

# Лекция 1: Использование графических процессоров для ускорения вычислений. Введение

Н.Д. Смирнова

Санкт-Петербургский государственный Политехнический университет

15.09.2011

# Содержание

## 1 Лекции

- Что?
- Почему?
- Зачем?

## 2 Лабораторные

- Что используем
- Что делаем

# Содержание

- 1 Лекции
  - Что?
  - Почему?
  - Зачем?
- 2 Лабораторные
  - Что используем
  - Что делаем

## **GPGPU - General-purpose computing on graphics processor units<sup>1</sup>**

- решение различных вычислительно сложных распараллеливаемых задач
  - от обработки изображений
  - до решения уравнения Навье-Стокса
- на графических процессорах (GPU)
  - "без" использования обычных процессоров (CPU)

---

<sup>1</sup><http://en.wikipedia.org/wiki/GPGPU>

## GPGPU - General-purpose computing on graphics processor units <sup>1</sup>

- решение различных вычислительно сложных распараллеливаемых задач
  - от обработки изображений
  - до решения уравнения Навье-Стокса
- на графических процессорах (GPU)
  - "без" использования обычных процессоров (CPU)

---

<sup>1</sup><http://en.wikipedia.org/wiki/GPGPU>

## GPGPU - General-purpose computing on graphics processor units <sup>1</sup>

- решение различных вычислительно сложных распараллеливаемых задач
  - от обработки изображений
  - до решения уравнения Навье-Стокса
- на графических процессорах (GPU)
  - "без" использования обычных процессоров (CPU)

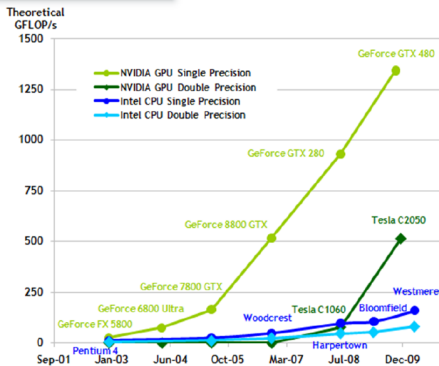
---

<sup>1</sup><http://en.wikipedia.org/wiki/GPGPU>

Что?

# Почему GPU?

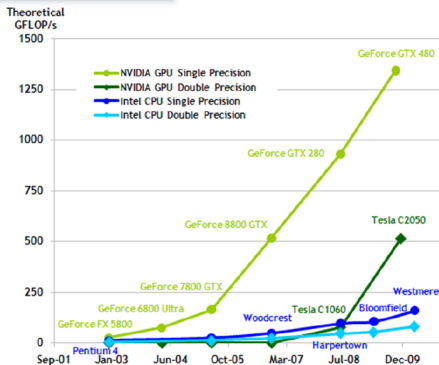
- быстрее, дешевле
- требует узко специальных знаний



Что?

# Почему GPU?

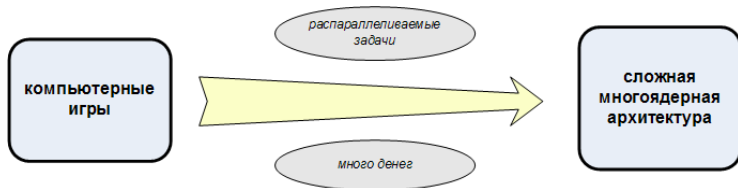
- быстрее, дешевле
- требует узко специальных знаний





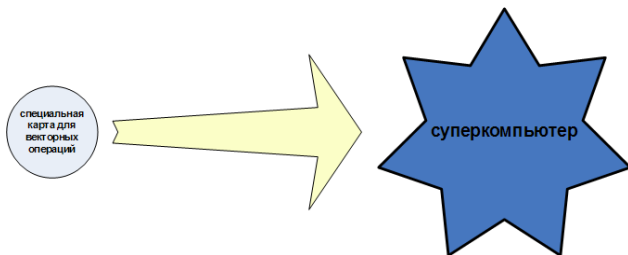
Что?

