Расчет торможения электронов в веществе с использованием вычислений на GPU методом Монте-Карло

М.Е. Вахно, Д.В. Коротяев, В.А. Дубинин

Поморский Государственный Университет им. М.В. Ломоносова, Архангельск

Введение

Вторичные электроны и рентгеновское излучение, возникающие при торможении в веществе электронных пучков с энергией E_0~10-100 КэВ, дают важную информацию о строении вещества [1,2]. Детали основных физических процессов можно найти в [1,2]. Цель работы – реализовать вычисления траекторий и скоростей электронов в веществе с использованием мощностей процессоров графического ускорителя (GPU) и расчет распределения электронов в фазовом пространстве на основе приведенных в [1,2] алгоритмов.

Для решения задачи было решено ориентироваться на графические процессоры серии G80 фирмы NVIDIA, которая благодаря .разработки технологии CUDA, на языке C, значительно упростила создание и отладку приложений использующих вычисления проводимые на GPU . Использование современных GPU позволяет существенно повысить производительность при выполнении математических вычислений в сравнении с процессором общего назначения. Высокий уровень параллелизма здесь достигается за счет большого количества вычислительных ядер на одном устройстве. Основная модель вычислений строится по методу SIMD (Одна инструкция, много данных), в терминологии NVIDIA – SIMT (Одна инструкция, много нитей). Такой подход позволяет производить большое количество однотипных вычислений параллельно, для различных наборов данных. Метод Монте-Карло, используемый в качестве основы расчета в данной работе, является одним из наиболее подходящих кандидатов такой модели вычисления, благодаря простоте его декомпозиции на вычислительные потоки.

Реализация метода Монте-Карло для вычислений на GPU

Создание программы для графического процессора несколько специфично, и отличается от подхода, применяемого при кодировании программ для процессоров общего назначения. В данном случае работает множество вычислительных потоков и, необходимо тщательно планировать синхронизации параллельно исполняющихся потоков, а также эффективно использовать иерархию памяти, предупреждая возможные конфликты при обращениях к банкам разделенной памяти. Потоки, которые должны исполняться, в зависимости от количества доступных процессоров формируют связки потоков(warps) и компонуются в блоки(blocks) , которые в свою очередь собираются в наборы мультипроцессоров(grids). Таким образом, необходимо исходя из возможностей аппаратуры, провести декомпозицию вычислительных потоков на вычислительную архитектуру графического процессора, наиболее эффективно используя разделенную память

В связи с тем, что в алгоритме присутствует ветвления, когда ход исполнения в зависимости от результатов розыгрыша по методу Монте-Карло должен идти по тому или другому пути, при использовании одного ядра(kernel) для расчета новых параметров каждого электрона не будет достигаться оптимальная загрузка мощностей GPU [3, стр. 47]. Большая производительность GPU должна будет достигаться в том случае, когда для каждой из ветвей исполнения будет реализовано собственное ядро и блоки нитей, рассчитывающие ход разных ветвей будут исполняться разными наборами мультипроцессоров.

Таким образом, весь процесс обработки движения электронов проходит в несколько этапов:

1. Вычисление времени взаимодействия, координат электрона на следующем шаге и типа его взаимодействия.
2. Распределение электронов в 2 массива по виду их взаимодействия с веществом на данном этапе (сильное и слабое взаимодействия)
3. Подсчет новых параметров скорости на основе типа взаимодействия (в 2-х различных ядрах для каждого из массивов).
4. Выборка электронов, для которых необходимо проводить следующий этап вычислений.

Как только скорость электрона становится ниже минимальной, расчеты для него прекращаются. После этого производится расчет количества электронов в ячейках фазового пространства. Необходимое количество шагов определяется исходя из объема доступной оперативной памяти на устройстве GPU. При необходимости расчеты можно повторить при помощи тех же структур, используя результаты последнего шага вычислений как начальные параметры скоростей и координат электронов.

Доступная в данной работе аппаратура ограничивает точность вычислений одинарной. Тем не менее, как показали проведенные эксперименты, благодаря особенностям метода Монте-Карло, накапливаемая из-за низкой точности ошибка, нивелируется числом проведенных розыгрышей. Метод Монте-Карло подразумевает использование случайных величин для вычислений. Для вычислений на GPU отсутствуют стандартные функции получения случайного числа, а стандартные функции языка C не могут быть использованы в процедурах и ядрах для GPU. При решении задачи использовалась функция генерации случайного числа, входящая в состав CUDA SDK.

В докладе представлены материалы для сравнения скорости работы алгоритма на различных вычислительных системах, среди которых представлены: однопроцессорные CPU, многопроцессорные CPU и графические процессоры серии G80 и G90 фирмы NVIDIA. Представлены результаты расчета коэффициента эффективности распараллеливания. Также представлен подход к реализации алгоритма расчета торможения электронов в веществе по методу Монте-Карло и постановка вычислений на GPU с использованием технологии CUDA.

Литература

1. В.А.Астапенко, В.В.Березовский, П.Л.Меньшиков, Л.И. Меньшиков. Подходы к разработке быстрого кода Монте-Карло для расчёта торможения электронов в веществе. Труды МФТИ. вып. 2.
2. В.А.Астапенко, В.В.Березовский, П.Л.Меньшиков, Л.И. Меньшиков. Разработка вычислительной системы для расчёта торможения электронов в веществе**.**
3. NVIDIA CUDA C Programming Best Practices Guide