Welcome to 15-779: Advanced ML Systems (LLM Edition)

Zhihao Jia

Computer Science Department

Carnegie Mellon University

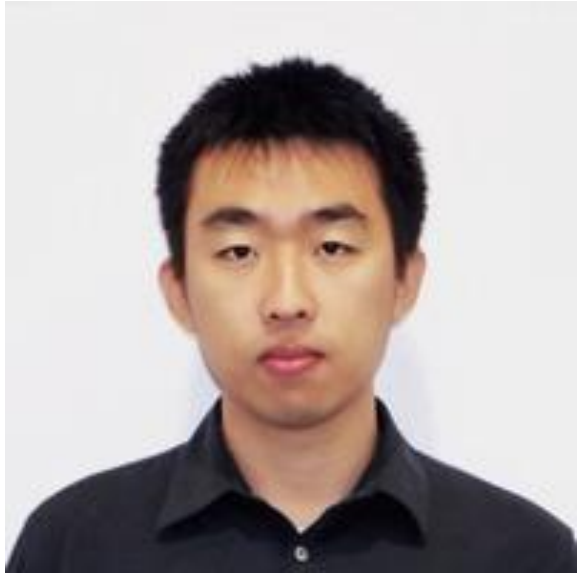
Course Information

- **Website**: https://www.cs.cmu.edu/~zhihaoj2/15-779/
  - Contains links to all resources

