CSE 585: Advanced Scalable Systems for GenAI

Mosharaf Chowdhury

Today's Agenda

- Administrivia
- Topics
- Projects

About Mosharaf

- Associate Professor of CSE
  - http://www.mosharaf.com/
  - https://symbioticlab.org/
- Office hours:
  - Appointment-only

ViNEYard
virtual network embedding (2008-2012)

in-memory computing (2009-2014)

Coflow
data-parallel communication (2010-2016)

Infiniswap
memory disaggregation (2016-2022)

Salus
GPU resource management (2017-2022)

FedScale
systems for federated learning (2019-2024)

AI energy optimization (2021-)


systems for multimodal AI (2022-)

●●●

Jae-Won Chung (GSI)

- 5th-year PhD student at SymbioticLab

- Office hours from next week
  - Check course website
  - No office hours this week
  - jwnchung@umich.edu

Status

- As of today: 63 registered or w/ override
- If you are not planning to take the class, drop ASAP
  - Existing overrides that have not converted will be revoked

Course Schedule

- **Webpage:** https://github.com/mosharaf/cse585
- **Meetings**
  - 10:30 AM – 12PM (**T/Th** for lectures and seminars)
  - 1:30 PM – 2:30 PM (**Fri** for makeups and projects)
- **Pay attention to the online announcements and schedule**
  - On average, two meetings per week
  - Friday makeups will be added on a need-to-add basis

Prerequisites

- **EECS 482 / 484 / 489 / 491**
  - Equivalent courses are acceptable as well
- **Good programming skills**
  - Build substantial systems for course project

Course Requirements

Paper Summary	15%
Paper Presentation	15%
Participation	10%
Project Report	40%
Project Presentations	20%

Topics (#Lectures)

- Basics (3)
- Pre-Training (4)
- Post-Training (2)
- Inference (5)
- Agentic Systems (2)
- Hardware/Infrastructure (1)
- Power and Energy (2)
- Ethical Considerations (1)

Group-Based Work

- ALL activities will be done in groups except for participation
  - Paper presentation
  - Paper summary
  - Research projects

Form Groups ASAP

- **Submit** Google Form
  - By September 4 the latest, but **right now** is better
    - We need a group to pickup duties for Sep 4!!!
  - Use Ed to find group members
  - Group size should be 4

Readings

- **~40 papers/articles across**
  - Primarily from systems venues like SOSP, OSDI, NSDI, ASPLOS, FAST, etc.
  - A couple from traditional AI/ML venues but still with systems-y flavor

Paper Presentation

- **This is a seminar-style course**
  - Each group must present at least one lecture (required papers and the rest)
  - Paper presentation account for **15%** of the total grade
- **The entire class will be dedicated to the assigned paper(s)**
  - Aim for 40-minute presentation without interruption
  - But there will be intermittent discussions
- **Lead the discussion**
  - Go through the paper in details, along with its strengths and weaknesses
  - Include companion papers and other related papers

Paper Presentation

- Share your slides to cse585-staff@umich.edu 24 hours before the class
  - Use Google Sheets so we can provide in-place comments/feedback
- Prepare early
- Practice a lot
- Also, read
  - How to Give a Bad Talk, by David A. Patterson

Paper Summaries

- **This is a paper-reading course**
  - Paper summaries account for **15%** of the total grade
- **Roughly 1-2 summary per-group (assigned)**
- **Each summary must follow the template and address the following**
  - What is the problem and why is it important?
  - What is the hypothesis of the work?
  - What is the proposed solution, and what key insight guides their solution?
  - What is one (or more) drawback or limitation of the proposal, and how will you improve it?
- **Summary must include the gist of class discussion**

Paper Summaries

- **Reviews must be shared to cse585-staff@umich.edu within 24 hours of class presentation**
  - Use Google docs so we can provide in-place comments/feedback
- **Delayed submission will receive NO CREDIT**
  - There will be NO extensions

Panel Discussion

- **The Authors**
  - Groups that present and write summary
- **The Reviewers**
  - Each group will be assigned to at least one of these slots
  - Will have their own questions to ask to the authors
  - Will receive questions raised by the class (described below) from the GSI before the lecture
- **Rest of the Class**
  - Submit one insightful question for each presented papers by 3PM the day before
  - Ask questions directly too

