Использование графических процессоров для поиска пар ортогональных диагональных латинских квадратов порядка 10*

И.В. Шутов[1], С.Е. Кочемазов[2], О.С. Заикин[2], И.И. Курочкин[1], Э.И. Ватутин[3]

Институт проблем передачи информации им. А.А. Харкевича Российской академии наук[1]
Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук[2]
Юго-Западный государственный университет[3]

Рассматривается задача поиска всех возможных ортогональных пар, содержащих заданный диагональный латинский квадрата порядка 10. Для решения этой задачи разработан алгоритм, основывающийся на методе Эйлера-Паркера. На первом этапе этого алгоритма для данного на вход диагонального латинского квадрата находятся все трансверсали. На втором этапе формируются все возможные множества непересекающихся трансверсалей. Каждому такому множеству трансверсалей соответствует ортогональная пара, в которой присутствует исходный квадрат. Алгоритм был адаптирован для запуска на графических процессорах на архитектуре CUDA. Проведенные эксперименты показали, что GPU-реализация значительно превосходит по скорости CPU-реализацию.

Ключевые слова: латинский квадрат, ортогональность, графический процессор.

1. Введение

Латинский квадрат (ЛК) порядка n – это квадратная таблица размеров $n \times n$, заполненная элементами множества $\{0, \dots, n-1\}$ таким образом, что в каждой строке и в каждом столбце таблицы каждый элемент из этого множества встречается в точности один раз [1]. Латинские квадраты и системы, построенные на их основе, применяются в многих прикладных областях: планировании экспериментов, криптографии и т.д. *Диагональный латинский квадрат* (ДЛК) порядка n – это латинский квадрат, в котором каждый элемент из множества $\{0, \dots, n-1\}$ встречается в точности один раз не только в каждой строке и каждом столбце, но и в главной и побочной диагоналях. Пара латинских квадратов одинакового порядка называется *ортогональной*, если различны все упорядоченные пары элементов (a, b), где a – элемент в некоторой ячейке первого латинского квадрата, а b – элемент в ячейке с тем же номером второго латинского квадрата. *Трансверсаль* – это множество из n различных элементов, такое что никакие два элемента из этого множества не находятся на одной и той же строке или в одном и том же столбце ЛК.

Первая пара ортогональных диагональных латинских квадратов порядка 10 была найдена в 1992 г. [2]. Конкретнее, были найдены три такие пары. В статье [3] описано, как еще несколько десятков таких пар были найдены в проекте добровольных распределенных вычислений SAT@home [4]. Важной открытой комбинаторной проблемой является следующая – определить, существует ли тройка попарно ортогональных ДЛК порядка 10. Данная проблема чрезвычайно сложна, поэтому представляет интерес также решение различного рода ослабленных ее вариантов. В [3] был исследован следующий ослабленный вариант – найти такую тройку ДЛК порядка 10, в которой условие ортогональности может выполняться только частично. В на-

* Работа была частично поддержана РФФИ (гранты № 14-07-00403-а, 15-07-07891-а, 16-07-00155-а, 15-29-07095-офи_м и 16-07-00659-а) и советом по грантам Президента РФ (стипендия № СП-1184.2015.5, гранты МК-9445.2016.8 и НШ-8081.2016.9). Работа выполнена в рамках государственного задания для Юго-Западного государственного университета на 2014–2017 гг., № НИР 2246.

стоящей статье рассматривается другой вариант ослабления: необходимо проверить, можно ли на основе заданного частичного заполнения первых нескольких ячеек ДЛК достроить хотя бы один ДЛК, который будет состоять в тройке попарно ортогональных ДЛК порядка 10. Для решения данной задачи нами был разработан алгоритм, основанный на методе Эйлера-Паркера [5]. Далее в статье описывается данный алгоритм, а также результаты сравнения его реализаций для центральных процессоров (далее CPU) и графических процессоров (далее GPU).

2. Алгоритм поиска всех возможных ортогональных пар, содержащих заданный латинский квадрат

Для поиска всех ортогональных пар, в которые входит заданный ДЛК порядка n, был выбран метод Эйлера-Паркера [5]. На первом этапе данного метода находится множество всех трансверсалей данного ДЛК. На втором этапе находятся все сочетания из n непересекающихся трансверсалей – на основе каждого из таких сочетаний можно эффективно построить ровно одну искомую ортогональную пару. Соответственно, если ни одного такого сочетания построить нельзя, то рассматриваемый ДЛК не входит ни в одну ортогональную пару. Нами был разработан алгоритм, основанный на этом методе. Алгоритм изначально был ориентирован именно на GPU, но часть оптимизаций дали ускорение и для CPU.

