоков, а вторая – окончанию обработки зависимых блоков по дереву. Видно, что при обработке зависимых блоков вычислений все еще остается значительный дисбаланс загрузки ядер, что не позволяет вплотную приблизиться к линейной масштабируемости. В даль-

нейшем планируется уделить больше внимания этим местам в алгоритме. Таким образом, использование обоих механизмов распараллеливания в современных процессорах позволяет достичь ускорения в более чем 30 раз.

Заключение

В работе представлена реализация на гибридной параллельной MPI+threads+SIMD архитектуре параллельного алгоритма вычисления QR разложения многоуровневой разреженной верхней квази- треугольной матрицы со структурой разреженности типа вложенных сечений. Результаты численных экспериментов с предложенным алгоритмом для тестовых задач на гибридной параллельной MPI+threads+SIMD архитектуре показывают высокую эффективность предложенных алгоритмов: ускорение до 3 раз от использования векторных инструкций AVX2, ускорение до 11 раз при использовании 18 ядер процессора.

Литература

1. Тыртышников Е.Е. Методы численного анализа : учеб. пособие для студ. вузов / — М. : Издательский центр «Академия», 2007. — 320 с. — (Университетский учебник. Сер. Прикладная математика и информатика), ISBN 978-5-7695-3925-1.
2. Харченко С.А. Параллельный алгоритм разреженного QR разложения для прямоугольных верхних квази- треугольных матриц со структурой типа вложенных сечений // Представлена в качестве доклада на первую объединенную международную конференцию "Суперкомпьютерные дни в России", Москва, 28-29 сентября 2015 г.
3. Davis T. A. 2011. Algorithm 915: SuiteSparseQR, a multifrontal multithreaded sparse QR factorization package. ACM Trans. Math. Softw. 38, 1 (2011), 8:1–8:22.
4. Yeralan S.N., Davis T.A., Ranka S. Algorithm 9xx: Sparse QR Factorization on the GPU. ACM Transactions on Mathematical Software, Vol. 1, No. 1, Article 1, Publication date: January 2015.
5. Haidar A., Dong T., Tomov S., Luszczek P., Dongarra J. Framework for Batched and GPU-resident Factorization Algorithms Applied to Block Householder Transformations.
6. Применение векторных инструкций в алгоритмах блочных операций линейной алгебры / Андреев А.Е., Егунов В.А., Насонов А.А., Новокщенов А.А. // Известия ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах". Вып. 21 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2014. - № 12 (139). - С. 5-11.

Parallel implementation of the sparse QR decomposition for rectangular upper quasi triangular matrix with ND-type sparsity

Sergey Kharchenko and Alexey Yushchenko

Keywords: sparse rectangular matrix, upper quasi triangular matrix, nested dissection, QR decomposition, Householder transformations, MPI, multithreading, SIMD

The paper considers parallel MPI+threads+SIMD implementation of the algorithm for computing sparse QR decomposition of a specially ordered rectangular matrix. Decomposition is based on block sparse Householder transformations. The algorithm starts with independent parallel QR decompositions for sets of matrix rows; and then, according to the computations tree, the QR decomposition is performed for matrices, combined with elements of R factors of rows decompositions. The results of numerical experiments for test problems show efficiency of the parallel implementation. The algorithm can also be efficiently implemented on heterogeneous cluster architectures with GPGPU accelerators.