In general,

- No extensions
- Everyone must come to class after reading the required papers of the day

What Do We Talk About
When We Talk About
"Advanced Scalable Systems for GenAI"

Resource-Centric View

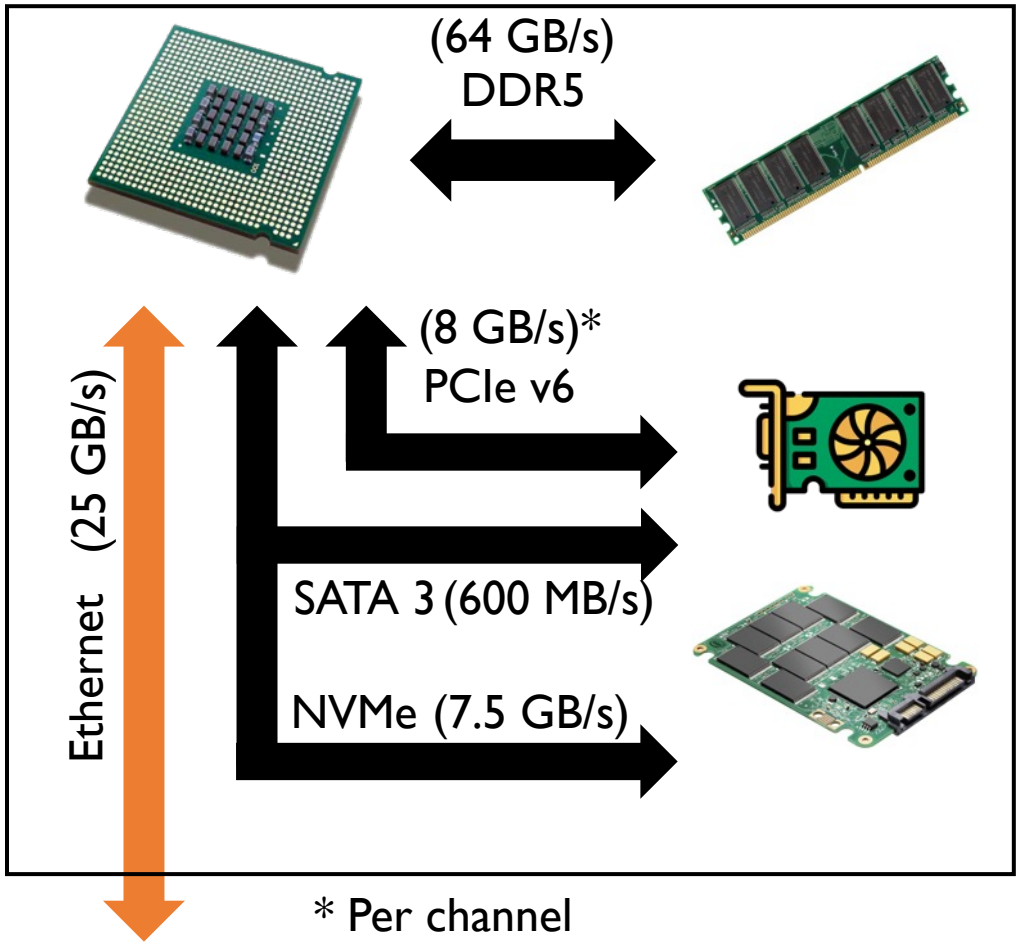
What's in a (Simplified) Server?

