Библиотека для работы с конечными множествами, использующая графический процессор в качестве основного вычислительного устройства

Новосибирский Государственный Университет Механико-Математический Факультет Каф. Программирования, Лаборатория "Интел"

Выполнил:

Студент Зего курса ММФ НГУ Лыков Кирилл Олегович

Email: lykov.kirill@gmail.com

Научные руководители:

К.ф.-м.н., доцент каф. программирования Скопин И.Н. Инженер Google-Moscow Пасько Е.В.

В рамках проекта: Использование графических процессоров для вычислений с плавающей точкой в Java-программах. Exploiting Graphics Processing Units (GPUs) for general purpose computation in Java programming language

Мотивация использования GPU

• Скорость

- 3.0 GHz dual-core Pentium4: 24.6 GFLOPS
- NVIDIA GeForceFX 7800: 165 GFLOPs
- 1066 MHz FSB Pentium Extreme Edition: 8.5 GB/s
- ATI Radeon X850 XT Platinum Edition: 37.8 GB/s

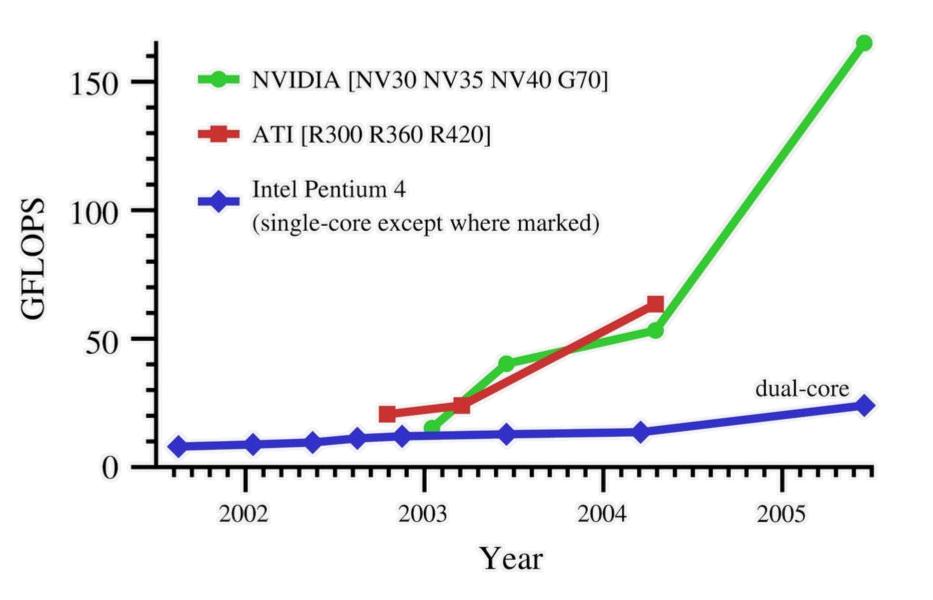
• GPUs становятся все быстрее

- CPUs: 1.4×годичный рост
- GPUs: 1.7×(pixels) to 2.3× (vertices)годичный рост

• Современные GPU обеспечивают приемлемую точность вычислений

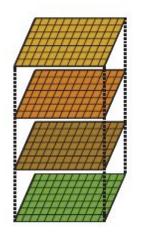
- Поддержка чисел с плавающей точкой (до 32 бит)

GPUs становятся все быстрее



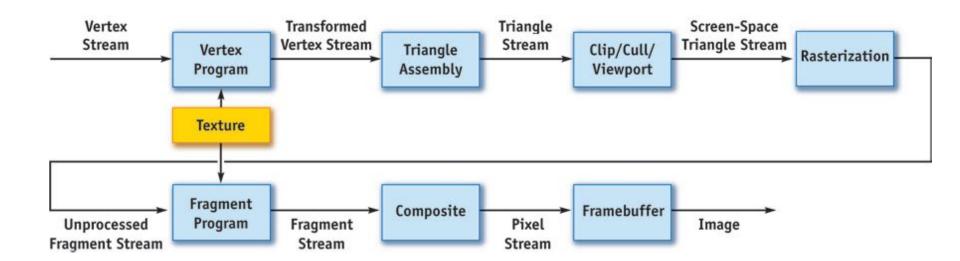
Основные термины и понятия GPU

Текстура - это трехмерный массив чисел X*Y*F, где F = 1..4, а X и Y размеры текстуры.



