

Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики
Кафедра «Прикладная математика»

КУРСОВАЯ РАБОТА

**Оптимизация обработки потока частиц, сталкивающихся
с препятствием**

по дисциплине «Использование графических процессоров для
вычислений»

Выполнил,
студент гр. 23641/1

Гарбузов Ф.Е.

Руководитель,
ассистент

Чуканов В.С.

Санкт-Петербург
2018

Условные обозначения

CPU – Central Processor Unit, центральный процессор

GPU – Graphics processing unit, графический процессор

kernel – подпрограмма, исполняемая на графическом процессоре

мкс – микросекунда

Постановка задачи

1. Написать графическое приложение, которое визуализирует поток частиц, испускаемых точечным источником, причём частицы
 - имеют случайную в некотором диапазоне начальную скорость и случайное время жизни,
 - двигаются в поле силы тяжести,
 - упруго сталкиваются с препятствием произвольной формы.

Для обработки потока частиц приложение должно использовать графический процессор.

2. Провести анализ кода, найти узкие места, замедляющие работу приложения, и оптимизировать приложение.

Решение

Задание 1

В качестве препятствия выбран вращающийся куб. Количество частиц – около 10^5 . Количество используемых блоков – 128, размерность блока – 64 процесса.

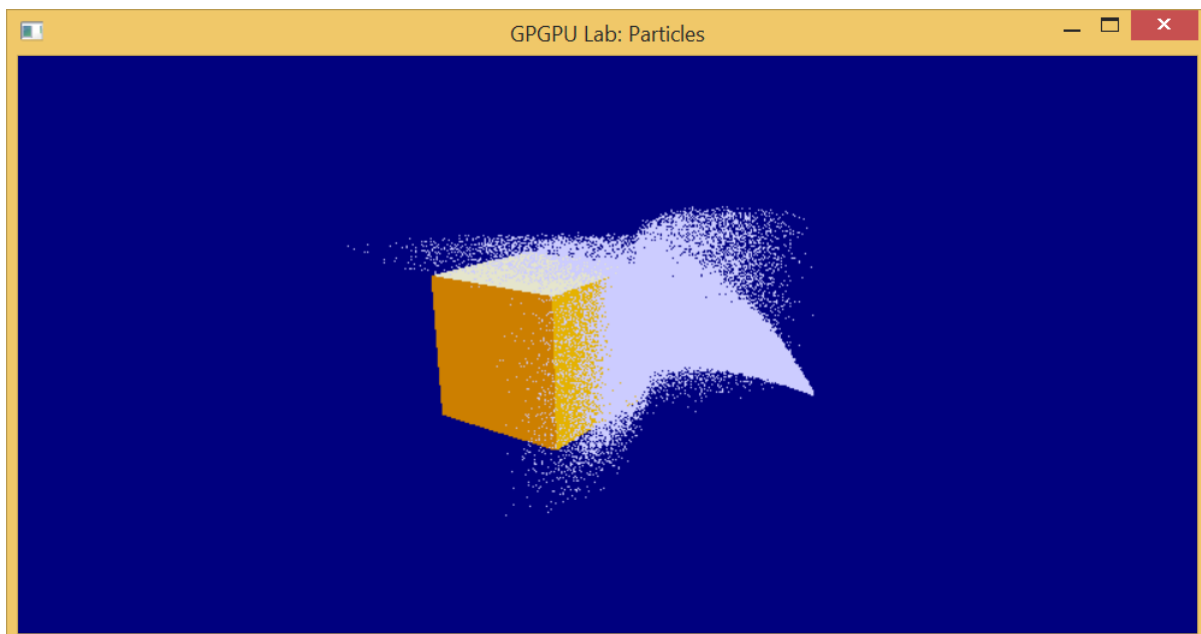


Рисунок 1: скриншот приложения.

Оценивать эффективность работы kernel будем по её среднему времени работы. Среднее время работы для данного приложения составила 2410 мкс. Скриншот профайлера представлен на рисунке 2.

