УДК 004.051

ОПТИМИЗАЦИЯ РЕСУРСОЁМКИХ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ НА ГРАФИЧЕСКОМ ПРОЦЕССОРЕ

© 2016 А.А. Сытник, С.П. Ивженко, И.В. Гвоздюк

Саратовский государственный технический университет имени Гагарина Ю.А.

Статья поступила в редакцию 11.11.2016

Произведён анализ возможности запуска вычислительных задач на графических процессорах (GPU), исследовано, за счёт чего достигается такая эффективность и какие меры необходимо предпринять, для её достижения. В работе продемонстрировано преимущество вычислений на графических процессорах в задачах общего назначения, на примере работы с матрицами, которое было достигнуто за счёт оптимизации кода для выполнения в параллельном режиме на GPU. Авторами предлагается программная библиотека, предоставляющая API для выполнения задач обработки массивов данных, с оптимизацией для выполнения на графическом процессоре.
Ключевые слова: параллельные вычисления, графический процессор, OpenGL, GPU, производительность, матрицы.

Параллельные вычисления на сегодняшний день являются одной из наиболее актуальных тем для исследований [1]. Множество прикладных задач в приборостроении требуют значительных вычислительных ресурсов при скромных технических возможностях исследователей. Противоречие решается использованием графических процессоров (GPU) с присущим им параллелизмом обработки информации и особых вычислительных алгоритмов. Центральный процессор не всегда способен решать сложные вычислительные задачи за приемлемое время. Графические процессоры изначально проектируются таким образом, чтобы обрабатывать огромные объемы данных.

Графические процессоры имеют свои особенности в сравнении с центральными процессорами (CPU), которые могут быть критичными для принятия решения, привлекать ли его для использования в той или иной задаче. В отличие от современных универсальных центральных процессоров, видеочипы предназначены для параллельных вычислений с большим количеством арифметических операций. И значительно большее число одинаковых вычислительных устройств — потоковых процессоров графического процессора (см. рис. 1) работает по прямому назначению – обработке массивов данных, а не управляет исполнением немногочисленных последовательных вычислительных потоков. На рисунке показано, сколько места в CPU и GPU занимает разнообразная логика [2].

На данный момент имеется несколько проектов, чьей конечной целью является создание программных продуктов, позволяющих использовать мощности GPU в задачах общего назначения. Одной из самых популярных стала разработка компании NVIDIA. Результатом усилий этой команды стала NVIDIA CUDA (Compute Unified Device Architecture) – новая программно-аппаратная архитектура для параллельных вычислений на NVIDIA GPU [4]. Основной проблемой библиотеки CUDA является ограниченный набор поддерживаемых графических процессоров: данная библиотека не работает на видеокартах AMD.

Библиотека OpenCL лишена этого недостатка и может в ближайшее время стать стандартом. Стоит отметить, что язык написания программ для этой библиотеки абстрагирован от типа устройства, на котором этот код будет выполняться. Один и тот же код можно выполнить как на графическом, так и на центральном процессорах [4].

Выполнение расчётов на GPU показывает отличные результаты в алгоритмах, когда одну и ту же последовательность математических операций применяют к большому объёму данных. При этом лучшие результаты достигаются, если отношение числа арифметических инструкций к числу обращений к памяти достаточно велико [5].

Это относится к операциям умножения плотных матриц, тогда как при сложении матриц при-

Сытник Александр Александрович, доктор технических наук, профессор, заведующий кафедрой «Информационно-коммуникационные системы и программная инженерия». E-mail: as@sstu.ru

Ивженко Сергей Петрович, кандидат физико-математических наук, доцент кафедры «Информационно-коммуникационные системы и программная инженерия». E-mail: sarvizir@mail.ru

Гвоздюк Илья Вячеславович, аспирант кафедры «Информационно-коммуникационные системы и программная инженерия». E-mail: gvozdiuk@gmail.com

