DOI: 10.17117/na.2015.11.03.135

<http://ucom.ru/doc/na.2015.11.03.135.pdf>

Поступила (Received): 13.11.2015

Егунов В.А., Шумейко О.О.

Параллельные алгоритмы решения задач линейной алгебры на неоднородных вычислительных системах

Egunov V.A., Shumeiko O.O.

Parallel algorithms for solving linear algebra problems on heterogeneous computing systems

The article discusses the development of parallel algorithms for solving problems of linear algebra at the example of performing QR decomposition using the reflection method

The article discusses the development of parallel algorithms for solving problems of linear algebra at the example of performing QR decomposition using the reflection method

Ключевые слова: QR – разложение, параллельные алгоритмы, неоднородные вычислительные системы

Key words: QR decomposition, parallel algorithms, heterogeneous computing system

Егунов Виталий Алексеевич

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

Egunov Vitaly Alekseevich

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

Шумейко Олеся Олеговна

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

Shumeiko Olesya Olegovna

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

Задачи линейной алгебры играют огромную роль в современных научных исследованиях и при решении большого числа прикладных проблем. Это связано с тем, что решение целого ряда проблем сводится к решению задач линейной алгебры. Причем часто именно на матричные преобразования, такие как решение систем линейных алгебраических уравнений, поиск собственных значений и системы собственных векторов, поиск сингулярного разложения, обращение матриц и т.д., приходится основная часть операционной сложности решения. В связи с этим в последние десятилетия ведутся интенсивные исследования в области разработки эффективного математического обеспечения и развития вычислительных возможностей оборудования, предназначенных для решения указанных задач.

Наиболее часто используемыми задачами линейной алгебры в области ЦОС, обработки изображений и т.д., являются решение систем линейных алгебраических уравнений, поиск собственных значений и системы собственных векторов и поиск сингулярного разложения. Для создания высокоэффективных вычислительных средств, предназначенных для решения этих задач, необходимо выработать единый унифицированный подход, позволяющий решать любую из вышеперечисленных задач. Выработка такого подхода позволит создать унифицированные вычислительные средства, которые могут быть использованы для решения любой задачи из заданного спектра.

Можно показать [1], что в качестве универсального матричного преобразования можно использовать QR – разложение. Кроме того факта, что QR – разложение позволяет решать все основные задачи линейной алгебры, необходимо отметить, что для решения проблемы собственных значений и поиска сингулярного разложения матриц общего вида QR – разложение является наиболее быстрым и удобным. Конечно, для решения систем линейных алгебраических уравнений существует ряд других более быстрых методов (например, метод Гаусса), которые, однако, не могут претендовать на универсальность и не могут быть использованы для решения других задач линейной алгебры.

Существует два распространенных подхода к вычислению QR – разложения – преобразования Хаусхолдера и преобразования Гивенса. Использование преобразование Хаусхолдера в общем случае приводит к меньшему числу итераций и снижает общую вычислительную сложность [2].

QR – разложение с использованием преобразования Хаусхолдера можно описать следующим образом:

- для выполнения преобразования необходимо выполнить несколько итераций, число которых зависит от размерности матрицы;

- на каждой итерации все столбцы матрицы подвергаются одному и тому же преобразованию (отражению относительно некоторой гиперплоскости).

В соответствии с этим в данном алгоритме можно выделить последовательную часть – выполнение итераций должно выполняться строго друг за другом и распараллеливанию не подлежит, и часть, которая может быть распараллелена – выполнение одной итерации преобразования. Код программы, реализующей данное преобразование, с использованием технологии OpenMP может выглядеть следующим образом (iter – число итераций, num_cols – число столбцов, обрабатываемых на данной итерации):

```
for (int i = 0; i < iter; i++)  
{  
    //Код итерации – выполняются последовательно  
    //Определяем вектор отражения  
    #pragma omp parallel for  
    for (int j = 0; j < num_cols; j++)  
    {  
        //Обрабатываем очередной столбец  
    }  
}
```

Для систем с общей памятью данный подход уже может значительно повысить эффективность программ. Наряду с этим, в алгоритме Хаусхолдера можно выделить дополнительные резервы для распараллеливания, и первый – векторизация операций. Преобразование Хаусхолдера заключается в последовательном применении векторных преобразований к столбцам исходных матриц, поэтому повысить общую производительность программы можно, используя векторные расширения инструкций процессора, таких как SSE, AVX.

Наконец, при проведении расчетов на вычислительных системах, оснащенных GPU, можно получить дополнительное ускорение за счет использования большого числа вычислительных ядер, которыми оснащены современные GPU. Прямое использование классического преобразования Хаусхолдера на GPU не даст ожидаемого эффекта, т.к. число базовых подзадач (равное числу столбцов матрицы) значительно меньше числа доступных вычислительных ядер, однако, использование ДЛП – реализации преобразования Хаусхолдера [3, 4] позволяет значительно увеличить число базовых подзадач. Для программирования вычислительных ядер можно использовать технологии CUDA или OpenCL.

Таким образом, на неоднородной вычислительной системе, содержащей многоядерный процессор (процессоры) и один или несколько GPU, реализация многомерного преобразования Хаусхолдера может использовать:

- технологию OpenMP для распараллеливания на ядрах процессоров общего назначения;
- SSE, AVX для векторизации операций;
- CUDA, OpenCL для конвейеризации итераций ДЛП – преобразования Хаусхолдера.

Для получения максимальной эффективности необходимо осуществить балансировку нагрузки CPU и GPU.

Список используемых источников:

1. Духнич Е.И., Егунов В.А., Деревенсков С.О. Универсальный матричный алгоритм, ориентированный на систолическую реализацию // Автоматиз. процессов упр. и обраб. информ. Вологда, 1998. С. 54-58.
2. Духнич Е.И., Егунов В.А. Алгоритм быстрого преобразования Хаусхолдера // Концептуальное проектирование в образовании, технике и технологии. Волгоград, 1999. С. 71-75.
3. Andreev A., Egunov V., Zharikov D., Shapovalov O., Artuh S., Doukhnitich E. Evaluation of hardware implementations of cordic-like algorithms in fpga using opencl kernels // Communications in Computer and Information Science. 2014. T. 466 CCIS. С. 228-242.
4. Egunov V.A., Artyukh S.V. FPGA as coprocessors in solving problems of linear algebra // Инновационные информационные технологии. 2014. № 2. С. 93-97.