DOI: 10.17117/na.2016.10.03.146

<http://ucom.ru/doc/na.2016.10.03.146.pdf>

Поступила (Received): 19.10.2016

Егунов В.А., Старостина А.В. **Реализация преобразования отражения Хаусхолдера** **на параллельных вычислительных системах** **с общей памятью**

Egunov V.A., Starostina A.V.
Implementation of Householder reflection transformations
on shared memory parallel computing systems

В статье рассматривается реализация преобразования отражения Хаусхолдера на параллельных вычислительных системах с общей памятью. Приводятся алгоритмы левостороннего и правостороннего умножения на матрицу отражения, оценивается использование кэш-памяти в соответствующих параллельных программах, проводится оценка их эффективности

Ключевые слова: общая память, преобразование отражения, эффективность программ

The article discusses the implementation of Householder reflection transformations on parallel computing systems with shared memory, algorithms left-hand and right-hand multiplication on the reflection matrix. Evaluated the using of cache in the respective parallel programs, an evaluation of their efficiency

Key words: shared memory, reflection transformation, program efficiency

Егунов Виталий Алексеевич

*Кандидат технических наук, доцент
Волгоградский государственный технический университет
г. Волгоград, пр. Ленина, 28*

Egunov Vitaly Alekseevich

*Candidate of Technical Sciences, Associate Professor
Volgograd state technical university
Volgograd, Lenina ave., 28*

Старостина Алина Владиславовна

*Студент
Волгоградский государственный технический университет
г. Волгоград, пр. Ленина, 28*

Starostina Alina Vladislavovna

*Student
Volgograd state technical university
Volgograd, Lenina ave., 28*

Преобразование отражения Хаусхолдера является одним из базовых преобразований линейной алгебры, используемых при выполнении ряда матричных преобразований, например, QR – разложения, которое может использоваться для поиска собственного и сингулярного разложений, решения линейных систем и ряда других задач. Поиск QR – разложения является трудоемкой задачей, поэтому естественным является ее распараллеливание и решение на параллельных вычислительных системах. В данной работе рассматривается выполнение преобразования отражения на параллельных вычислительных систе-

мах с общей памятью. Выбор подобного класса параллельных систем обусловлен их широкой распространенностью. Выбор объекта исследований обусловлен активным применением данных матричных преобразований при решении широкого класса прикладных задач [1, 2].

Особенностью систем с общей памятью является снижение общей эффективности программ при интенсивном обращении вычислительных узлов (например, ядер в многоядерном процессоре) к памяти. В связи с этим, одной из главных задач при разработке подобных программ, является минимизация обращений к памяти с одновременным увеличением роли кэш – памяти. Сосредоточив данные в кэш – памяти отдельных вычислительных узлов, можно снизить общую интенсивность запросов по общей шине. При разработке алгоритмов для данных программ, большое внимание уделяется увеличению доли обращений к смежным ячейкам памяти в общем потоке обращений, что приводит к уменьшению числа кэш – промахов, и, как следствие, уменьшению времени выполнения программ.

При использовании традиционной схемы выполнения QR – разложения исходной матрицы с использованием преобразования отражения Хаусхолдера, осуществляется умножение на матрицу отражения справа с постолбцовой обработкой исходной матрицы. На k-ом этапе преобразования осуществляется следующее преобразование (1).

$$\begin{aligned} s_k &= \text{sign}(a_{kk}) \left(\sum_{i=k}^n a_{ik}^2 \right)^{\frac{1}{2}}, \varphi_k = (s_k^2 - s_k a_{kk})^{-1} \\ \mathbf{u}_k^T &= (0, \dots, 0, a_{kk} - s_k, a_{k+1,k}, \dots, a_{nk}) \\ a_{kk} &= s_k \end{aligned} \quad (1)$$

Для j = k+1 до n

$$\begin{aligned} \lambda_j &= \varphi_k \mathbf{u}_k^T \mathbf{a}_j \\ \mathbf{a}_j &= \mathbf{a}_j - \lambda_j \mathbf{u}_k \end{aligned}$$

Здесь k – номер этапа преобразования. Данная вычислительная схема подробно описана в [3]. На параллельных вычислительных системах с общей памятью можно выделить следующие особенности (предполагается, что матрицы хранятся по строкам):

- элементы вектора отражения вычисляются по элементам столбцов исходной матрицы, что приводит к большому числу кэш-промахов;
- циклы вычисления скалярных произведений и элементов матрицы являются ориентированными по столбцам исходной матрицы, что также приводит к неэффективному использованию кэш-памяти.

Данный алгоритм оказывается неэффективным на системах с общей памятью, что подтверждается результатами вычислительных экспериментов. С учетом приведенных выше замечаний относительно распараллеливания данного алгоритма, распараллеливать стоит вычисления в рамках этапа преобразова-

ния, сами этапы стоит выполнять строго последовательно. Чаще всего предлагаются варианты параллельной реализации преобразования, в которых вычисление элементов вектора отражения и результирующей матрицы выполняются последовательно в силу наличия информационных зависимостей.

Исправить недостатки предыдущей вычислительной схемы можно, полностью переориентировавшись на построчную обработку матрицы. Этого можно добиться, если умножать на матрицу отражения слева. В результате будет получена матрица L – нижнетреугольная (LQ разложение). На каждом этапе выполняются следующие вычисления (2).

Для $j = k+1$ до n

$$\begin{aligned}\lambda_j &= \varphi_k \mathbf{a}_j^T \mathbf{u}_k \\ \mathbf{a}_j^T &= \mathbf{a}_j^T - \lambda_j \mathbf{u}_k^T\end{aligned}\quad (2)$$

Реализация данного алгоритма напоминает реализацию предыдущего алгоритма за исключением:

- обработка матрицы ведется полностью построчно;
- элементы вектора отражения $\mathbf{u}$ определяются элементами строк матрицы (в первом алгоритме они определялись элементами столбцов).

Данный алгоритм является более эффективным для систем с общей памятью. Так как данная вычислительная схема напоминает вычислительную схему первого алгоритма, ее также можно эффективно распараллелить, совместив вычисление результирующих строк матрицы с вычислением элементов вектора отражения. Эффективность данного алгоритма обеспечивается оптимальным использованием кэш – памяти, т.к. построчная обработка матриц приводит к генерации гораздо меньшего числа промахов.

Список используемых источников:

1. Андреев А.Е., Егунов В.А., Жариков Д.Н., Малолетков В.А. Реализация вычислительно-интенсивных алгоритмов на гибридных системах с реконфигурируемыми сопроцессорами: монография. Волгоград, 2013. 179 с.
2. Егунов В.А., Лукьянов В.С. Аппаратные методы решения задач линейной алгебры: монография. Волгоград: Политехник, 2007. 152 с.
3. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. Мир, 1991. 367 с.

© 2016, Егунов В.А., Старостина А.В.

Реализация преобразования отражения Хаусхолдера на параллельных вычислительных системах с общей памятью

© 2016, Egunov V.A., Starostina A.V.

Implementation of Householder reflection transformations on shared memory parallel computing systems