Interconnected compute and storage resources

- Different bandwidth and latency constraints

Simplified diagram

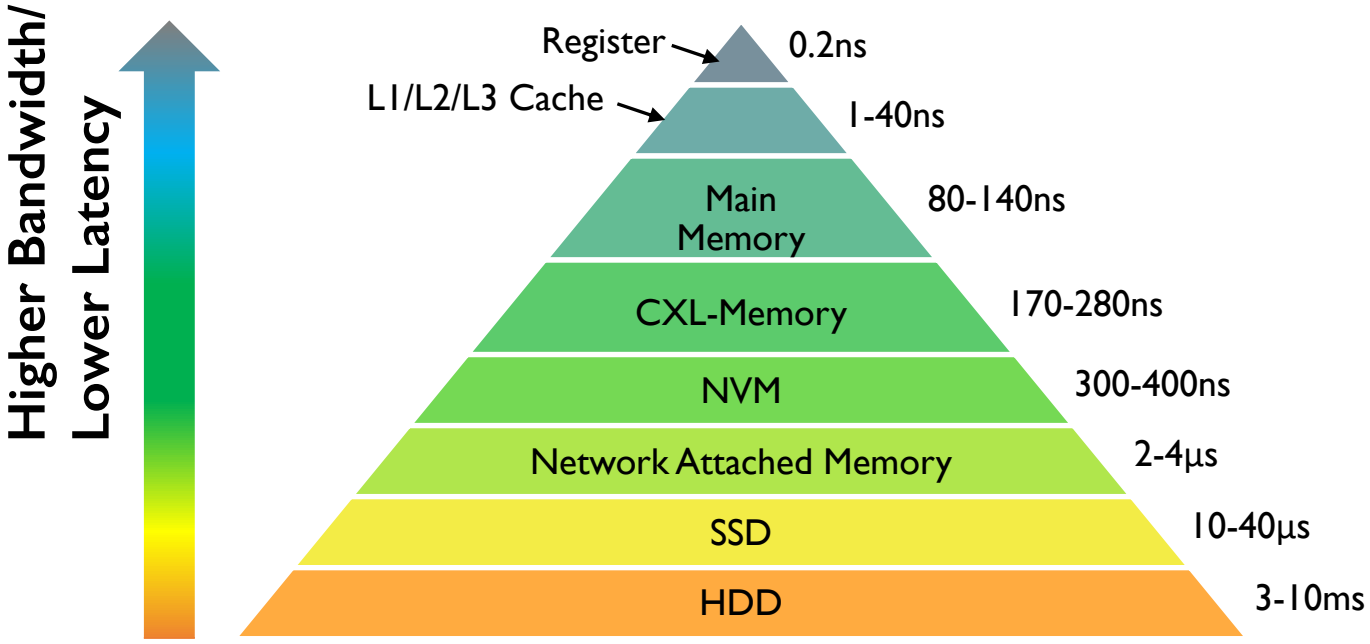
- Doesn't include faster networks