Опишем часть алгоритма, которая касается первого этапа. Для распределения работы между потоками был выбран следующий подход. Каждый ДЛК обрабатывается в пределах одного блока потоков, что позволяет предварительно загрузить его в разделяемую память. Каждый поток обрабатывает некоторое подмножество трансверсалей своего блока. При этом первые два элемента трансверсали $t_0,\ t_1$ задаются следующим образом:

$$t_0 = \left\lfloor \frac{tId}{n} \right\rfloor,\ t_1 = tId\ (mod\ \mathrm{n}),$$

где tId – индекс текущего потока (нумерация идет с единицы), а n – порядок заданного ДЛК (в формуле используется округление вниз). Далее в рамках каждого потока осуществлялся полный перебор по элементам $t_2 \ldots t_9$. При этом в пределах каждого блока на GPU создавалось 100 потоков (увеличение количества потоков не привело к ускорению), что позволило полностью покрыть все возможные комбинации первых двух элементов каждой трансверсали на этапе инициализации алгоритма. Схема выполнения первого этапа алгоритма показана на рис. 1.

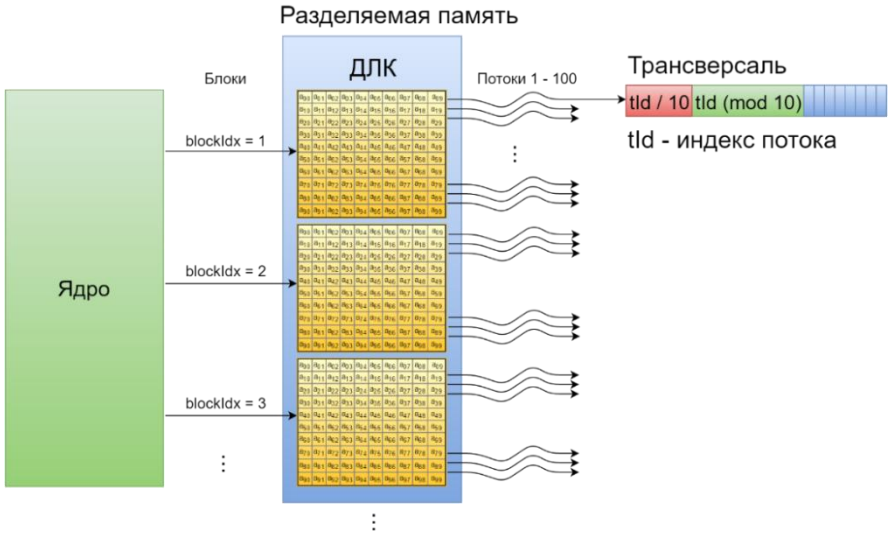

Рис. 1. Схема первого этапа алгоритма (поиска трансверсалей)

Отметим, что тривиальная реализация первого этапа заключается в рекурсивном поиске всех возможных трансверсалей. Однако данный подход обладает рядом минусов, связанных с дополнительными затратами ресурсов на рекурсивный вызов функций и сохранение стека. В рамках вычислений на GPU эти операции крайне нежелательны, т.к. стек записывается в гло-

бальную память, обладающую низкой скоростью работы. Именно поэтому при разработке алгоритма рекурсия на первом этапе использована не была.

В разработанном алгоритме на обоих этапах необходимо часто проверять занятость столбцов в текущий момент времени. Хранение таких данных в массивах приводит к их выгрузке в глобальную память устройства, что создает узкое место в работе всего алгоритма. В качестве решения данной проблемы применяется подход с хранением данных в регистрах в виде битовых масок:

$$M = \langle m_0, m_1 \dots m_9 \rangle, m_i \in \{0,1\}, i = 0, \dots 9.$$

В момент инициализации алгоритма все элементы кортежа M равны 0. При заполнении определенной ячейки элемент кортежа M с соответствующим индексом заменяется на 1.

Опишем второй этап алгоритма. Сначала все множество трансверсалей ДЛК упорядочивается по значению нулевого элемента трансверсали. Каждой трансверсали в соответствие ставится кортеж, который является битовой маской элементов ДЛК, покрываемых трансверсалью:

$$M = \langle m_0, m_1 \dots m_{99} \rangle, m_i \in \{0,1\}, i = 0, \dots 99.$$

Каждый поток также содержит аналогичную битовую маску, что позволяет сохранять информацию о заполнении элементов таблицы ДЛК в регистровой памяти. На этапе инициализации алгоритма все элементы данной маски заполняются нулями. Распараллеливание алгоритма на данном этапе производится путем фиксирования индекса нулевой трансверсали в покрытии. Более конкретно, для каждой трансверсали, битовая маска которой содержит элемент $m_0 = 1$ запускается отдельный поток. Схема выполнения второго этапа алгоритма показана на рис. 2.