# Почему так произошло?



Что?

# Почему так произошло?



- от специализированных ядер с арифметикой низкой точности к унифицированным ядрам с арифметикой "высокой" точности

# Не все так прекрасно...

- GPU все-таки ориентирован на графику
- Архитектура GPU
  - параллельная обработка данных
  - очень быстро меняется
  - секретна
  - сильно отличается у разных производителей
- нельзя просто взять и портировать CPU-код, надо менять алгоритмы
- решение новой задачи крайне сложный уникальный процесс с трудно предсказуемым результатом

# Не все так прекрасно...

- GPU все-таки ориентирован на графику
- Архитектура GPU
  - параллельная обработка данных
  - очень быстро меняется
  - секретна
  - сильно отличается у разных производителей
- нельзя просто взять и портировать CPU-код, надо менять алгоритмы
- решение новой задачи крайне сложный уникальный процесс с трудно предсказуемым результатом

## Не все так прекрасно...

- GPU все-таки ориентирован на графику
- Архитектура GPU
  - параллельная обработка данных
  - очень быстро меняется
  - секретна
  - сильно отличается у разных производителей
- нельзя просто взять и портировать CPU-код, надо менять алгоритмы
- решение новой задачи крайне сложный уникальный процесс с трудно предсказуемым результатом

## Не все так прекрасно...

- GPU все-таки ориентирован на графику
- Архитектура GPU
  - параллельная обработка данных
  - очень быстро меняется
  - секретна
  - сильно отличается у разных производителей
- нельзя просто взять и портировать CPU-код, надо менять алгоритмы
- решение новой задачи крайне сложный уникальный процесс с трудно предсказуемым результатом

## Не все так прекрасно...

- GPU все-таки ориентирован на графику
- Архитектура GPU
  - параллельная обработка данных
  - очень быстро меняется
  - секретна
  - сильно отличается у разных производителей
- нельзя просто взять и портировать CPU-код, надо менять алгоритмы
- решение новой задачи крайне сложный уникальный процесс с трудно предсказуемым результатом

# Структура курса (примерно)

- Графические процессоры: архитектура, возможности
  - обзор текущей ситуации на рынке
- API: DirectX, HLSL, Cg, специализированные GPGPU языки (OpenCL, DirectCompute, ...)
  - обзор текущей ситуации на рынке
- Алгоритмы и структуры данных
  - хранение разреженных матриц
  - перемножение матриц
  - сортировки
  - моделирования взаимодействия система твердых тел



# Структура курса (примерно)

- Графические процессоры: архитектура, возможности
  - обзор текущей ситуации на рынке
- API: DirectX, HLSL, Cg, специализированные GPGPU языки (OpenCL, DirectCompute, ...)
  - обзор текущей ситуации на рынке
- Алгоритмы и структуры данных
  - хранение разреженных матриц
  - перемножение матриц
  - сортировки
  - моделирования взаимодействия система твердых тел

# Структура курса (примерно)

- Графические процессоры: архитектура, возможности
  - обзор текущей ситуации на рынке
- API: DirectX, HLSL, Cg, специализированные GPGPU языки (OpenCL, DirectCompute, ...)
  - обзор текущей ситуации на рынке
- Алгоритмы и структуры данных
  - хранение разреженных матриц
  - перемножение матриц
  - сортировки
  - моделирования взаимодействия система твердых тел

# Структура курса (примерно)

- Графические процессоры: архитектура, возможности
  - обзор текущей ситуации на рынке
- API: DirectX, HLSL, Cg, специализированные GPGPU языки (OpenCL, DirectCompute, ...)
  - обзор текущей ситуации на рынке
- Алгоритмы и структуры данных
  - хранение разреженных матриц
  - перемножение матриц
  - сортировки
  - моделирования взаимодействия система твердых тел

# Содержание

- 1 Лекции
  - Что?
  - Почему?
  - Зачем?
  