Шейдер - это программа, исполняемая непосредственно на GPU

<u>Графический конвейер с точки зрения</u> <u>потоковой модели вычислений</u>



Основные концепции GPGPU

- 1. Массив = текстура.
- CPU array (1D адресация) -> GPU texture (2D адресация)
- 2. Вычислительное ядро программы = шейдер.
- Часть программы для CPU ответственная за вычисления переносится соответствующим образом на GPU
- 3. Вычисление = отрисовка.
- 4. Получение результатов вычислений в буфер кадра (frame buffer).

Цель работы

• Создание библиотеки, позволяющей работать с битовыми множествами, причем операции над ними реализованы не на CPU, а на GPU. Особенностью такой реализации является использование видеокарты в качестве основного вычислительного устройства, что обуславливает использование потоковой модели вычислений на GPU.

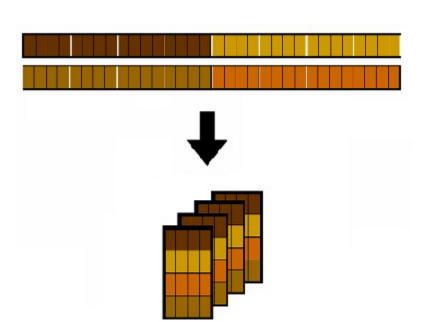
Представление битовых множеств на GPU

<u>Битовое представление</u> <u>множеств</u> 145983...

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 ...

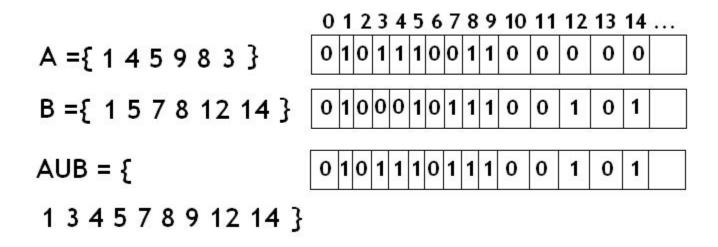
0 1 0 1 1 1 0 0 1 1 0 0 0 0 0

<u>Хранение данных в</u> <u>текстурной</u> памяти

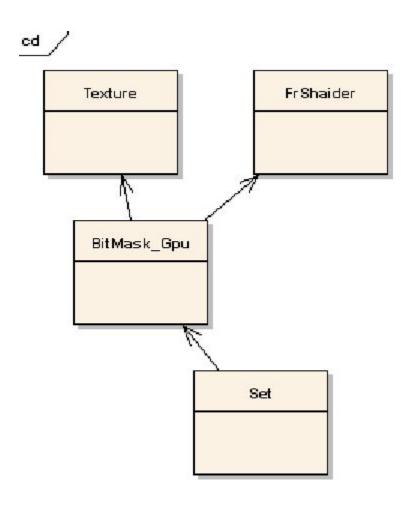


Операции над битовыми множествами

• Операциям над конечными множествами соответствуют логические операции над битовыми масками



Архитектура библиотеки



Результаты и планы

- Реализована основная часть необходимой функциональности.
- Необходимо добавить проверки на корректность в текст кода. Включить исключения.
- Необходимо провести комплексное тестирование работы на различных GPU, выделить слабые места в производительности. Оптимизировать код.

Литература

- 1. David Luebke. General-Purpose Computation on Graphics Hardware. University of Virginia.
- 2. Mark Harris. Mapping Computational Concepts to GPUs. NVIDIA
- 3. Dominik Göddeke. GPGPU::Basic Math Tutorial. www.mathematik.uni-dortmund.de/~goeddeke/gpgpu/tutorial.html