- **Piazza**: discussions and announcements

- **Gradscope**: submit assignments, project proposals, final papers

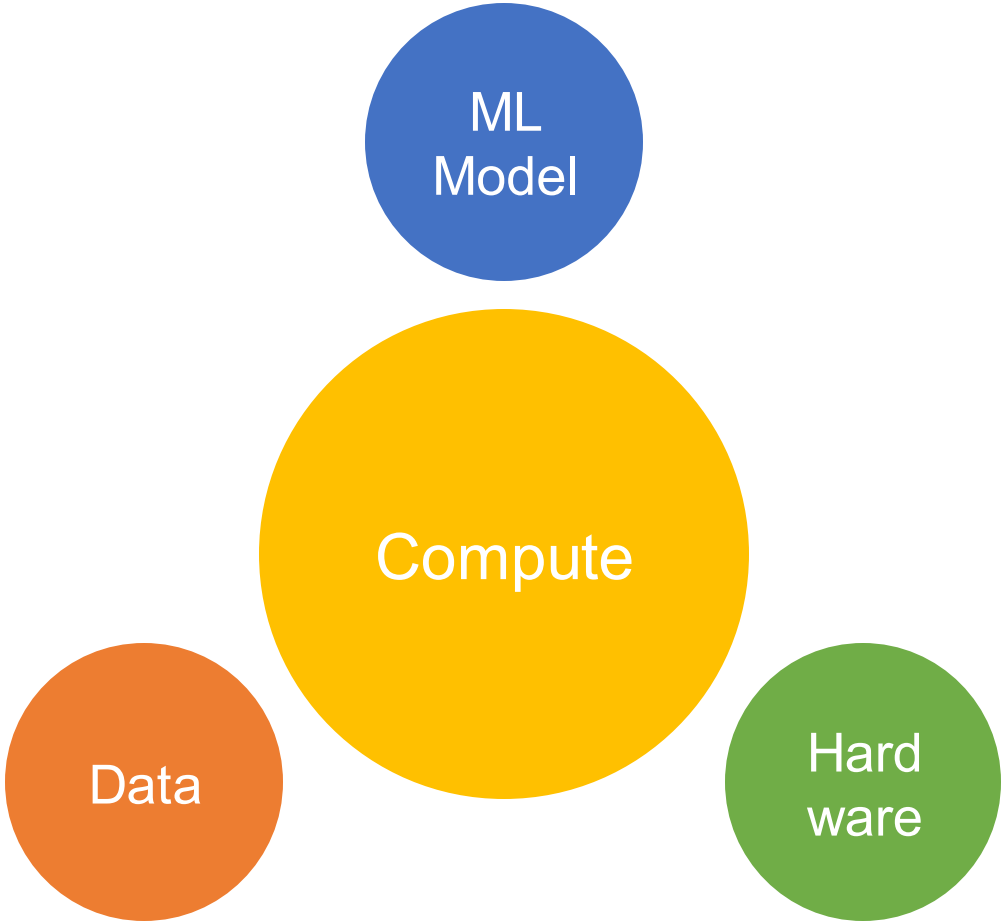
Instructors



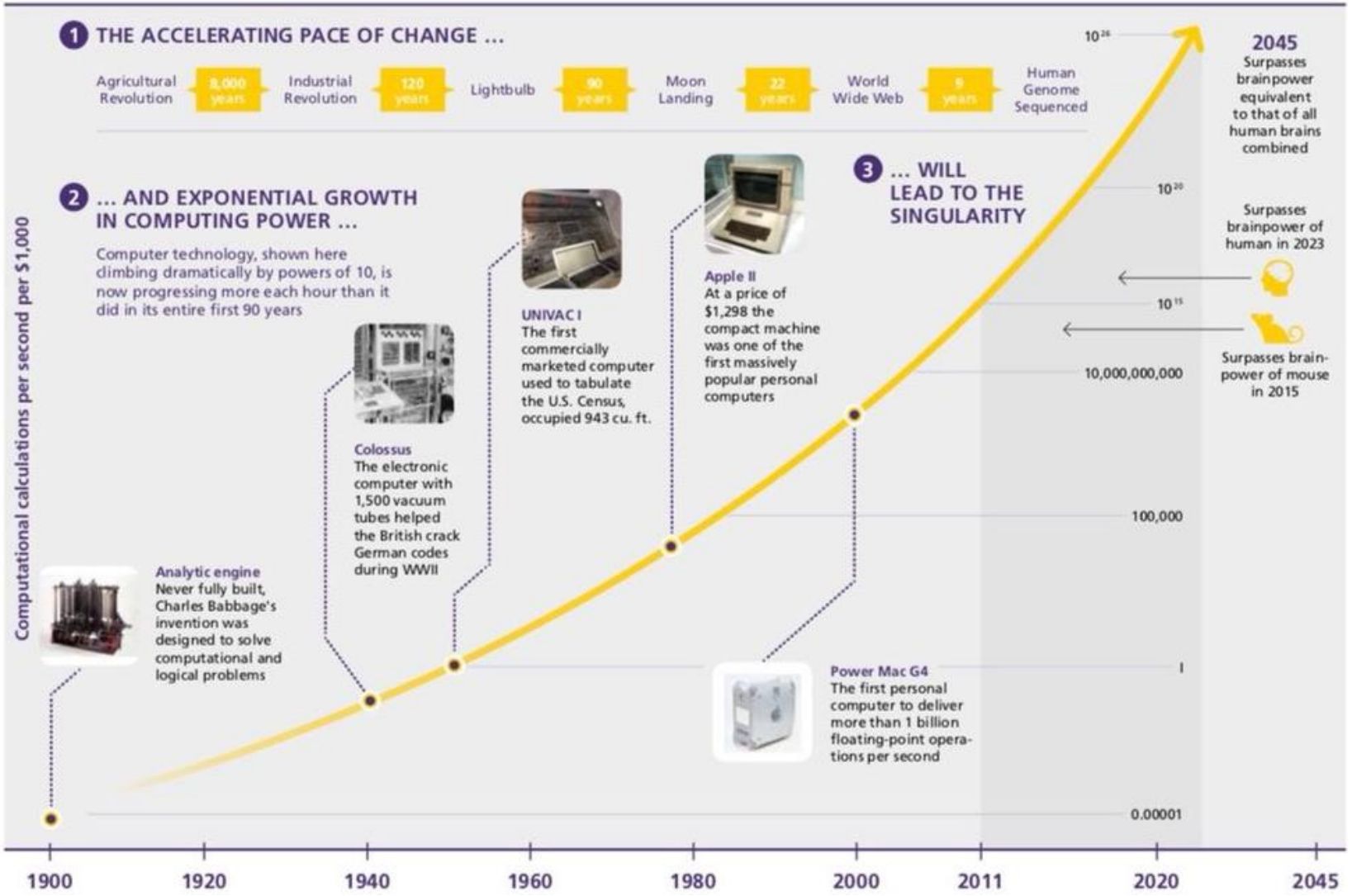
Zhihao Jia
Office hours: TBA

Xinhao Cheng
Office hours: TBA

What are the fundamental driving forces behind the success of ML?


ML
Model
Compute
Data
Hard
ware

Compute Per Second Per Dollar

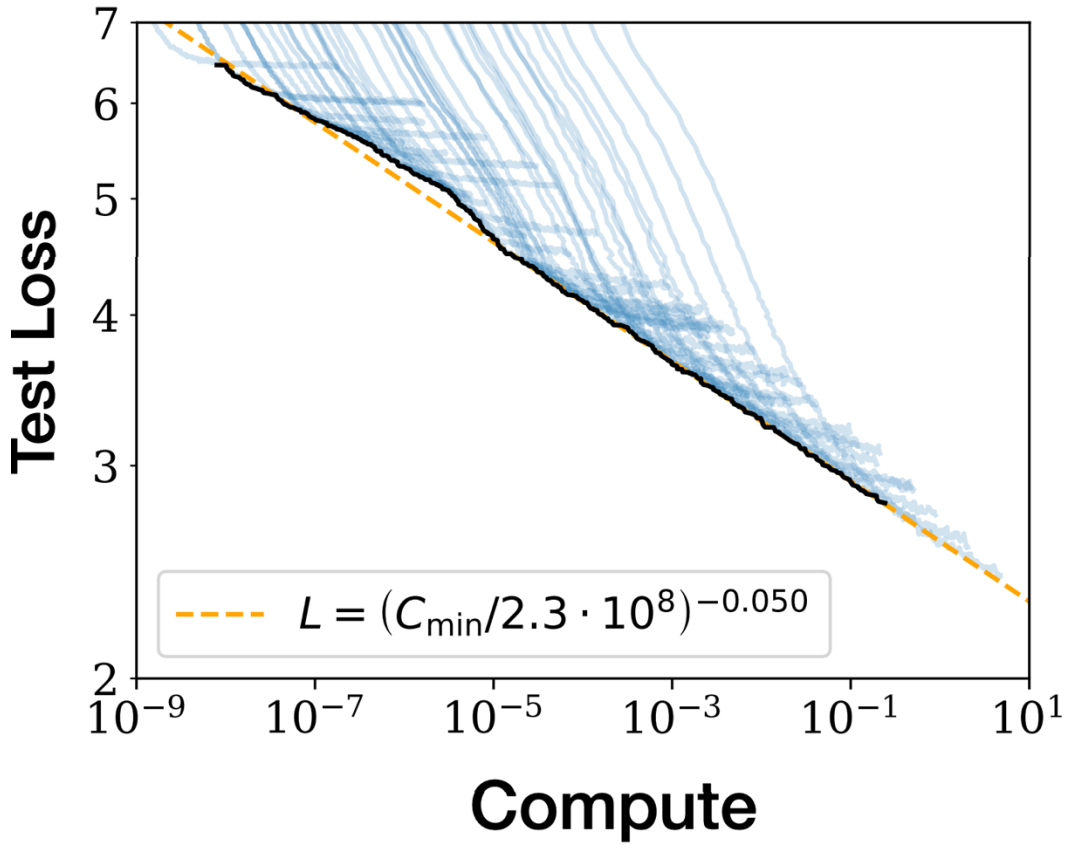


Surpass human brainpower in 2023

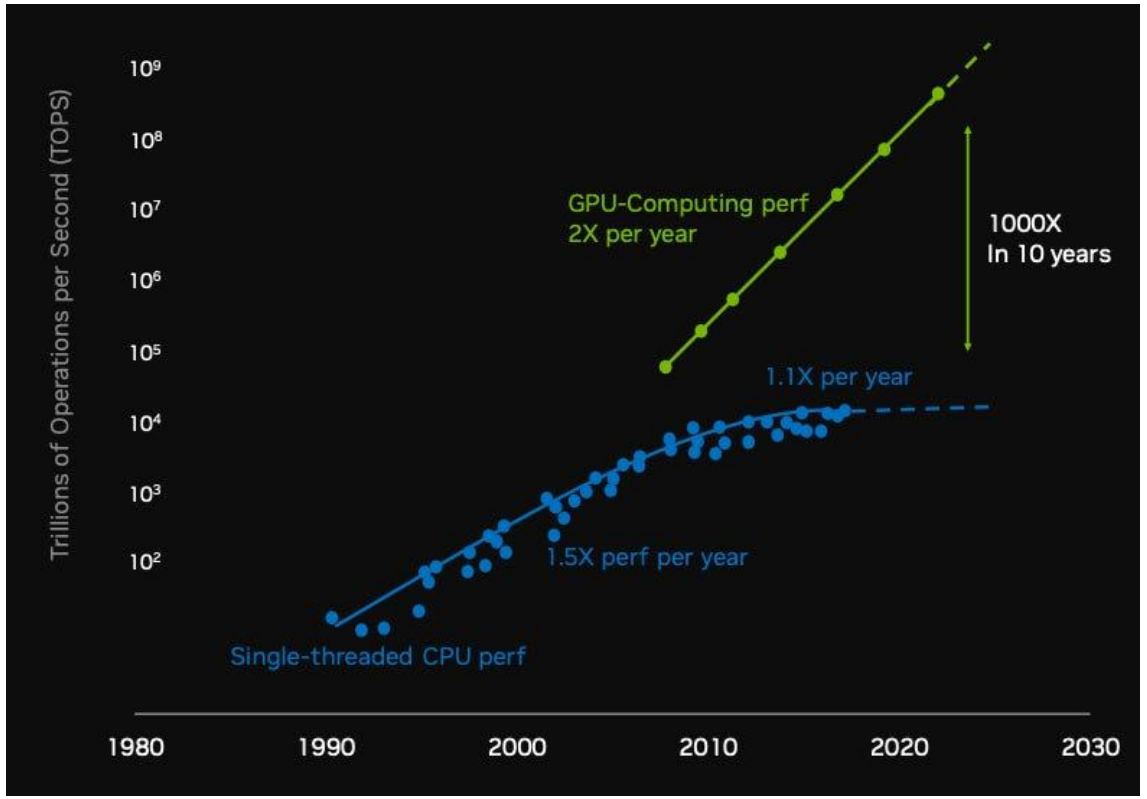
* Ray Kurzweil. *The Singularity Is Near: When Humans Transcend Biology.* 2005

