



Казанский
федеральный
университет

ВЫСШАЯ ШКОЛА
информационных технологий
и информационных систем

Библиотеки OpenCL

Библиотеки OpenCL

- ▶ Как и для CUDA для OpenCL существует множество специализированных библиотек
 - STL-style: Boost.Compute
 - Линейная алгебра: clBLAS, Eigen, ViennaCL, VexCL
 - FFT: clFFt, VexCL
 - CV и машинное обучение: OpenCV



Boost.Compute

- ▶ Библиотека заголовков для работы с многоядерными CPU и GPU
- ▶ Основана на OpenCL
- ▶ <https://github.com/boostorg/compute>
- ▶ При компиляции кода необходимо указать
 - -I/путь к папке с boost/
- ▶ Требуемая версия Boost ≥ 1.54
- ▶ Включен в Boost начиная с версии 1.61



Boost.Compute

- ▶ Предоставляет C++ обертку над API OpenCL
- ▶ Содержит набор стандартных контейнеров, алгоритмов и итераторов
- ▶ Дает возможность сочетать высокоуровневые абстракции объектов с пользовательскими ядрами



Boost.Compute

- ▶ Пример: обход всех доступных платформ и устройств



Boost.Compute

- ▶ Идеология применения библиотеки схожа с STL – операции над контейнерами данных
- ▶ Boost.Compute позволяет писать C++ код не используя C-синтаксис OpenGL
- ▶ Список алгоритмов:
 - http://www.boost.org/doc/libs/1_62_0/libs/compute/doc/html/boost_compute/reference.html#boost_compute.reference.api_overview.algorithms



Boost.Compute

- ▶ Пример: передача данных с хоста на устройство посредством Boost.Compute
- ▶ Обработка данных на устройстве через функцию `transform()`
- ▶ Передача обработанных данных с устройства на хост



Boost.Compute

- ▶ Библиотека предоставляет возможность создавать пользовательские функции, для передачи в алгоритмы типа `transform()` и `reduce()`
- ▶ Boost.Compute поддерживает лямбда-выражения для передачи в соответствующие алгоритмы в качестве параметра



Boost.Compute

- ▶ Пример: использования собственных функций и лямбда-выражений для передачи в `transform()`



VexCL

- ▶ Шаблонная библиотека векторных выражений
- ▶ Предназначена для облегчения разработки кода с использованием GPU
- ▶ В качестве бэкенда используются OpenCL/CUDA
- ▶ Поддержка вычислений на нескольких устройствах и платформах



VexCL

- ▶ Библиотека предоставляет STL-подобный интерфейс для работы с контейнерами и алгоритмами
- ▶ [Документация библиотеки](#)



VexCL

- ▶ Поставить boost (версия > 1.61)
- ▶ Скачать [VexCL](#)
- ▶ Сконфигурировать VexCL
 - `cd /path_to_vexcl_directory/`
 - `cmake . -Bbuild`



VexCL

- ▶ `cd /path_to_code_directory/`
- ▶ Создать код с использованием VexCL
- ▶ Создать CMakeLists.txt по образцу
- ▶ Собрать свой код
 - `mkdir build`
 - `cd build`
 - `cmake ..`
 - `make`



VexCL

- ▶ Пример: получение списка GPU поддерживающих вычисления в двойной точностью и включение их в текущий контекст



VexCL

- ▶ Основной фокус библиотеки – операции над векторами
 - Векторы должны иметь одинаковый размер
 - Должны размещаться в памяти одного и того же устройства
- ▶ Каждая операция над векторами приводит к запуску вычислительного ядра



VexCL

- ▶ Пример сложения векторов
- ▶ Пример создания пользовательской функции
- ▶ Пример сортировки



Задание на практику

- Редукция с использованием библиотеки VexCL или Boost.Compute