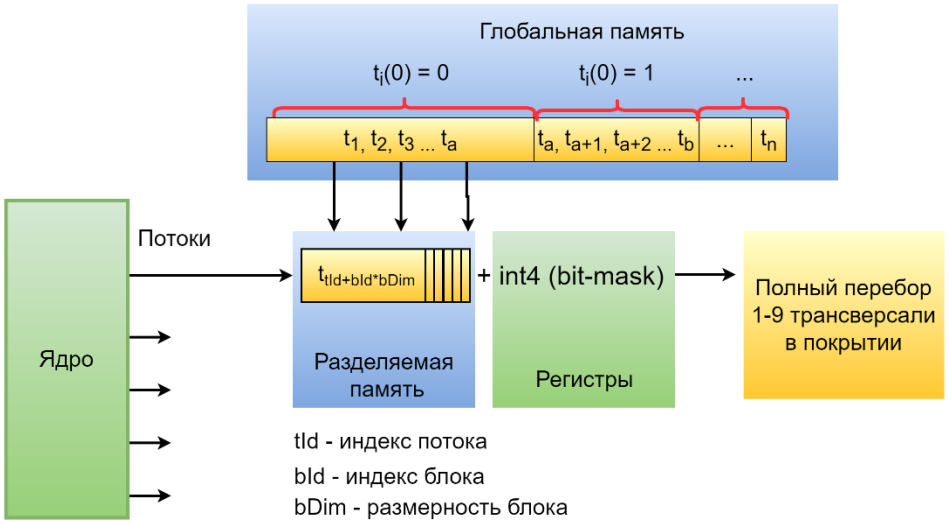

Рис. 2. Схема второго этапа алгоритма (поиск непересекающихся множеств трансверсалей)

3. Вычислительные эксперименты

В наших экспериментах было рассмотрено следующее частичное начальное заполнение ДЛК, состоящее из 40 ячеек:

0 1 2 3 4 5 6 7 8 9

2 4 9 7 0 3 5 1 6 8

9 7 1 5 6 4 8 3 0 2

1 0 8 6 9 2 7 4 5 3

Это частичное заполнение было взято из первого ДЛК порядка 10 в ортогональной паре, найденной в проекте SAT@home 20 апреля 2015 года [3]. С помощью программы, описанной в статье [6], были сгенерированы все возможные ДЛК, у которых значения первых четырех строк совпадают с указанным частичным заполнением. Всего оказалось 11 191 142 таких ДЛК.

Нами были сделаны две реализации предложенного в предыдущем разделе алгоритма – для CPU и GPU. В первом случае был написан кроссплатформенный x86-совместимый исходный

код на языке программирования Си. Во втором случае был написан исходный код на языке Си, ориентированный на архитектуру CUDA (данный исходный код доступен онлайн[1]). Оба разработанных приложения были запущены на обработку всего упомянутого выше множества ДЛК. Дополнительно была реализована проверка, можно ли на основе всех найденных в процессе обработки ортогональных пар составить тройку попарно ортогональных ДЛК порядка 10. Данная проверка требует значительно меньше ресурсов в сравнении с поиском ортогональных пар. GPU-приложение было запущено на GeForce GTX Titan (2688 CUDA-ядер), для запуска CPU-приложения использовался шестиядерный Xeon E5-1650. CPU-приложение было запущено на одном из шести ядер центрального процессора, GPU-приложение имело возможность задействовать все ядра графического процессора.

GPU-приложение завершило обработку описанных выше входных данных за 41 минуту 55 секунд, а CPU-приложение – за 16 часов 37 минут 23 секунды, т.е. GPU-приложение оказалось примерно в 23.8 раз быстрее. В обоих случаях была найдена только одна ортогональная пара – т.е. было доказано, что на основе указанного выше частичного заполнения нельзя построить ДЛК, который бы участвовал в тройке попарно ортогональных ДЛК порядка 10. Анализ результатов показал, что узким местом в GPU-реализации оказался первый этап алгоритма, заключающийся в построении всех трансверсалей для заданного ДЛК (см. раздел 2). В будущем мы планируем сконцентрироваться на оптимизации реализации именно этого этапа.

4. Заключение

Проведенные вычислительные эксперименты показывают, что разработанный алгоритм, основанный на методе Эйлера-Паркера, хорошо подходит для реализации на GPU. Мы планируем дальнейшую оптимизацию разработанного GPU-приложения. Также на основе разработанной GPU-реализации планируется запуск масштабного эксперимента в рамках проекта добровольных распределенных вычислений.

