

# Высокопроизводительные вычисления на кластерах с использованием графических ускорителей NVIDIA

17 июня 2012 г.

**Руководитель трека:** Андрей Комиссаров, [akomissarov@nvidia.com](mailto:akomissarov@nvidia.com), +79166853455

**Максимальное количество участников:** 15

**Краткая аннотация.** Вашему вниманию предлагается углубленный практический курс по технологии NVIDIA CUDA для разработчиков и исследователей, применяющих параллельные вычисления. В первой части последовательно излагаются основы программной модели CUDA применительно к языкам C и Fortran, сведения о типах памяти GPU и методы эффективного использования разделяемой памяти на примере некоторых алгоритмов. Затем даётся обзор основных прикладных библиотек и языковых средств со встроенной поддержкой вычислений на GPU. Отдельные лекции посвящены элементам профессиональной разработки – средствам анализа, отладки и диагностики. Рассмотрены методы управления несколькими GPU на рабочих станциях и распределенных кластерных системах. Курс проводится специалистами учебного центра Applied Parallel Computing совместно с компанией NVIDIA.

**Количество лекций:** 12

**Количество практических занятий:** 17

**Количество часов самостоятельной работы:** на каждый день предусмотрено задание для самостоятельной работы на 1 час (в программе трека помечены *курси-вом*).

Таблица 1: Расписание трека

N	Название	Преподаватели
<b>1 июля. Вводная часть.</b>		
Л1	Системы с графическими ускорителями в Российских учебно-научных центрах.	NVIDIA
Л2	Принципы работы графических ускорителей, программная модель CUDA.	Евгений Перепёлкин (APC)
П1	Вычислительный комплекс с GPU: получение информации о системе, средства мониторинга и диагностики, среды разработки и исполнения приложений.	Дмитрий Нгуен, Дмитрий Телегин (APC)
П2	Создание простейших программ для GPU: <code>sum_kernel</code> , <code>sine_calc</code> .	Андрей Сафронов (APC)
Л3	Устройство CUDA-компилятора: стадии обработки кода, промежуточные представления, загрузка ядер, JIT-компиляция. <i>Задание для самостоятельной работы: видимость глобальных данных в CUDA.</i>	Андрей Сафронов (APC)
<b>2 июля. Эффективные алгоритмы и быстрая разработка.</b>		
Л4	Иерархия памяти CUDA, эффективное использование разделяемой памяти. Общее виртуальное адресное пространство (UVA).	Евгений Перепёлкин (APC)
Л5	Разработка CUDA-приложений на языке Fortran. <code>ISO_C_BINDING</code> , замечания о способе передачи аргументов.	Дмитрий Нгуен (APC)
П3	Эффективная реализация редукции или перемножения плотных матриц с использованием разделяемой памяти и UVA.	Дмитрий Телегин (APC)
Л6	Быстрая разработка CUDA-приложений на C++ с помощью Thrust. Библиотека алгоритмов линейной алгебры с разреженными матрицами CUSP. Использование в C и Fortran.	Дмитрий Телегин, Александр Шевченко (APC)
П4	Реализации алгоритмов <code>saхру</code> и сортировки пар (ключ, значение) с помощью Thrust. <i>Задание для самостоятельной работы: решение 5-диагональной линейной системы с помощью CUSP.</i>	Дмитрий Телегин, Александр Шевченко (APC)
<b>3 июля. Прикладные библиотеки.</b>		

