Курс «Гибридные вычислительные системы»

ст. преп. каф. ВпВ ИКИТ СФУ Тарасов С. А.

Аннотация

Курс «Гибридные вычислительные системы» посвящен гетерогенным вычислениям (Heterogeneous Computing): рассматриваются общие аспекты гетерогенных вычислений; особое внимание уделяется технологии CUDA (GPGPU). Продолжительность — 2 семестра; формы оценивания: зачет, экзамен.

Гетерогенные вычисления

Гетерогенные вычисления — это парадигма вычислений, в рамках которой для решения задач используются процессоры разных архитектур, каждый из которых оптимизирован для определённых типов операций. В отличие от гомогенных систем (например, только CPU), гетерогенные системы комбинируют разнородные вычислительные устройства в рамках одной платформы. Классическим примером HC-системы является связка CPU + GPU.

Содержание курса

- Гетерогенные вычисления (обзор)
- Обзор архитектуры NVIDIA Tesla GPU
- Модель программирования CUDA
- CUDA Toolchain: компиляция, линковка и дистрибуция
- Тестирование
- Точность вычислений и вычисления низкой точности
- Проектирование гетерогенных вычислений/паттерны проектирования
- Оценка производительности и оптимизация CUDA kernels
- Математика глубокого обучения: автоматическое дифференцирование и обратное распространение ошибки
- Реализация моделей глубокого обучения на базе CUDA

Практика

Практическая часть курса состоит из 10 работ, которые охватывают основные составляющие разработки на CUDA; вот некоторые из них:

- написание и интеграция GPU-кода (kernels)
- параллельное программирование и конфигурация сетки нитей

- использование различных типов памяти (shared, global, texture и др.)
- программирование тензорных ядер
- внутриварповый обмен
- работа с CUDA Streams & Graphs
- оптимизация kernels (register pressure, warp divergence, occupancy и др.)

Каждая работа имеет базовую структуру:

- разработка классов и kernels
- интеграция kernels в C++-код
- написание тестов и тестирование
- написание бенчмарков и оценка производительности
- анализ полученных оценок производительности

Все работы объединены тематикой глубокого обучения. В результате успешного прохождения практической части курса студент получит навык реализации моделей глубокого обучения в фреймворке CUDA + PyTorch.

Задание каждой работы составлено таким образом, чтобы в результате ее выполнения студент получил программную библиотеку, которая будет использоваться в следующих работах.

К каждой работе прилагаются критерии ее оценивания.

Ознакомиться с заданиями можно по ссылке: https.

Технологический стек

- Аппаратное обеспечение: NVIDIA Tesla GPU (Tesla T4)
- CUDA Toolkit: NVCC, Nsight Compute, cuda-gdb, cuobjdump, Compute Sanitizer
- Языки программирования: CUDA C++, C++ (std=C++20), NVIDIA PTX, Python
- Библиотеки: CUDA Runtime API, CUDNN, Google Benchmark, Google Test, unittest/PyTest, Eigen, LibTorch, PyTorch, PyBind11
- Сборка: CMake + Ninja
- Компиляторы: NVCC, Clang++/G++
- Git/GitHub
- Google Colab
- IDE: clangd, clang-tidy, clang-format
- Паттерны проектирования: RAII, Data + View, Policy/Strategy, Expression Templates, CRTP и др.