- 2 Лабораторные
  - Что используем
  - Что делаем

# Почему курс читают вам?

- Вы специалисты по численным методам
- Вас применяют в моделирования сложных процессов
- Многие ваши задачи хорошо подходят для распараллеливания
- Исторически кафедра питает любовь к GPU, накоплены знания
- Вам также читается курс GPU-ориентированной "быстрой" компьютерной графики
- Вы - идеальная кандидатура!!!

# Почему курс читают вам?

- Вы специалисты по численным методам
- Вас применяют в моделирования сложных процессов
- Многие ваши задачи хорошо подходят для распараллеливания
- Исторически кафедра питает любовь к GPU, накоплены знания
- Вам также читается курс GPU-ориентированной "быстрой" компьютерной графики
- Вы - идеальная кандидатура!!!

# Почему курс читают вам?

- Вы специалисты по численным методам
- Вас применяют в моделирования сложных процессов
- Многие ваши задачи хорошо подходят для распараллеливания
- Исторически кафедра питает любовь к GPU, накоплены знания
- Вам также читается курс GPU-ориентированной "быстрой" компьютерной графики
- Вы - идеальная кандидатура!!!

# Почему курс читают вам?

- Вы специалисты по численным методам
- Вас применяют в моделирования сложных процессов
- Многие ваши задачи хорошо подходят для распараллеливания
- Исторически кафедра питает любовь к GPU, накоплены знания
- Вам также читается курс GPU-ориентированной "быстрой" компьютерной графики
- Вы - идеальная кандидатура!!!



# Почему курс читают вам?

- Вы специалисты по численным методам
- Вас применяют в моделирования сложных процессов
- Многие ваши задачи хорошо подходят для распараллеливания
- Исторически кафедра питает любовь к GPU, накоплены знания
- Вам также читается курс GPU-ориентированной "быстрой" компьютерной графики
- Вы - идеальная кандидатура!!!

# Почему курс читают вам?

- Вы специалисты по численным методам
- Вас применяют в моделирования сложных процессов
- Многие ваши задачи хорошо подходят для распараллеливания
- Исторически кафедра питает любовь к GPU, накоплены знания
- Вам также читается курс GPU-ориентированной "быстрой" компьютерной графики
- Вы - идеальная кандидатура!!!

# Почему курс читают вам?

- Вы специалисты по численным методам
- Вас применяют в моделирования сложных процессов
- Многие ваши задачи хорошо подходят для распараллеливания
- Исторически кафедра питает любовь к GPU, накоплены знания
- Вам также читается курс GPU-ориентированной "быстрой" компьютерной графики
- **Вы - идеальная кандидатура!!!**

# Содержание

## 1 Лекции

- Что?
- Почему?
- Зачем?

## 2 Лабораторные

- Что используем
- Что делаем

# А каковы бонусы

- GPGPU - это современная мировая тенденция
- до сих пор (с 2006) CPU нечем ответить
  - Cell (Sony, Toshiba, IBM) <sup>2</sup> , Larrabee (Intel) <sup>3</sup>
- математики-программисты-GPU уникальны
- широкий спектр задач вашей специальности подходит для GPGPU
  - уравнение теплопроводности
  - уравнение Навье-Стокса
  - метод конечных элементов
  - вообще матричные вычисления

---

<sup>2</sup>[http://en.wikipedia.org/wiki/Cell\\_\(microprocessor\)](http://en.wikipedia.org/wiki/Cell_(microprocessor))

<sup>3</sup>[http://en.wikipedia.org/wiki/Larrabee\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Larrabee_(microarchitecture))