such as RDMA, CXL, dedicated GPU interconnects such as NVlink, etc…



Typical Memory/Storage Hierarchy

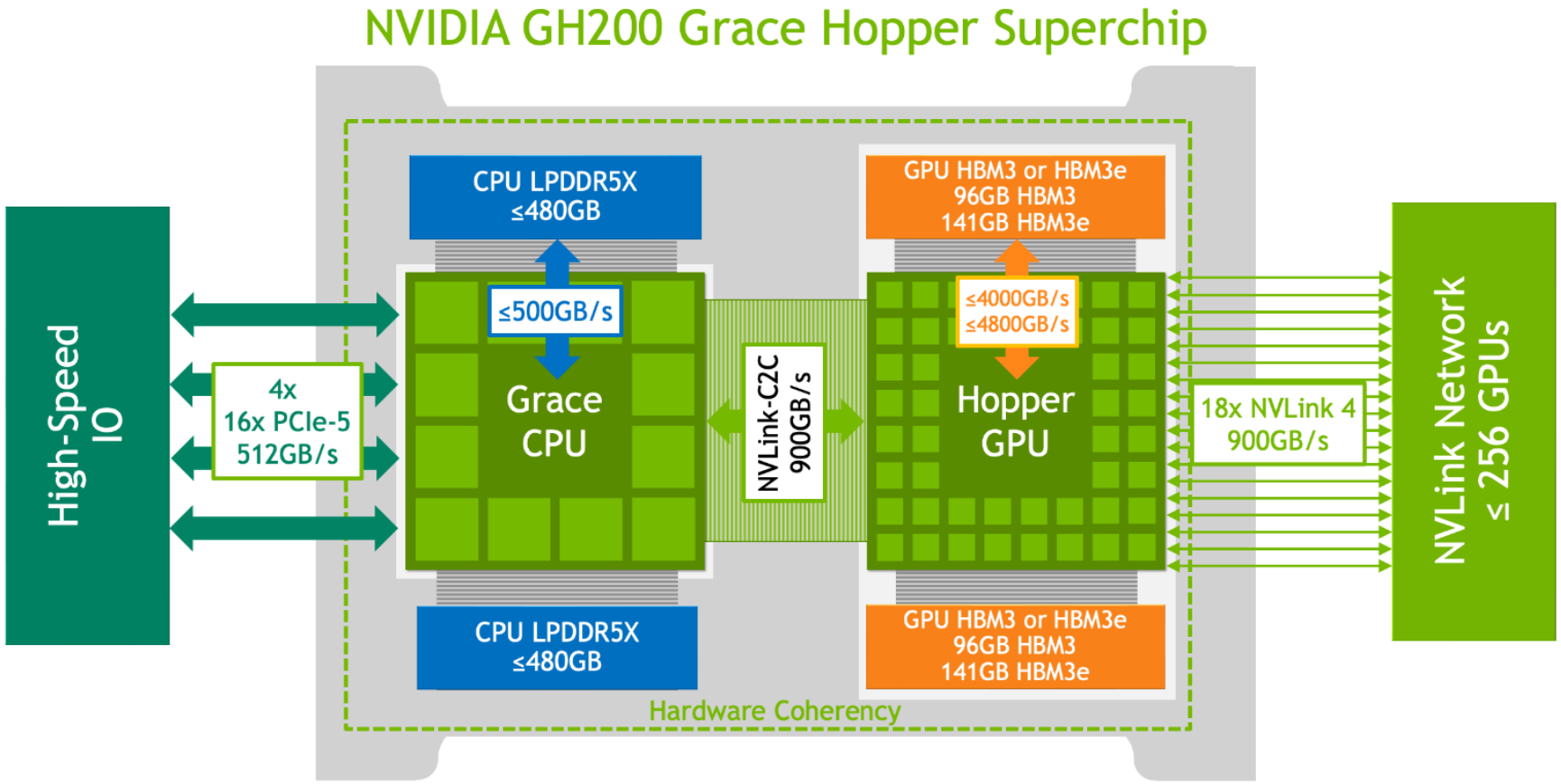
Fundamental Goals of (SW/HW) System Design