Scaling Law in ML

Improving model accuracy & capability ← Scaling training & inference compute ← Hardware parallelization and specialization

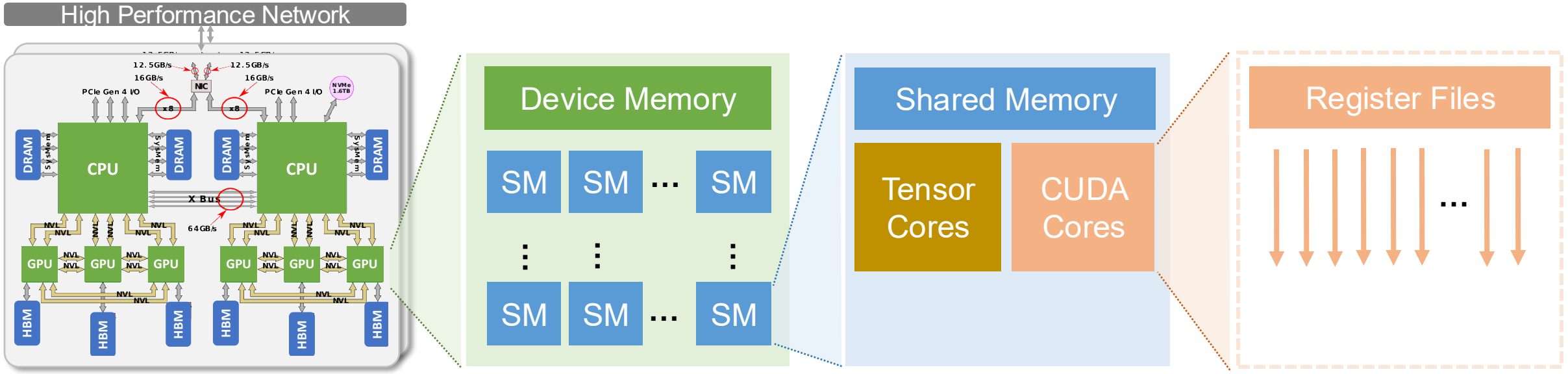


Source: OpenAI



Source: NVIDIA

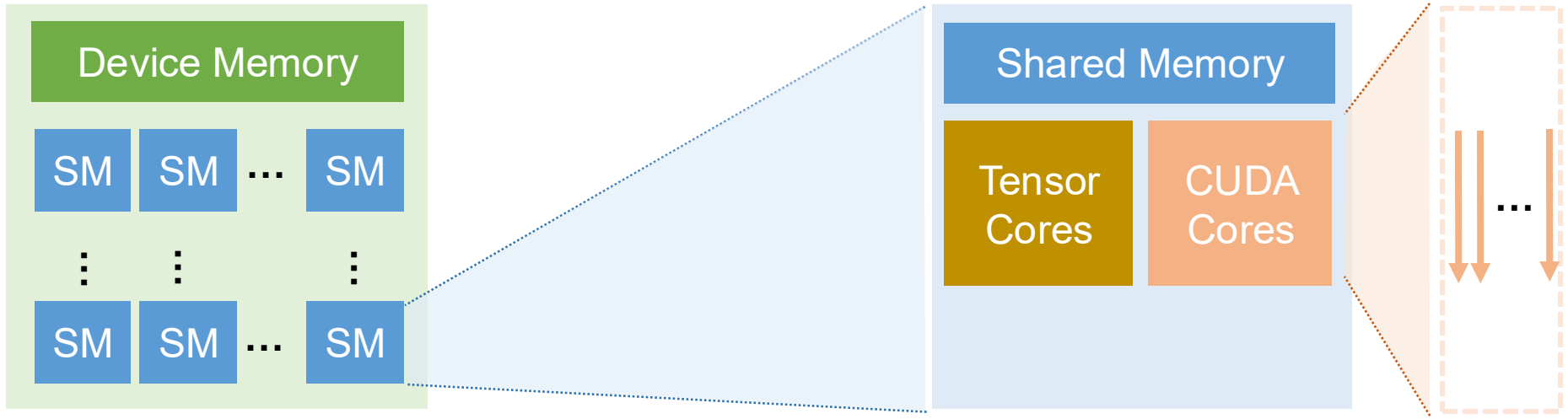
ML Hardware is Massively Parallel, Highly Heterogeneous



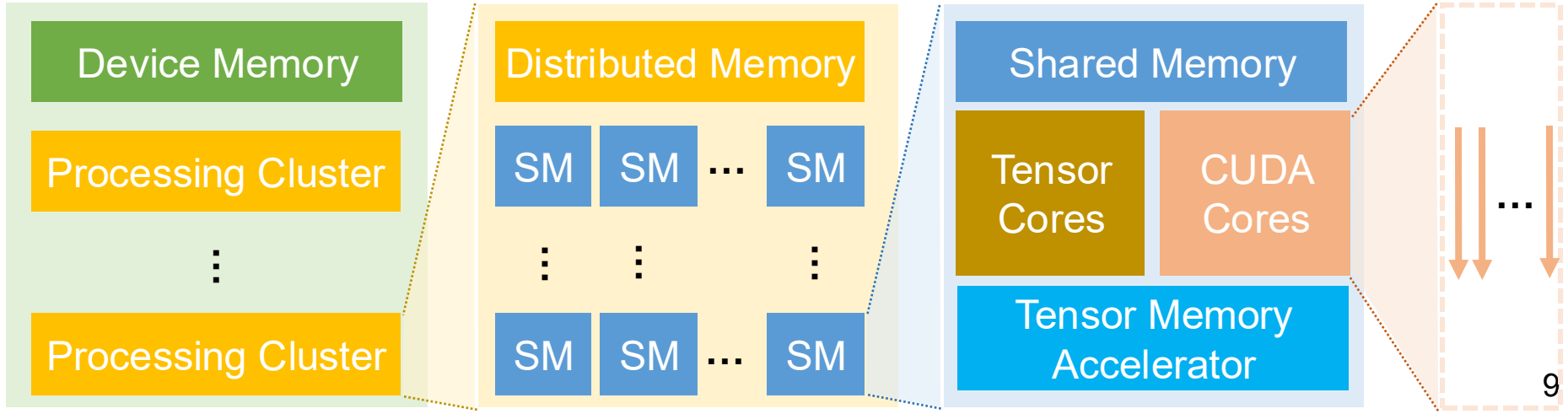
1K-100K GPUs × 100s streaming multiprocessors (SMs) × 128 CUDA cores 4 tensor cores = 10M-1B CUDA cores 400K-40M tensor cores

ML Hardware is Quickly Evolving

NVIDIA A100 GPU (2020)



