

# GPU programming Projects



**Politecnico  
di Torino**

Dipartimento  
di Automatica e Informatica

**Docente: Luca Sterpone**  
**luca.sterpone@polito.it**

**Esercitori: Corrado De Sio**  
**corrado.desio@polito.it**  
**Josie Esteban Rodriguez Condia**  
**josie.rodriquez@polito.it**

# Projects

2

- The purpose of the project is to apply data-parallelism and CUDA concepts
  
- Examples:
  - ▣ Thorough performance analysis and improvement of GPU code or architectures
  - ▣ Implement a GPU version of some existing computationally-intensive CPU code
  - ▣ Reproduce some existing GPU research work
  - ▣ Do novel GPU research

# An Example of Project Outline

3

Step	Broad Outline	Concrete Example
1.	Choose an application.	Dense Matrix-Matrix Multiply
2.	Determine what part of the application is taking the majority of the time.	Algorithm Evaluation through Analysis/Simulation
3.	Determine one or more data-parallel approaches to solving the problem.	Assign one output to each thread in a gather-style approach.
4.	Create multiple implementations of the approach.	One naïve version, one version with shared memory tiling, one version with register tiling.
5.	Measure the performance and execution characteristics of the implementations for various parameters	Record memory transfer time, kernel time, utilization, FLOPS, etc.
6.	Relate results to course concepts	Performance may be impacted by utilization, shared-memory accesses, memory coalescing, and control divergence.

# What is Evaluated

4

- **Presentation and Report**
  - Covers the required sections.
  - Technically sound, clear, organized, and well-written.
  - Shows good in-depth analysis of the problem, the approach, and the results.
  - No mandatory min-max number of pages. Don't be verbose, don't be superficial.
  - Ideal is 15 pages with meaningful pictures.
- **Functionality and Performance:**
  - Produces correct results
  - Achieves good speedup relative to base code and/or competitors
  - Optimizations applied
- **Code quality:**
  - Coding style is neat
  - Code is well documented

# Ideas (I)

5

- **Physical Simulations:**
  - E.g. – Particles or bodies interaction simulation
- **Fault Tolerance and Reliability**
  - E.g. – Error correction, redundancy, coherence
- **Games**
  - E.g. - Solving, Decision Taking
- **Artificial Intelligence and Machine Learning**
  - E.g. - Neural Networks

# Ideas (II)

6

- Famous Computational Challenges or Algorithms:
  - E.g. - Mathematical Algorithms, NP-problems
- Data Science
  - E.g. – Big Data Management, Pattern Detection
- Cybersecurity:
  - E.g. – Attacks, Hashing, Ciphers
- Computer Vision
  - E.g. – Video and image processing, segmentation, object detection, classification

# Ideas (III)

7

- Research and Investigation
  - E.g. – Hardware and Software Observability
  - E.g. – Models for Emulation / Simulation
  - E.g. – Reliability and Fault Tolerance analysis and technique