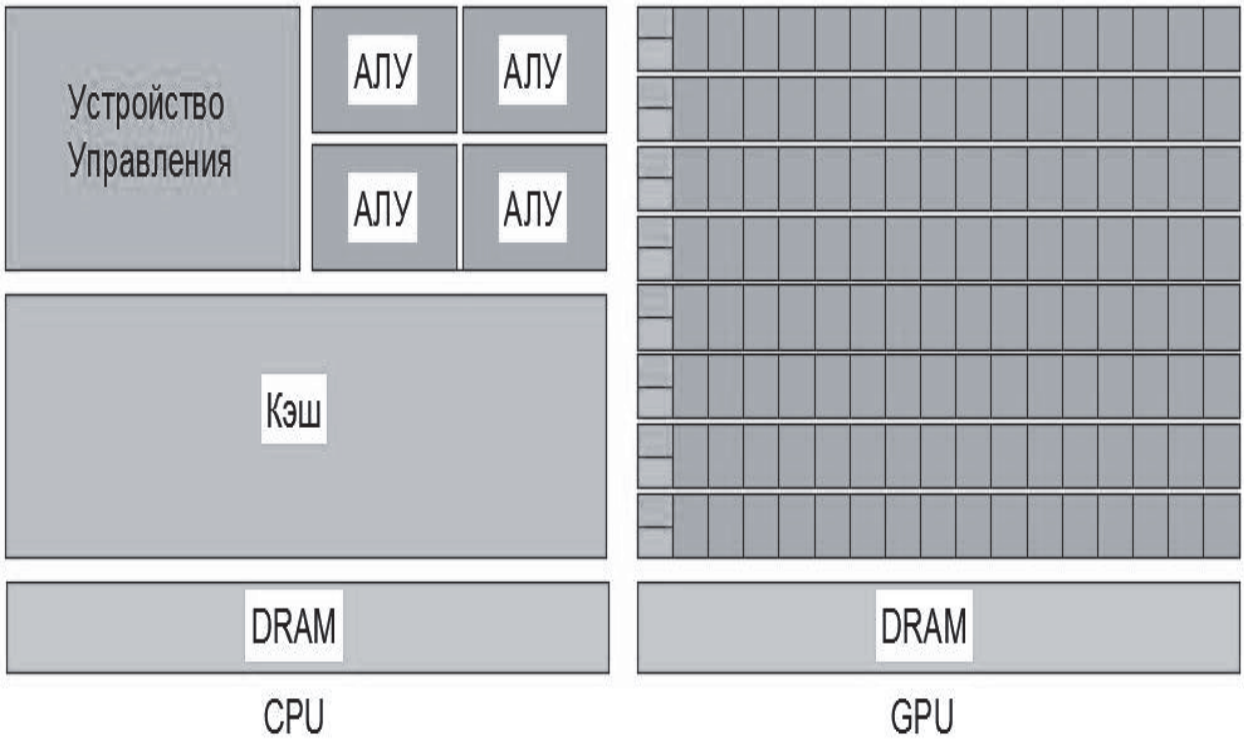

Рис. 1. Устройство CPU и GPU

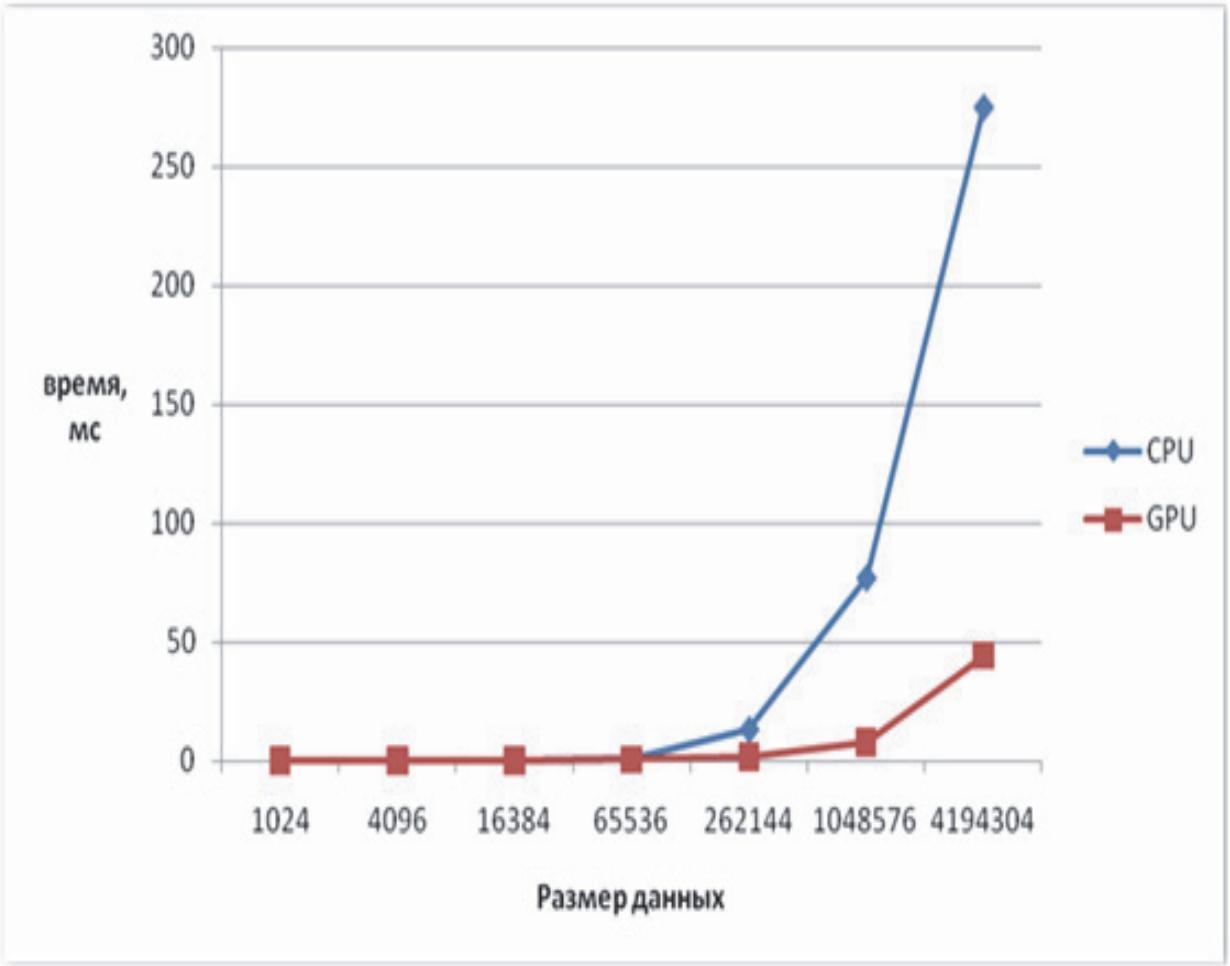

Рис. 2. Графики эффективности алгоритмов транспонирования матрицы

рост производительности считаные проценты [2]. Представляет трудность умножение разреженных матриц, которые имеют обычно сложную непредсказуемую структуру, и поэтому эффективных алгоритмов в этом случае не разработано. Также есть проблемы при реализации обращения плотной матрицы [1]. Обработка изображений естественным образом сочетается с GPU, однако фильтрация изображений показывает худшие результаты, чем преобразование Фурье из-за меньшего коэффициента повторного использования данных.

В связи с высоким вычислительным потенциалом графических процессоров образовалось направление GPGPU (General Purpose Graphics Processors Usage – использование графических процессоров для общих целей) [6]. Целью этого направления является не только обработка трехмерной графики, но и решение общих задач, например таких, как матричные вычисления в работе [3]. На данный момент данная концепция не получила широкого распространения, несмотря на то, что появились некие стандартизованные подходы к разработке. Во многом это обусловлено тем, что программы, предназначенные для выполнения на GPU, не оптимизированы и оптимизация таких программ требует дополнительных знаний о GPU, а также требует больших усилий.

Авторами работы разработана программная библиотека, реализующая оптимизированные алгоритмы вычислений для выполнения в параллельном режиме на графическом процессоре в соответствии с технологией GPGPU.