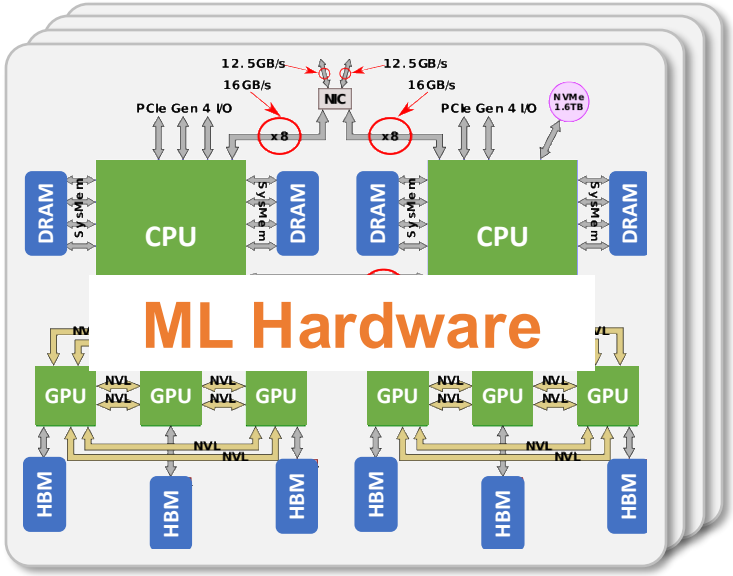
NVIDIA H100 GPU (2022)



This Course: ML Systems

Model

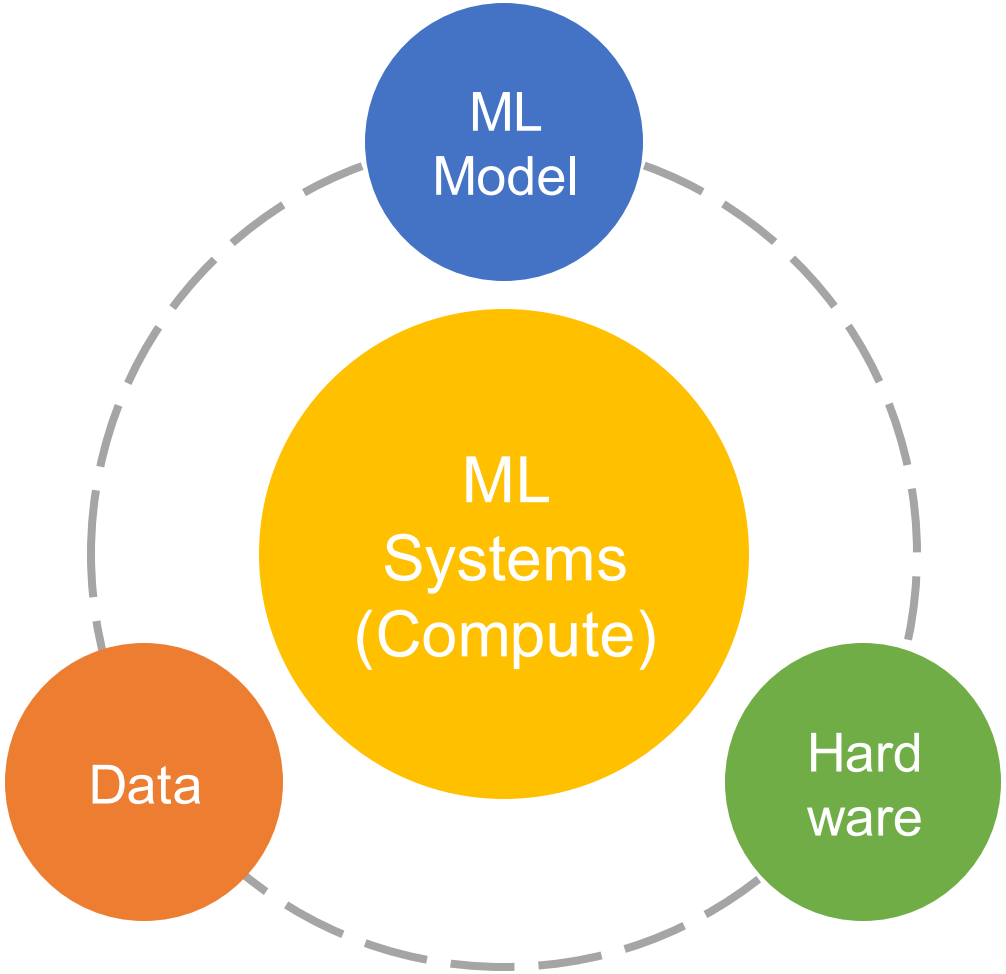
ML Systems



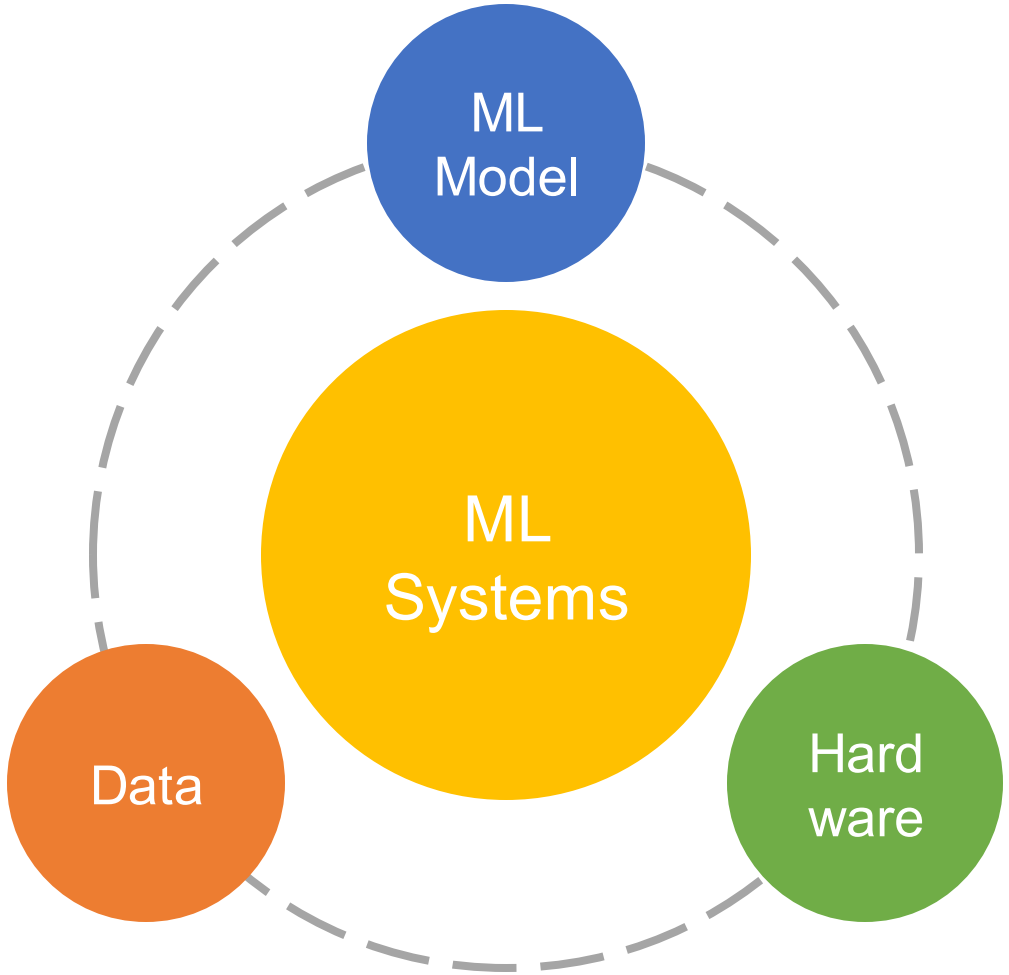
Goal:

Efficiently deploying ML applications on massively parallel, increasingly heterogeneous, rapidly evolving hardware platforms

ML Systems Bridge Model, Data, and Hardware

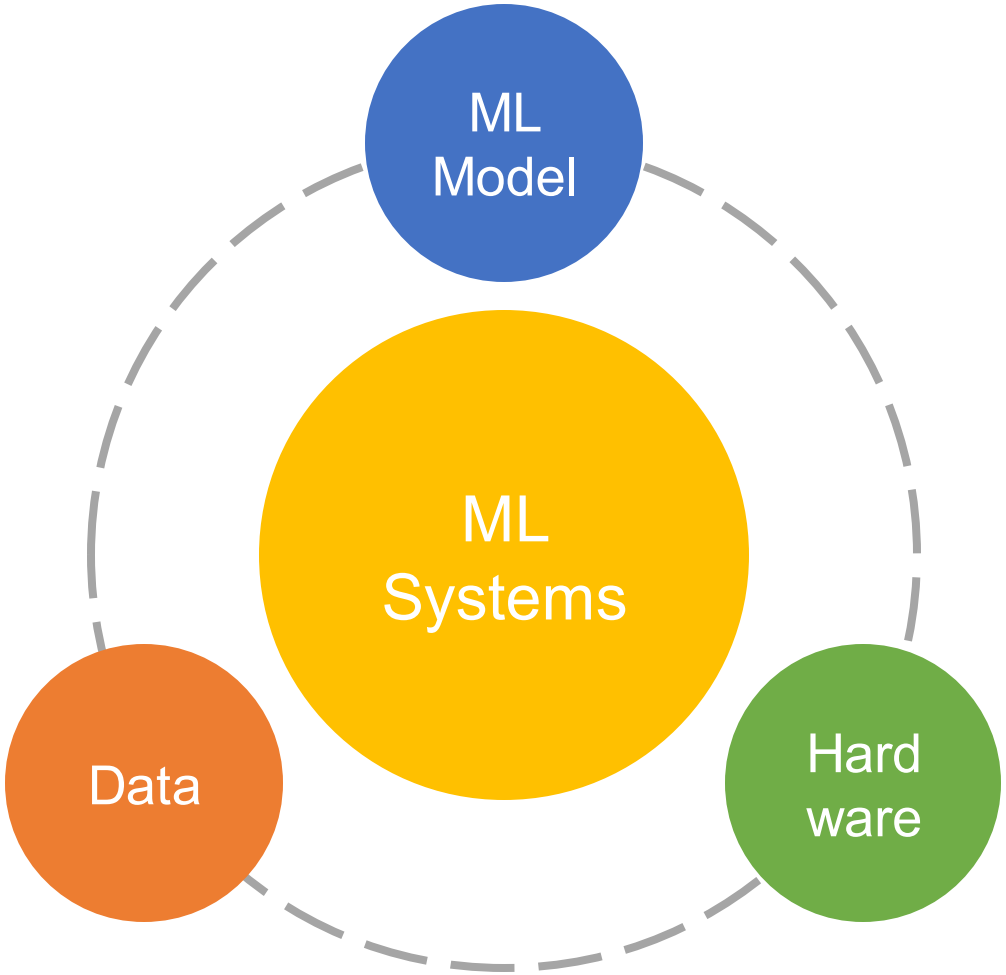


ML Systems as an Emerging Research Field

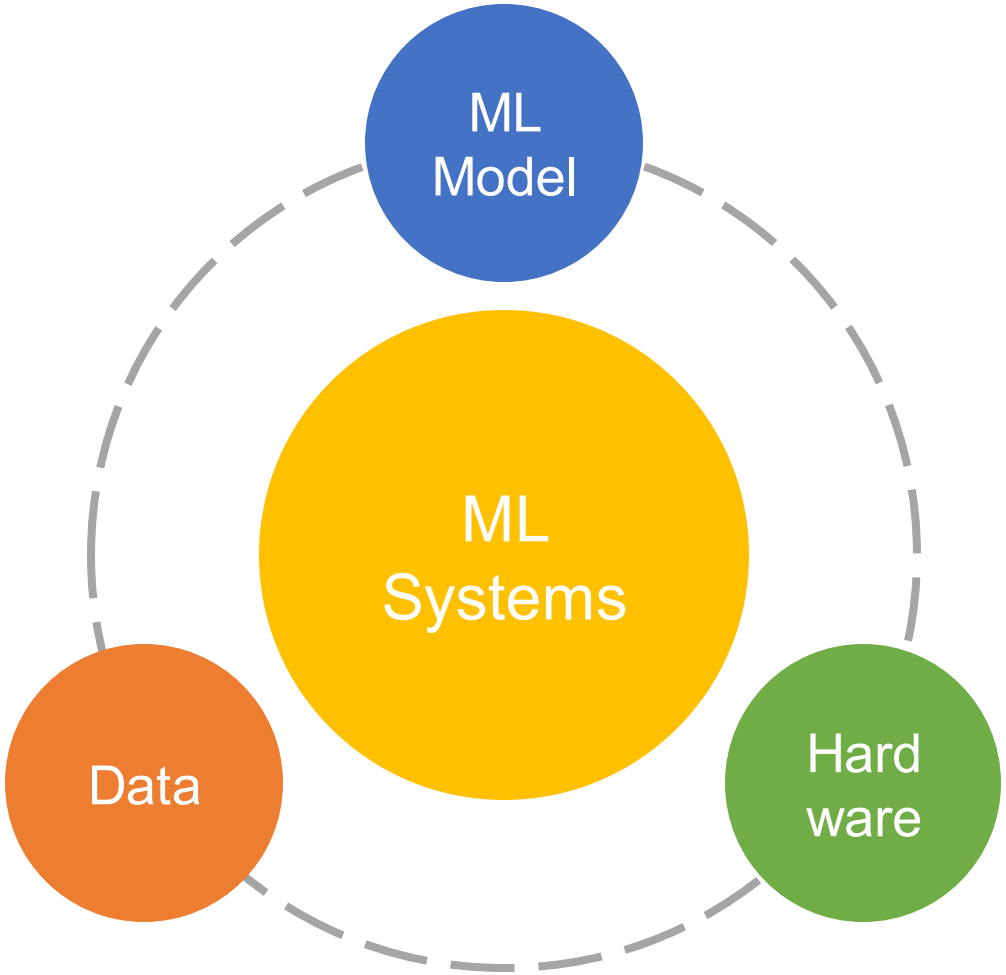


- MLSys papers at major ML and systems venues
- MLSys workshops at these venues
- mlsys.org: a new conference at the intersection of ML and systems

How is MLSys research different from typical ML and systems research?