# А каковы бонусы

- GPGPU - это современная мировая тенденция
- до сих пор (с 2006) CPU нечем ответить
  - Cell (Sony, Toshiba, IBM) <sup>2</sup> , Larrabee (Intel) <sup>3</sup>
- математики-программисты-GPU уникальны
- широкий спектр задач вашей специальности подходит для GPGPU
  - уравнение теплопроводности
  - уравнение Навье-Стокса
  - метод конечных элементов
  - вообще матричные вычисления

---

<sup>2</sup>[http://en.wikipedia.org/wiki/Cell\\_\(microprocessor\)](http://en.wikipedia.org/wiki/Cell_(microprocessor))

<sup>3</sup>[http://en.wikipedia.org/wiki/Larrabee\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Larrabee_(microarchitecture))

# А каковы бонусы

- GPGPU - это современная мировая тенденция
- до сих пор (с 2006) CPU нечем ответить
  - Cell (Sony, Toshiba, IBM) <sup>2</sup> , Larrabee (Intel) <sup>3</sup>
- математики-программисты-GPU уникальны
- широкий спектр задач вашей специальности подходит для GPGPU
  - уравнение теплопроводности
  - уравнение Навье-Стокса
  - метод конечных элементов
  - вообще матричные вычисления

---

<sup>2</sup>[http://en.wikipedia.org/wiki/Cell\\_\(microprocessor\)](http://en.wikipedia.org/wiki/Cell_(microprocessor))

<sup>3</sup>[http://en.wikipedia.org/wiki/Larrabee\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Larrabee_(microarchitecture))

# А каковы бонусы

- GPGPU - это современная мировая тенденция
- до сих пор (с 2006) CPU нечем ответить
  - Cell (Sony, Toshiba, IBM) <sup>2</sup> , Larrabee (Intel) <sup>3</sup>
- математики-программисты-GPU уникальны
- широкий спектр задач вашей специальности подходит для GPGPU
  - уравнение теплопроводности
  - уравнение Навье-Стокса
  - метод конечных элементов
  - вообще матричные вычисления

---

<sup>2</sup>[http://en.wikipedia.org/wiki/Cell\\_\(microprocessor\)](http://en.wikipedia.org/wiki/Cell_(microprocessor))

<sup>3</sup>[http://en.wikipedia.org/wiki/Larrabee\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Larrabee_(microarchitecture))



# А каковы бонусы

- GPGPU - это современная мировая тенденция
- до сих пор (с 2006) CPU нечем ответить
  - Cell (Sony, Toshiba, IBM) <sup>2</sup> , Larrabee (Intel) <sup>3</sup>
- математики-программисты-GPU уникальны
- широкий спектр задач вашей специальности подходит для GPGPU
  - уравнение теплопроводности
  - уравнение Навье-Стокса
  - метод конечных элементов
  - вообще матричные вычисления

---

<sup>2</sup>[http://en.wikipedia.org/wiki/Cell\\_\(microprocessor\)](http://en.wikipedia.org/wiki/Cell_(microprocessor))

<sup>3</sup>[http://en.wikipedia.org/wiki/Larrabee\\_\(microarchitecture\)](http://en.wikipedia.org/wiki/Larrabee_(microarchitecture))

# Содержание

- 1 Лекции
  - Что?
  - Почему?
  - Зачем?
  
- 2 Лабораторные
  - Что используем
  - Что делаем

# Содержание

- 1 Лекции
  - Что?
  - Почему?
  - Зачем?
  
- 2 Лабораторные
  - Что используем
  - Что делаем

- среда программирования: Visual Studio (Express) <sup>4</sup>
- язык программирования: C/C++
- API: OpenCL <sup>5 6</sup>
  - библиотека для разработки приложений для параллельных вычислений на GPU
  - имеет специальный язык для написания kernel
  - имеет API для управления GPU
  - поддерживается NVidia, ATI(AMD), Intel, ARM