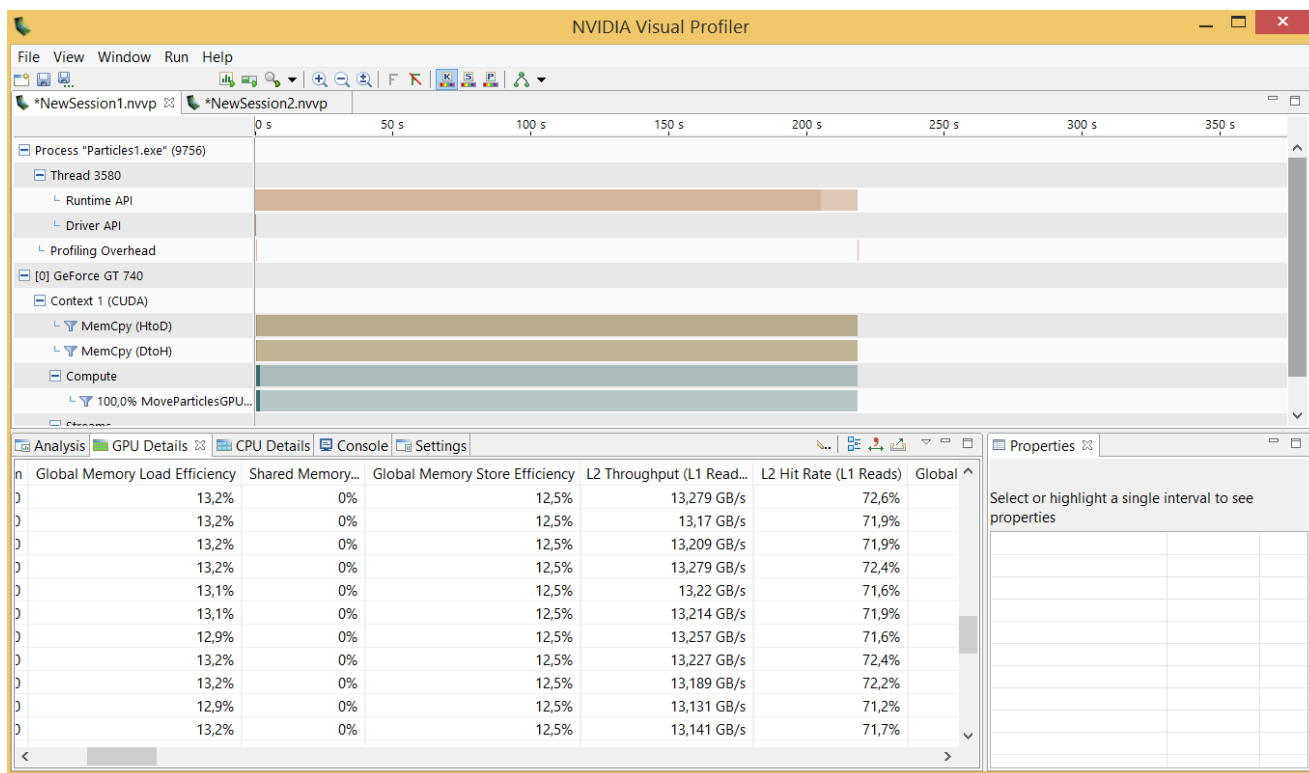


Рисунок 2: скриншот профайлера.

Задание 2

Основными источниками снижения производительности kernel являются:

- 1) операции чтения и записи в глобальную память на GPU,
- 2) операции над данными на GPU (арифметические операции и ветвления).