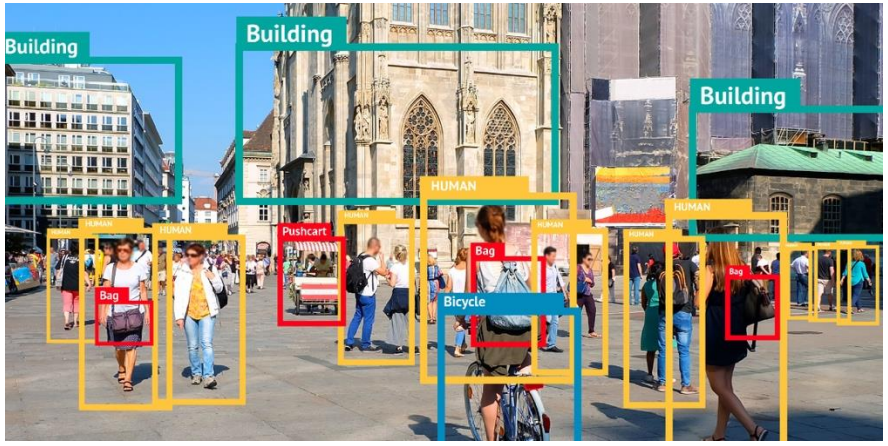


How is MLSys research different from typical ML and systems research?



MLSys provides a holistic approach to combining ML, data, systems, and hardware techniques to solve problems.

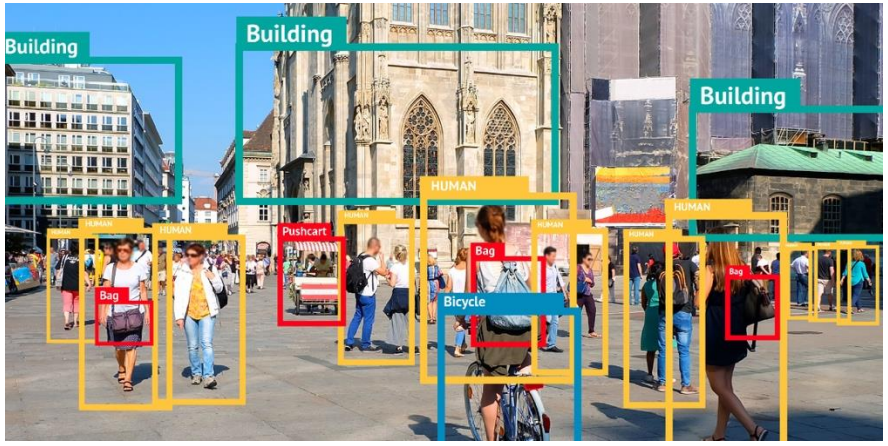
Exercise: Object Detection on surveillance camera



We want to deploy an object detection model on surveillance cameras:

- Accuracy >= 90%
- Latency <= 10ms
- Memory requirement <= 100 MB

A Typical ML Approach



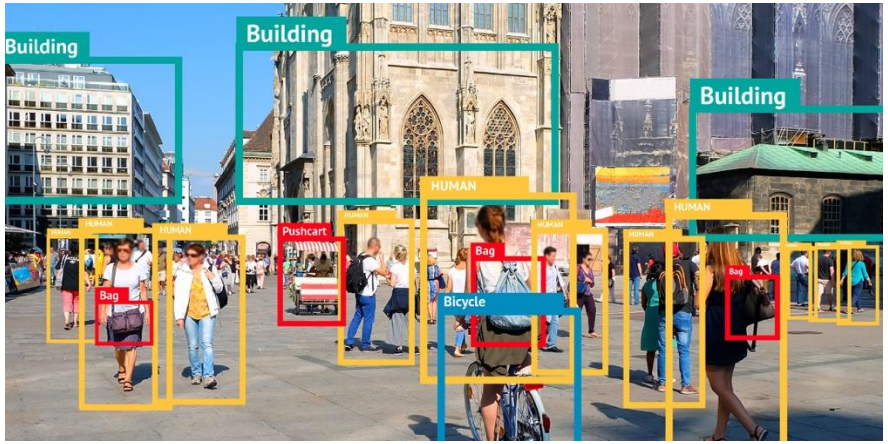
We want to deploy an object detection model on surveillance cameras:

- Accuracy >= 90%
- Latency <= 10ms
- Memory requirement <= 100 MB

Design models with better accuracy and smaller sizes

- Model pruning, quantization, distillation, low-rank approximation, etc..

A Typical Systems Approach

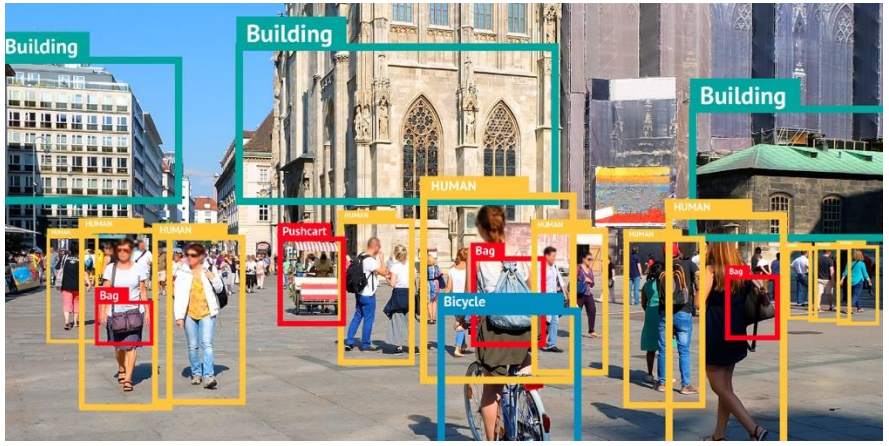


We want to deploy an object detection model on surveillance cameras:

- Accuracy >= 90%
- Latency <= 10ms
- Memory requirement <= 100 MB

Build a fast and memory-efficient inference engine with better resource utilization and runtime performance

An MLSys Approach



We want to deploy an object detection model on surveillance cameras:

- Accuracy >= 90%
- Latency <= 10ms
- Memory requirement <= 100 MB

Model/system/HW co-design and co-optimization

- Exploit specialized AI **hardware**
- Develop **models** optimized for the specific hardware
- Build ML **systems** that make use of the above points

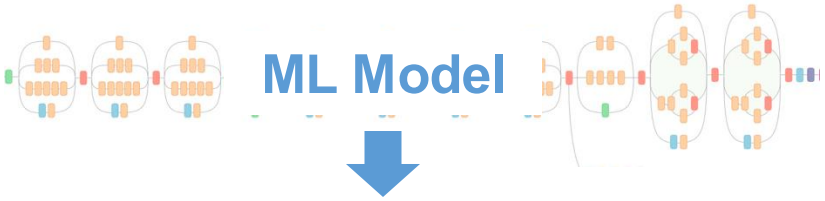
Why Study Machine Learning and Systems?

Reason #1 To push the frontier of modern AI applications, we need to have a holistic approach to the problem, understand and make use of existing systems more efficiently.

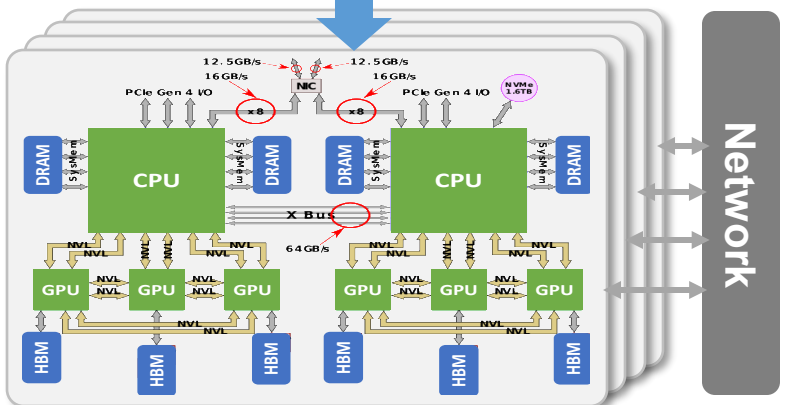
Reason #2 Prepare ourselves to build machine learning systems and work in the area of machine learning engineering.