---

<sup>4</sup><http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>

<sup>5</sup><http://www.khronos.org/>

<sup>6</sup><http://en.wikipedia.org/wiki/OpenCL>

- среда программирования: Visual Studio (Express) <sup>4</sup>
- язык программирования: C/C++
- API: OpenCL <sup>5 6</sup>
  - библиотека для разработки приложений для параллельных вычислений на GPU
  - имеет специальный язык для написания kernel
  - имеет API для управления GPU
  - поддерживана NVidia, ATI(AMD), Intel, ARM

---

<sup>4</sup><http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>

<sup>5</sup><http://www.khronos.org/>

<sup>6</sup><http://en.wikipedia.org/wiki/OpenCL>

- среда программирования: Visual Studio (Express) <sup>4</sup>
- язык программирования: C/C++
- API: OpenCL <sup>5 6</sup>
  - библиотека для разработки приложений для параллельных вычислений на GPU
  - имеет специальный язык для написания kernel
  - имеет API для управления GPU
  - поддерживается NVidia, ATI(AMD), Intel, ARM

---

<sup>4</sup><http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>

<sup>5</sup><http://www.khronos.org/>

<sup>6</sup><http://en.wikipedia.org/wiki/OpenCL>

- среда программирования: Visual Studio (Express) <sup>4</sup>
- язык программирования: C/C++
- API: OpenCL <sup>5 6</sup>
  - библиотека для разработки приложений для параллельных вычислений на GPU
  - имеет специальный язык для написания kernel
  - имеет API для управления GPU
  - поддерживается NVidia, ATI(AMD), Intel, ARM

---

<sup>4</sup><http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>

<sup>5</sup><http://www.khronos.org/>

<sup>6</sup><http://en.wikipedia.org/wiki/OpenCL>

# Содержание

- 1 Лекции
  - Что?
  - Почему?
  - Зачем?
  
- 2 Лабораторные
  - Что используем
  - Что делаем



## Задачи:

- ① установить соответствующую вашей платформе реализацию OpenCL, позапустить примеры.
- ② выбрать наиболее симпатичный, поиграть с ним и написать отчет:
  - постановка задачи
  - организация данных, реализация алгоритма
  - график зависимости скорости вычислений от объема данных
- ③ решить задачу на нахождение пар пересекающихся объектов<sup>7</sup>
  - пересекающиеся отрезки
  - пересекающиеся окружности

---

<sup>7</sup> детали обсудим позже

## Задачи:

- ❶ установить соответствующую вашей платформе реализацию OpenCL, позапустить примеры.
- ❷ выбрать наиболее симпатичный, поиграть с ним и написать отчет:
  - постановка задачи
  - организация данных, реализация алгоритма
  - график зависимости скорости вычислений от объема данных
- ❸ решить задачу на нахождение пар пересекающихся объектов<sup>7</sup>
  - пересекающиеся отрезки
  - пересекающиеся окружности

---

<sup>7</sup> детали обсудим позже

## Задачи:

- ❶ установить соответствующую вашей платформе реализацию OpenCL, позапустить примеры.
- ❷ выбрать наиболее симпатичный, поиграть с ним и написать отчет:
  - постановка задачи
  - организация данных, реализация алгоритма
  - график зависимости скорости вычислений от объема данных
- ❸ решить задачу на нахождение пар пересекающихся объектов<sup>7</sup>
  - пересекающиеся отрезки
  - пересекающиеся окружности

---

<sup>7</sup> детали обсудим позже

## Задачи:

- ❶ установить соответствующую вашей платформе реализацию OpenCL, позапустить примеры.
- ❷ выбрать наиболее симпатичный, поиграть с ним и написать отчет:
  - постановка задачи
  - организация данных, реализация алгоритма
  - график зависимости скорости вычислений от объема данных
- ❸ решить задачу на нахождение пар пересекающихся объектов<sup>7</sup>
  - пересекающиеся отрезки
  - пересекающиеся окружности

---

<sup>7</sup> детали обсудим позже

ПРОДОЛЖЕНИЕ СЛЕДУЕТ...