

Introduction CUDA Programming (CS131)

A presentation on CUDA parallel programming, featuring examples, some research materials, and demonstrations of GPU acceleration.

Overview

This project provides an introduction for understanding NVIDIA's CUDA (Compute Unified Device Architecture) platform. It includes code examples, Jupyter notebooks, documentation, and slides that demonstrate the massive performance improvements possible with GPU-accelerated parallel computing.

What is CUDA?

CUDA is NVIDIA's parallel computing platform and programming model that enables developers to harness the power of GPUs for general-purpose computing. While CPUs excel at sequential tasks with complex branching, GPUs shine when performing the same operation across thousands of data points simultaneously - making them ideal for scientific computing, machine learning, image processing, and mathematical simulations.

Project Contents

Exercises and Examples

E01 - Hello World (Vector Addition)

- Introduction to CUDA kernel functions
- Basic memory management (`cudaMalloc` , `cudaMemcpy` , `cudaFree`)
- Thread ID calculation and kernel launch syntax
- CUDA event timing and error handling
- Adds 1 million elements in parallel on the GPU

Eo2 - Matrix Multiplication

- CPU vs GPU performance comparison
- 2D thread organization with blocks and grids
- Demonstrates 50-500x speedup on typical hardware
- Result verification and timing breakdown

Eo3 - Mandelbrot Set // not included in video presentation

- fractal visualization
- CPU vs GPU speed comparison

Eo4 - Julia Set

- Fractal visualization exercise
- CUDA parallel image generation
- comparison between GPU and CPU speeds

Jupyter Notebooks (`src/`)

Interactive notebooks for hands-on learning using Google Colab:

- `Example01_hello.ipynb` - Vector addition walkthrough
- `Example02_vector.ipynb` - Vector operations
- `example03_mandelbrot.ipynb` - Mandelbrot Set generation // not used in this presentation
- `example04_juliaset.ipynb` - Julia Set visualization

Research Documentation (Research/)

covers:

1. **What is CUDA** - Introduction to GPU computing and GPGPU concepts
2. **CUDA Programming Model** - Host/device architecture, kernels, thread hierarchy, memory model
3. **Basic CUDA Programming** - Syntax, memory management, kernel launches
4. **Performance Optimization** - SIMD execution, memory patterns, shared memory, synchronization
5. **Advanced Topics** - CUDA libraries, streams, error handling, debugging and profiling

Prerequisites

- **NVIDIA GPU** with CUDA compute capability (check compatibility at developer.nvidia.com)
- **CUDA Toolkit** - Download from NVIDIA (includes nvcc compiler)

- **C/C++ Compiler** - GCC on Linux, Visual Studio on Windows
- **Optional:** Jupyter Notebook for interactive examples
- **Optional:** ImageMagick or GIMP for viewing PPM image outputs

Key CUDA Concepts Demonstrated

Thread Organization

- Hierarchical structure: Grids → Blocks → Threads
- Thread ID calculation: `blockIdx.x * blockDim.x + threadIdx.x`
- 2D/3D thread organization for matrix and image operations

Memory Management

- Host (CPU) and Device (GPU) memory allocation
- Data transfer between host and device
- Global, shared, constant, and local memory types

Kernel Functions

- `__global__` keyword for GPU kernels
- Kernel launch syntax: `kernel<<<blocks, threads>>>(args)`
- Thread synchronization with `cudaDeviceSynchronize()`

Performance Optimization

- Coalesced memory access patterns
- Shared memory utilization
- Occupancy maximization

- Timing with CUDA events

Performance

Based on typical hardware (e.g., NVIDIA Tesla T4, RTX series):

Example	CPU Performance	GPU Performance	Typical Speedup
Vector Addition (1M elements)	~10ms	~1ms	10-20x
Matrix Multiplication (2048 ²)	Several seconds	~50-100ms	50-100x
Mandelbrot Set (1920x1080)	0.5-3 FPS	30-300 FPS	50-500x

Additional Resources

- [NVIDIA CUDA Documentation](#)
- [CUDA Programming Guide](#)
- [CUDA Best Practices Guide](#)
- [CUDA Samples Repository](#)

Author

dlockwood - CS131 Course Project

https://github.com/dlockwood0281/dlockwood_cs131/

Accompanying video

<https://youtu.be/tuB4ggOZW-8>