- Minimize time to access data
- Maximize compute utilization
- **Balanced System**



Maruf et al, SIGMETRICS 2023

What's in a Modern-ish AI Server?



https://resources.nvidia.com/en-us-grace-cpu/nvidia-grace-hopper

Scale Out: Warehouse-Scale Computer (WSC)

Single organization

Homogeneity (to some extent)

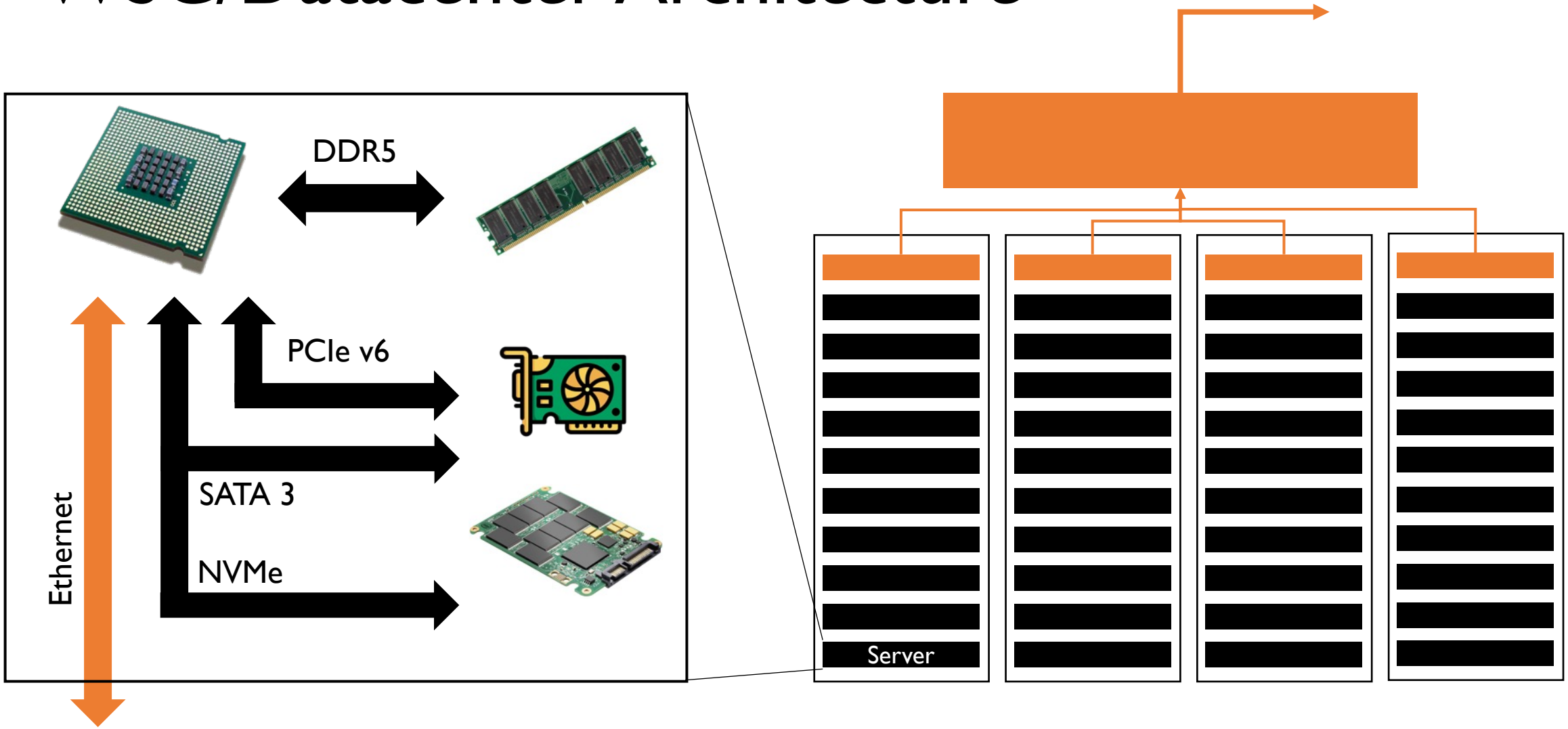
Cost efficiency at scale

- Multiplexing across applications and services
- Rent it out!

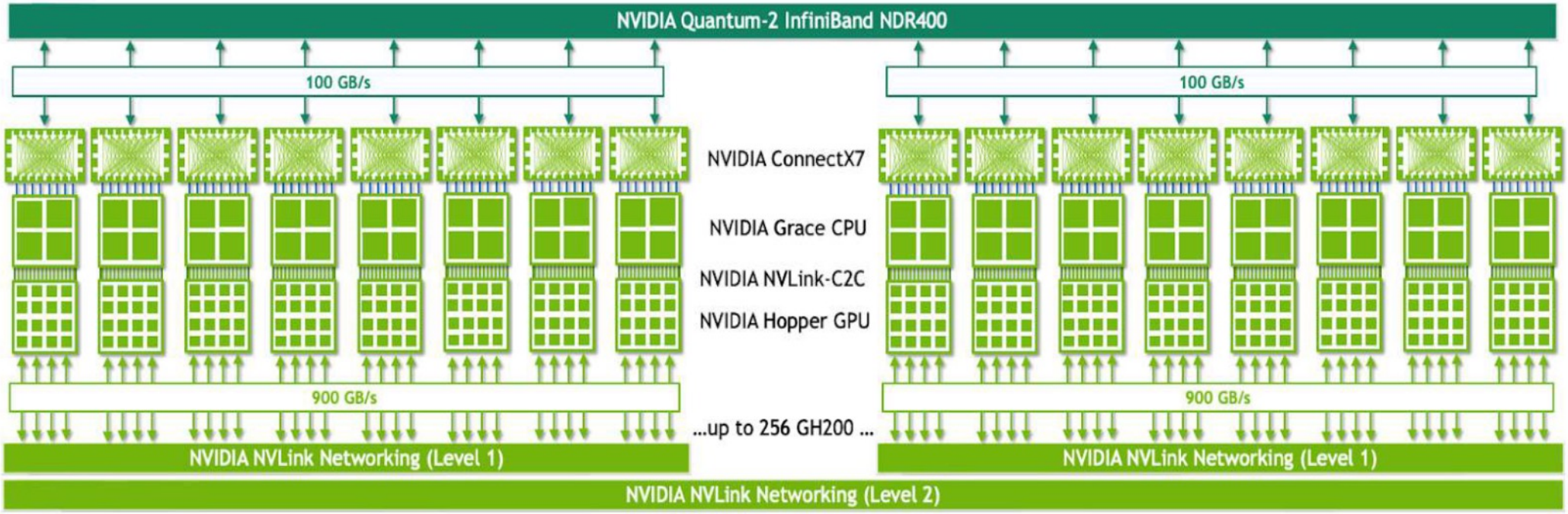
Many concerns

- Infrastructure
- Networking
- Storage
- Software
- Power/Energy
- Failure/Recovery
- …