Была исследована производительность системы при решении вычислительных задач общего назначения на графическом процессоре и были выработаны действия, необходимые для её повышения за счет использования технологии OpenGL. Реализовать простейший алгоритм обработки данных на GPU не трудно, но оптимизировать таким образом, чтобы обеспечить максимальное ускорение по сравнению с последовательной версией алгоритма, значительно трудней. В ходе исследования особенностей написания кода для выполнения на GPU были выделены следующие направления оптимизации:

- оптимизация работы с памятью;
- оптимизация арифметических операций;
- увеличение количества потоков в блоке;
- уменьшение объёма работы, выполняемой потоком.

Была разработана библиотека функций, позволяющая выполнять действия по перебору массивов с оптимизацией на параллельные вычисления. На данный момент, возможности разработанной библиотеки весьма ограниченны, но, несмотря на это, уже можно видеть результат. К примеру, оптимизировав код по транспонированию матриц для запуска на графическом процессоре по технологии CUDA, удалось достичь результатов, графически представленных на рис. 2.

По графику видно, что с увеличением объёма входных данных вырастает и производительность GPU по сравнению с CPU, то есть начинает сказываться режим выполнения вычислений в параллельном режиме. За счет последовательного применения предложенных способов оптимизации можно добиться значительной скорости вычислений.

Таким образом, авторами предлагается библиотека, предоставляющая API для выполнения разнообразных операций над данными с оптимизацией под GPU, которая позволяет облегчить труд программистов по реконструированию алгоритмов. Разработанная библиотека позволила увеличить эффективность решения вычислительных задач на графических процессорах, что в полной мере подтверждают результаты анализа, полученные в ходе выполнения работы.

СПИСОК ЛИТЕРАТУРЫ

1. *Адинец А., Воеводин В.* Графический вызов суперкомпьютерам. URL: http://www.osp.ru/os/2008/04/5114497 (дата обращения 20.09.2016).
1. *Дымченко Л.* Параллельные вычислительные процессоры NVIDIA: настоящее и будущее. URL: http://nvworld.ru/articles/cuda_parallel/ (дата обращения 20.09.2016).
1. *Максимов А.А., Папшев С.В.* Индексы и периоды нечетких матриц // Вестник Саратовского государственного технического университета. 2011. № 55. С. 147-157.
1. *Хомюк С.С.* Использование графического процессора для решения задач общего назначения. // Информационные технологии и математическое моделирование (ИТММ - 2008): Материалы VII Всероссийской научно-практической конференции с международным участием (14-15 ноября 2008). Томск. Изд-во Том. ун-та, 2008. Ч1. С. 132-134
1. *Tarditi D., Puri S., Oglesby J.* Accelerator: Using Data Parallelism to Program GPUs for General-Purpose Uses. Microsoft Research, 2006. 11 с.
1. GPGPU: General Purpose Computation on Graphics Hardware / *D. Luebke, M. Harris, J. Kruger, и др.* SIGGRAPH, 2005. 277 с.

OPTIMIZATION OF RESOURCE-INTENSIVE COMPUTING TASKS ON GRAPHICS PROCESSOR

© 2016 A.A. Sytnik, S.P. Ivzhenko, I.V. Gvozdiuk

Yuri Gagarin State Technical University of Saratov

Produced analyze of the possibility of running computational tasks on graphics processors (GPU), investigated why such efficiency is achieved and what measures need to be taken to achieve it. Demonstrated the advantage of computing on graphics processor units in general purpose of the example matrix, which was achieved by optimizing the code to run in parallel on the GPU. Offered a software library that provides API to perform the tasks of processing data arrays optimized for execution on the GPU.
Keywords: parallel computing, graphics processor, OpenGL, GPU, performance, matrices.

Aleksandr Sytnik, Doctor of Technics, Professor, Head at the Information and Communication Systems and Software Engineering Department. E-mail: as@sstu.ru
Sergey Ivzhenko, Candidate of Physics and Mathematics, Associate Professor at the Information and Communication Systems and Software Engineering Department. E-mail: sarvizir@mail.ru
Ilya Gvozdyuk, Graduate Student at the Information and Communication Systems and Software Engineering Department. E-mail: gvozdiuk@gmail.com