Reason #3 Have fun building our own ML systems!

What will this course cover?

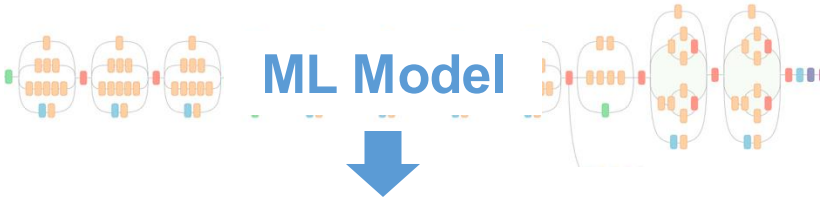


- Algorithmic Optimization
- Graph-Level Optimization
- Parallelization / Distributed Training
- ML Compilation
- Memory Management
- GPU Programming



We will learn the current design and key techniques across **full stack** in ML systems

ML Systems



Algorithmic Optimization

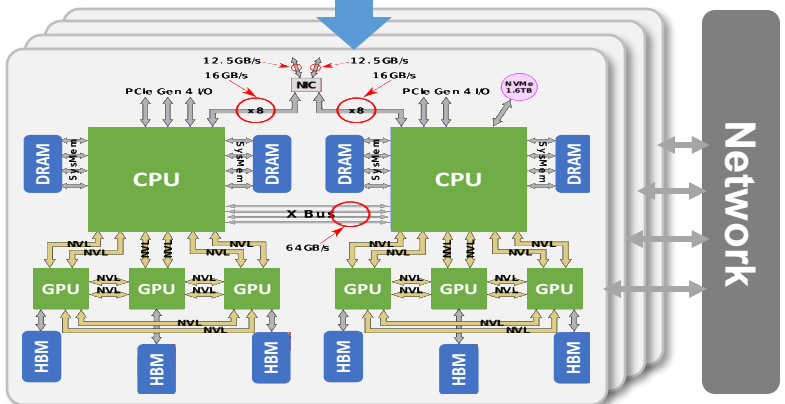
Graph-Level Optimization

Parallelization / Distributed Training

ML Compilation

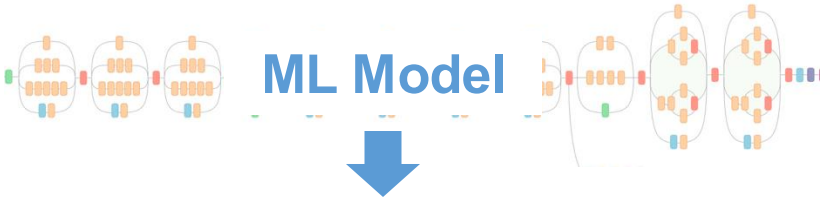
Memory Optimization

GPU Programming



- Modern GPU architectures
- CUDA programming
- Warp specialization
- Persistent kernel (mega-kernel)

ML Systems



Algorithmic Optimization

Graph-Level Optimization

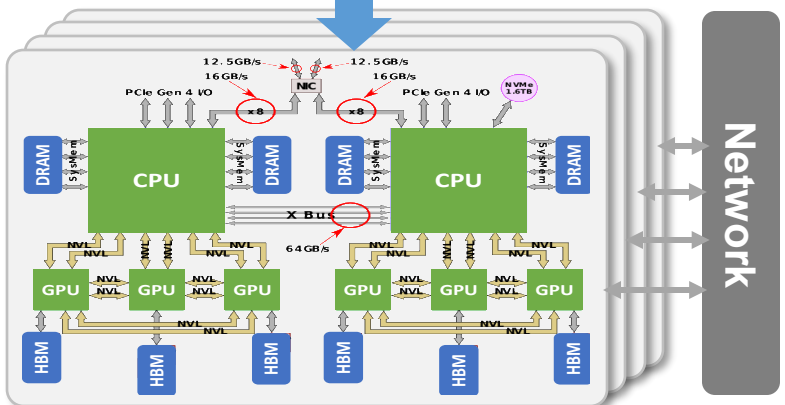
Parallelization / Distributed Training

ML Compilation

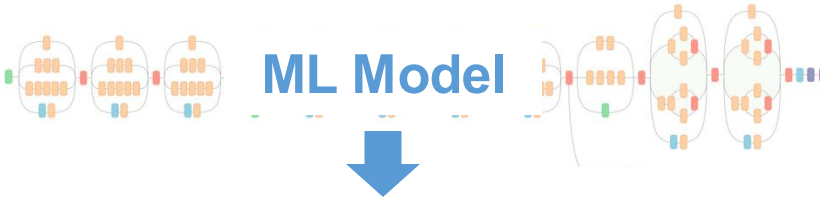
Memory Optimization

Kernel Programming

- Tile-based ML compilers
- Learning/search-based optimizations

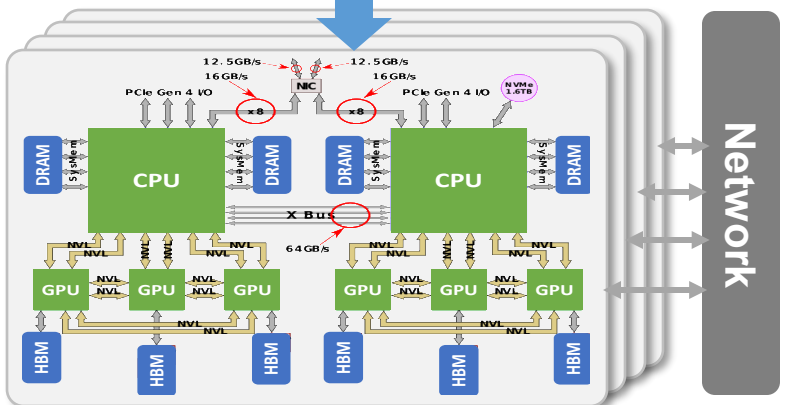


ML Systems

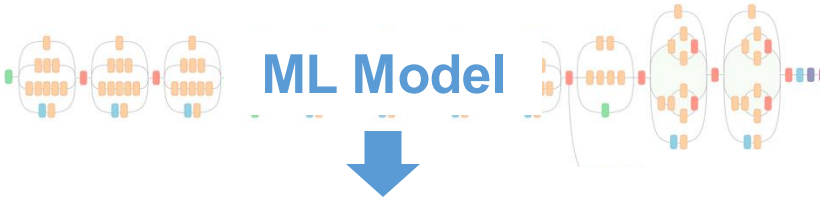


- Algorithmic Optimization
- Graph-Level Optimization
- Parallelization / Distributed Training
- ML Compilation
- Memory Optimization
- Kernel Programming

- Data parallelism
- Model parallelism
- Pipeline parallelism
- Auto parallelization