Л7	Прикладные библиотеки со встроенной поддержкой GPU, часть I: CUBLAS, MAGMA, CUSPARSE, CUFFT, CURAND.	Александр Шевченко (АПС)
П5	Реализация степенного метода поиска наибольшего собственного значения плотной или разреженной матрицы.	Александр Шевченко (АПС)
П6	Реализация метода покоординатного расщепления с помощью CUFFT и прогонки для задачи Дирихле с границами различных типов. <i>Самостоятельная работа.</i>	Александр Шевченко (АПС)
Л8	Прикладные библиотеки со встроенной поддержкой GPU, часть II: PetSc, Trilinos.	Олег Рябков, Дмитрий Голицын, Дмитрий Буров (МГУ)
П7	Решение уравнения Пуассона с помощью PetSc и Trilinos.	Олег Рябков, Дмитрий Голицын, Дмитрий Буров, Олег Рябков (МГУ)
<b>4 июля. MultiGPU.</b>		
Л9	Асинхронное исполнение, CUDA Streams. Измерение времени, CUDA Events. Управление несколькими GPU: взаимодействие CUDA с другими программными моделями параллельных вычислений.	Дмитрий Телегин (АПС)
П8	Реализация конкурентного исполнения нескольких ядер на GPU с промежуточными синхронизациями. Асинхронные и блокирующие операции.	Дмитрий Телегин (АПС)
П9	Параллельное использование нескольких GPU в последовательном приложении: serial_cuda. Несколько GPU в многопоточном приложении на основе интерфейса POSIX: pthreads_cuda_p2p. <i>Задание для самостоятельной работы: использование нескольких GPU в многопоточном приложении, управляемом директивами OpenMP.</i>	Александр Шевченко (АПС)
П10	Расширения OpenMPI для CUDA, обмен данными в памяти GPU с помощью MPI: mpi_cuda_sendrecv.	Александр Шевченко (АПС)

П11	Реализация взаимодействия между несколькими CUDA-приложениями с помощью интерфейса IPC.	Дмитрий Телегин (APC)
<b>5 июля. Анализ, диагностика, отладка.</b>		
Л10	Средства анализа, диагностики и отладки CUDA-приложений. Профилировка с помощью CUDA Profiler, диагностика ошибок памяти (cuda-memcheck), интерактивная отладка GPU-ядер (cuda-gdb).	Дмитрий Телегин (APC)
П12	Анализ эффективности приложения с помощью CUDA Profiler. Основные аппаратные счётчики.	Дмитрий Телегин (APC)
П13	Демонстрация работы отладчика: основные возможности, стандартные сценарии использования. Типичные ошибки в приложениях.	Дмитрий Телегин (APC)
П14	Программируемая профилировка с помощью CUPTI.	Дмитрий Телегин (APC)
<b>6 июля. Оптимизация программ, архитектура и внутреннее устройство GPU.</b>		
Л11	Архитектура GPU, методы анализа и оптимизации приложений.	Кто к нам придёт сюда из девтеков NVIDIA?
П16	Управление кэшем GPU, эффект на производительность при различных шаблонах доступа к памяти. <i>Задание для самостоятельной работы: кеширование данных в текстурной памяти.</i>	Николай Лихогруд (APC)
Л12	Язык промежуточного представления программы PTX и Fermi ISA. Формат исполняемого образа ядра CUBIN, начальная загрузка. Ассемблер и дизассемблер.	Николай Лихогруд (APC)
П17	Анализ эффективности компилятора на низком уровне: распределение регистров, локальная память, векторизация. Отладка GPU-программы без исходного кода.	Николай Лихогруд (APC)
+	Заключение: вопросы и пожелания, обсуждение перспективных совместных проектов. Собеседование по результатам выполнения практических заданий.	

**Требования к знаниям и умениям потенциальных участников**

- Способности к интенсивному обучению и самообучению: увлечённость задачами, умение разделять проблему на отдельные вопросы и умение быстро и самостоятельно находить ответы на большую их часть.
- Владение языками программирования C, C++ или Fortran.
- Навыки эффективной работы в консольной среде UNIX (текстовые редакторы, поиск, компиляторы и системы сборки приложений).
- Технический английский для самостоятельного изучения документации и научных статей.

**Количество преподавателей:** около 5, для всех необходимы пропуска

**Возможность полной/частичной публикации материалов:** возможна полная публикация