Установлено, что именно операции чтения и записи в глобальную память больше всего снижают производительность. Для оптимизации внесены следующие изменения:

- 1) для ускорения доступа к параметрам препятствия-куба эти параметры отправляются в константную память, обладающую лучшим по сравнению с глобальной памятью механизмом кеширования;
- 2) для ускорения обработки частиц каждый поток в блоке вместо обработки соседних в векторе частиц обрабатывает частицы, расположенные друг от друга на расстоянии, равном количеству процессов в блоке;
- 3) развёрнут цикл, пробегающий по всем граням препятствия-куба.

Изменение, приведённое в п.2, позволяет эффективнее использовать кэширование глобальной памяти, поскольку теперь соседние процессы в блоке обрабатывают соседние в векторе частицы.

Приведённые операции позволили понизить среднее время работы kernel до 478 мкс, т. е. примерно в 5 раз по сравнению с исходным значением.

Приложение	Среднее время kernel, мкс	Ускорение
Первое приложение	2410	
Оптимизированное	478	5.04x

Сводная таблица: сравнение производительности.

Скриншот профайлера представлен на рисунке 3. Видно, что произошло улучшение параметров Global Memory Load/Store Efficiency и L2 Hit Rate.

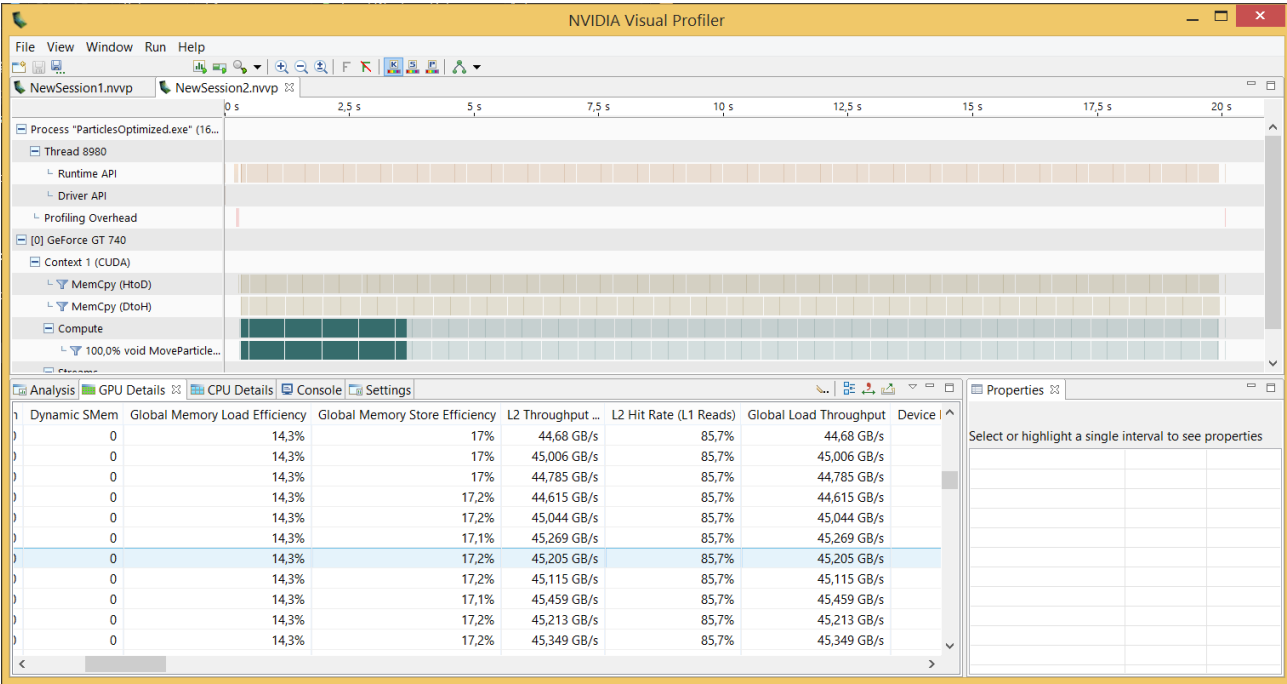


Рисунок 3: скриншот профайлера.