ML Systems



Algorithmic Optimization

Graph-Level Optimization

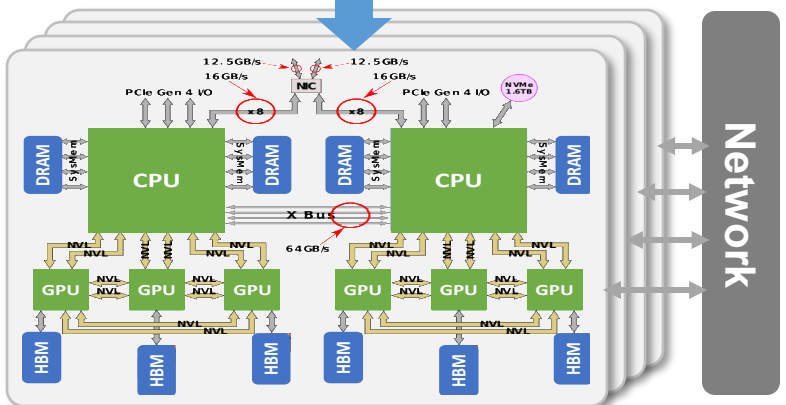
Parallelization / Distributed Training

ML Compilation

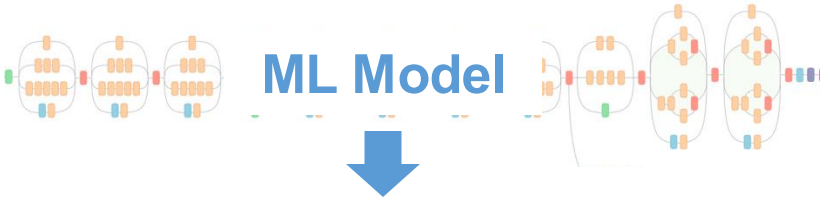
Memory Optimization

Kernel Programming

- Algebraic transformations
- Partially-equivalent transformations
- Superoptimization



ML Systems



Algorithmic Optimization

Graph-Level Optimization

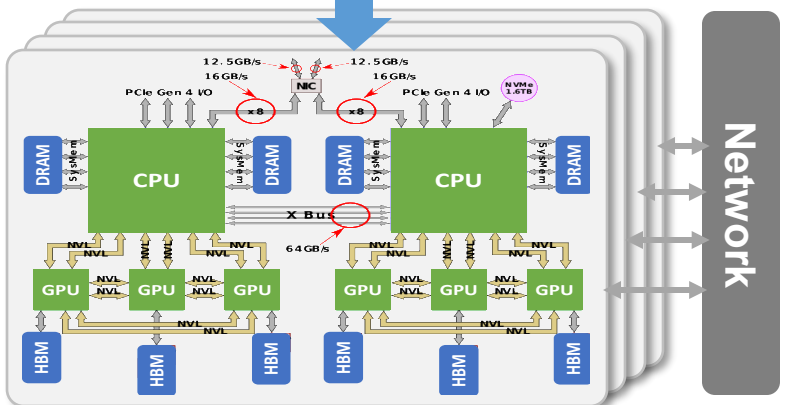
Parallelization / Distributed Training

ML Compilation

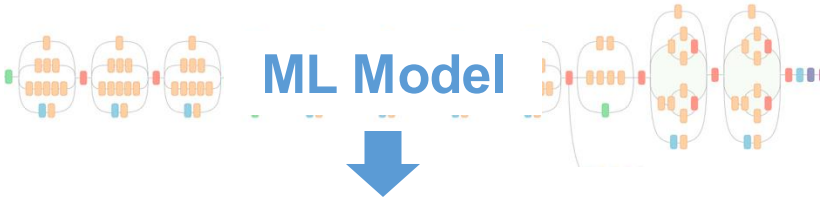
Memory Optimization

Kernel Programming

- Rematerialization
- Zero redundancy



ML Systems



Algorithmic Optimization

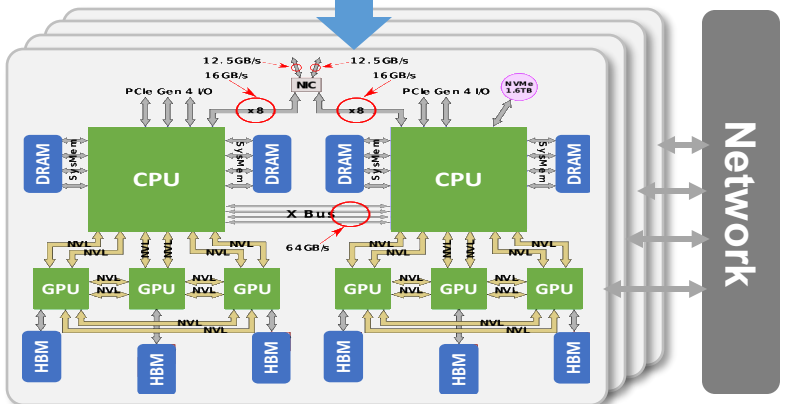
Graph-Level Optimization

Parallelization / Distributed Training

ML Compilation

Memory Optimization

Kernel Programming



- Pruning (quantization, sparsity, low-rank approximation)
- Fine-tuning
- Mixture of experts
- Retrieval augmentation

Learning Objects

By the end of this course, you will …

… understand the general components of modern machine learning frameworks

… learn MLSys techniques for emerging generative AI applications (LLMs)

… implement your own MLSys projects

Relations with Other MLSys Classes at CMU

- 10-414/714: deep learning systems (fall)
  - DL algorithm design and implementation
- 15-442/642: machine learning systems (spring)
  - Systems aspects of MLSys
- 15-779: advanced machine learning systems
  - Advanced MLSys techniques
  - Paper reading and discussions

Class Format

- Lecture (50 mins)
  - Introduce basic concepts for a topic
- Paper discussion (30 mins)
  - Discuss two recent MLSys papers on the topic
- Read papers and write reviews before each class
  - Familiarize with the topic and papers to discuss
  - Understand their strength and limitations
  - Learn and generalize ideas

Paper Reading (Starting from Week 3) How to Read a Paper

In each lecture, we will discuss one MLSys topic and two MLSys papers

Read these papers before the class and write a review

- Review details in the next slide

Keep in mind:

- What problem does this paper try to solve?
- Why is this an important and hard problem?
- Why can't previous work solve this problem?
- What is novel in this paper?
- Does it show good results?

Paper Review (due before each class)

- One short paragraph summarizing the first paper, in your own words
- One short paragraph summarizing the second paper, in your own words
- One short paragraph on any connections between the papers, such as
  - Compare and contrast: how one work is better than the other
  - Apply the ideas from one paper to solve the problem in the other
  - A new idea that can incorporate results from both papers

Final Course Project

- Team of 1-3 students (sign up in week 4), find your teammates early
- We will provide a list of potential project ideas. You are more encouraged to bring your own MLSys topics and ideas

Milestones:

- 1-page proposal
- Informal mid-term check-in with instructors
- Final presentation
- Paper writeup

Grading

- Course project: 50%
- Paper review: 30%
- Class participation: 20%

- All reviews and reports are submitted on Gradscope
- Ask questions and discuss on Piazza

Always refer to the website for more info:
https://www.cs.cmu.edu/~zhihaoj2/15-779/

Stay safe and have a great semester!