Физическое моделирование на OpenCL

|  |  |
| --- | --- |
| Боголепов Д. bogdencmc@inbox.ru | Блохин О. blohin.o.d@gmail.com |
| Захаров М. maxim.zakharov@inbox.ru | Сопин Д. sopindm@gmail.com |
| Удалова Т. udalova.t@gmail.com | Калишев Г. gleb-kalishev@rambler.ru |

# Введение

В конце 2008 года организация Khronos Group утвердила новый стандарт OpenCL. Он предназначен для осуществления вычислений на гетерогенных системах. В апреле 2009 года Nvidia и AMD, выпустили драйвера с поддержкой OpenCL, но до сих пор мало информации о производительности решений на базе этой технологии.

# Постановка задачи

Для тестирования производительности была выбрана задача моделирования из класса N-body problems[1], задачи такого типа хорошо поддаются распараллеливанию.

Рассматривается эволюция системы N взаимодействующих точечных зарядов в магнитном поле. Действие зарядов друг на друга определяется по закону Кулона [2 p129], а влияние магнитного поля на частицу – силой Лоренца [2 p182].

# Описание решения

Система дифференциальных уравнений, которая определяет изменение ускорения частиц, интегрируется при помощи метода Эйлера. Основную долю вычислений при решении задачи составляет расчёт ускорения каждой частицы.

Мы хотим узнать, насколько теоретическую производительность реально достичь на практике, используя OpenCL.

При подсчёте, мы предполагаем, что будут произведены все возможные оптимизации исходного кода. Таким образом, получим оценку для числа затраченных операций снизу, а для частоты кадров – сверху.

|  |  |  |  |
| --- | --- | --- | --- |
| Процедура расчета ускорения для одной частицы | flops | GPU cycles | CPU cycles |
| 1|for ( int i = 0; i < N; ++i ) |  |  |  |
| 2|{ |  |  |  |
| 3| // Calculate acceleration caused by particle j  on particle i |  |  |  |
| 4| float3 r = p - pi; | 3 | 3 | 1 |
| 5| float dist = dot ( r, r ); | 5 | 5 | 2 |
| 6| float invDist = inversesqrt ( dist + SOFTENING ); | 2 | 2 | 4 |
| 7| float invDistCube = invDist \* invDist \* invDist; | 2 | 2 | 2 |
| 8| float s = q \* qi \* invDistCube; | 2 | 2 | 2 |
| 9| // accumulate effect of all particles |  |  |  |
| 10| acc += s \* r; | 6 | 3 | 2 |
| 11|} |  |  |  |
| Итого: | 20 | 17 | 10 |

Для CPU цикл for был полностью развернут компилятором. Различие в количестве циклов, затрачиваемых на выполнение одинаковых операций GPU и CPU, обусловлено особенностями их архитектуры. Благодаря векторизации, CPU способен выполнять несколько однотипных операций за один цикл. В связи с этим операция dot выполняется на GPU за 5 циклов (3 умножения и 2 сложения), а на CPU – за 2 (1 векторное умножение и одно векторное сложение).

Цифры 10-ой строчки таблицы обусловлены тем, что в системе команд GPU есть инструкция одновременного умножения и сложения (mad – multiple add). На CPU же операция просто векторизуется.

Мы получили, что на расчет движения одной частицы приходится 20∙N flops, а на одну итерацию по методу Эйлера – 20∙N2 flops. Число операций растёт пропорционально квадрату количества частиц.

# Результаты экспериментов

Эксперименты проводились для N = 16384. Для теоретической оценки производительности мы пользовались формулой:

Таблица Характеристики оборудования

|  |  |  |
| --- | --- | --- |
| Устройство | Число процессоров | Частота одного процессора, MHz |
| GeForce 8800 GTS | 96 | 1200 |
| GeForce 8800 GT | 112 | 1500 |
| AMD Athlon 64 X2 3800+ | 2 | 2010 |
| Intel Core 2 Duo E8200 | 2 | 2670 |

Таблица Сравнение теоретической и практической производительности OpenCL для CPU и GPU

|  |  |  |
| --- | --- | --- |
| Архитектура | Теоретический FPS | Реальный FPS |
| GeForce 8800 GTS | 25.2 | 14.9 |
| GeForce 8800 GT | 36.8 | 34.7 |
| AMD Athlon 64 X2 3800+ | 1.98 | 0.2 |
| Intel Core 2 Duo E8200 | 1.98 | 0.2 |

Таблица Сравнение OpenCL и OpenMP на CPU для N = 256

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Архитектура | Теоретический FPS | OpenCL | OpenMP, MS C++ 9.0 | OpenMP, Intel C++ 11.0 |
| Athlon 3800+ | 1533.5 | 158.3 | 52.1 | 312.2 |
| Intel E8200 | 2034.7 | 240.8 | 252.4 | 1719.3 |

Таблица Сравнение производительсности OpenCL, CUDA и GLSL на GPU для N=16384

|  |  |  |  |
| --- | --- | --- | --- |
| Архитектура | OpenCL | CUDA | GLSL |
| GeForce 8800 GTS | 14.9 |  | 15.5 |
| GeForce 8800 GT | 34.7 |  | 22.5 |

# Заключение

Несмотря на то, что стандарт продолжает развиваться и не везде поддерживается в полной мере, OpenCL предоставляет приятные перспективы развития отрасли. На тестах новая технология показывает близкие и, иногда, даже более хорошие результаты, по сравнению с существующими решениями.

# Литература

1. *Erich Elsen V., Vishal Mike Houston и др. N-Body Simulations on GPUs*.  
   <http://arxiv.org/pdf/0706.3060>
2. *Курс Физики. Трофимова Т. И. 1990г.*