Литература

1. Малых А.Е., Данилова В.И. Об историческом процессе развития теории латинских квадратов и некоторых их приложениях // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2010. № 4. С. 95–104.
2. Brown J.W., Cherry F., Most L., Most M., Parker E.T., Wallis W.D. Completion of the spectrum of orthogonal diagonal Latin squares // Lecture notes in pure and applied mathematics. 1992. Vol. 139. P. 43–49.
3. Заикин О.С., Ватутин Э.И., Журавлев А.Д., Манзюк М.О. Применение высокопроизводительных вычислений для поиска троек взаимно частично ортогональных диагональных латинских квадратов порядка 10 // Труды десятой международной научной конференции Параллельные вычислительные технологии (ПаВТ'2016). Архангельск, Россия. 2016. С. 155-166.
4. Заикин О.С., Семенов А.А., Посыпкин М.А. Процедуры построения декомпозиционных множеств для распределенного решения SAT-задач в проекте добровольных вычислений SAT@home // Управление большими системами. 2013. Вып. 43. С. 138–156.
5. Parker E.T. Computer investigation of orthogonal Latin squares of order ten // Proceedings of Symposia in Applied Mathematics. 1963. Vol. 15. P. 73–81.
6. Ватутин Э.И., Журавлев А.Д., Заикин О.С., Титов В.С. Особенности использования взвешивающих эвристик в задаче поиска диагональных латинских квадратов // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2015. № 3 (16). С. 18–30.

[1] https://github.com/Banteu/EulerParkerCuda/

Using GPU for searching pairs of orthogonal diagonal Latin squares of order 10*

I.V. Shutov[1], S.E. Kochemazov[2], I.I. Kurochkin[1], O.S. Zaikin[2], E.I. Vatutin[3]
Kharkevich Institute for Information Transmission Problems
of Russian Academy of Sciences

Matrosov Institute for System Dynamics and Control Theory
of Siberian Branch of Russian Academy of Sciences[2]
Southwest State University[3]

The problem of searching for all diagonal Latin squares of order 10, which form an orthogonal pair with a given diagonal Latin square of order 10 is considered. To solve this problem an algorithm was developed, which is based on the Euler-Parker method. On the first stage of the algorithm all transversals of the given square are found. On the second stage all possible sets of disjoint transversals are formed (number of transversals in every such set must be equal to the order of the given square). Each such set of the transversals corresponds to an orthogonality pair, in which the original square is present. Two implementations of the algorithm were made - for CPU and CUDA-based GPU. The computational experiments showed that the GPU implementation greatly outperforms the CPU implementation.

Keywords: Latin square, orthogonality, GPU.

References

1. Malih A.E., Danilova V.I. Ob istoricheskom processe razvitiya teorii latinskih kvadratov i nekotoryh ih prilozheniyah [About historical process of the evolution of Latin squares and some their applications] // Vestnik Permskogo universiteta. Seria: Matematika, Mehanika, Informatika [Bulletin of Perm University: Mathematics, Mechanics, Computer Science]. 2010. No. 4. P. 95–104.
2. Brown J.W., Cherry F., Most L., Most M., Parker E.T., Wallis W.D. Completion of the spectrum of orthogonal diagonal Latin squares // Lecture notes in pure and applied mathematics. 1992. Vol. 139. P. 43–49.
3. Zaikin O.S., Vatutin E.I., Zhuravlev A.D., Manzyuk M.O. Applying high-performance computing to searching for triples of partially orthogonal Latin squares of order 10 // Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies (PCT'2016). Arkhangelsk, Russia. 2016. CEUR-WS. Vol. 1576. P. 155-166.
4. Zaikin O.S., Semenov A.A., Posypkin M.A. Constructing decomposition sets for distributed solution of SAT problems in volunteer computing project SAT@home // Large-Scale Systems Control. 2013. Issue 43. P. 138–156.
5. Parker E.T. Computer investigation of orthogonal Latin squares of order ten // Proceedings of Symposia in Applied Mathematics. 1963. Vol. 15. P. 73–81.
6. Vatutin E.I., Zhuravlev A.D., Zaikin O.S., Titov V.S. Features of the use of weighting heuristics in the search for diagonal Latin squares // Proceedings of the South-West State University. Series Control, computer engineering, information science. Medical instruments engineering. 2015. No. 3 (16). P. 18–30.

* This work was partially supported by Russian Foundation for Basic Research (grants 14-07-00403-a, 15-07-07891-a, 16-07-00155-a, 15-29-07095-ofi_m and 16-07-00659-a) and by Council for Grants of the President of the Russian Federation (stipend SP-1184.2015.5, grants MK-9445.2016.8 and NSh-8081.2016.9). This work was done within the state assignment for Southwest State University, NIR 2246, 2014–2017.