WSC/Datacenter Architecture



Example: Scaling Out Using NVIDIA GH200



https://resources.nvidia.com/en-us-grace-cpu/nvidia-grace-hopper

Datacenter Needs an Operating System

Datacenter is a collection of

- Compute
- Memory
- All connected by an interconnect

Not unlike a computer

Some differences

1. VERY high level of parallelism
2. VERY large scale
3. Diversity of workloads
4. Resource heterogeneity
5. Failure is the norm

Three Categories of Software

1. **Platform-level**
   - Software firmware that are present in every machine
2. **Cluster-level**
   - Distributed systems to enable everything
3. **Application-level**
   - User-facing applications built on top

Common "Systems" Techniques

Technique	Performance/Efficiency	Availability/Resilience
Replication & Erasure coding	X	X
Sharding/partitioning	X	X
Scheduling & Load balancing	X	
Health & Integrity checks		X
Compression & Quantization	X	
Centralized controller	X	
Canaries		X
Speculation & Redundant execution	X	

Break!

Workload-Centric View

Machine Learning Fleet Efficiency @ Google

Table 1: Comparison of Machine Learning (ML) Fleet, Warehouse Scale Computer (WSC), and High-performance Computing (HPC).

Category	Warehouse Scale Computer	High-Performance Computing	Machine Learning Fleet
Workload types	Diverse web services (search, email, social networking, media streaming)	Scientific simulations, graph computations, solvers	Training of ML models, real-time serving, bulk inference
Fleet composition	More stable, as most user demand has reached a steady state or known patterns	A large portion of demand is predetermined, as it is driven by scientific missions	Rapidly changing due to newly emerging ML models and increasing user demand
Hardware heterogeneity	General-purpose CPUs	CPUs, GPUs, other ASICs	CPUs, GPUs, TPUs, FPGAs, other ASICs.
Hardware/Software co-design	Hardware is workload-agnostic	Hardware is chosen for specialized applications	ASICs often co-designed with workloads in mind

Anatomy of an ML Fleet: Hardware and Software Infrastructure

Accelerators

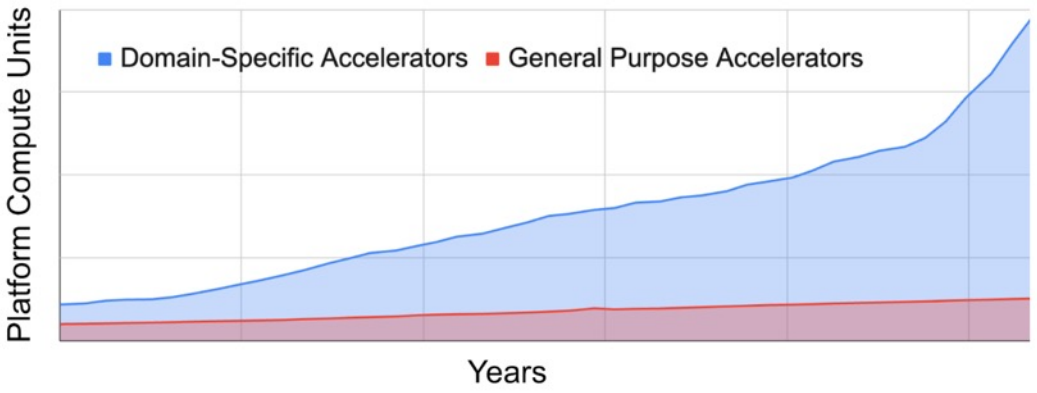


Figure 1: Five-year historical ML fleet breakdown by accelerator type. The rapid proliferation of domain-specific accelerators in response to ML-based workloads has presented novel challenges in optimizing ML fleets. Managing these domain-specific accelerators means effectively handling hardware and workload heterogeneity, as well as hardware-software co-design at scale.

Scheduler

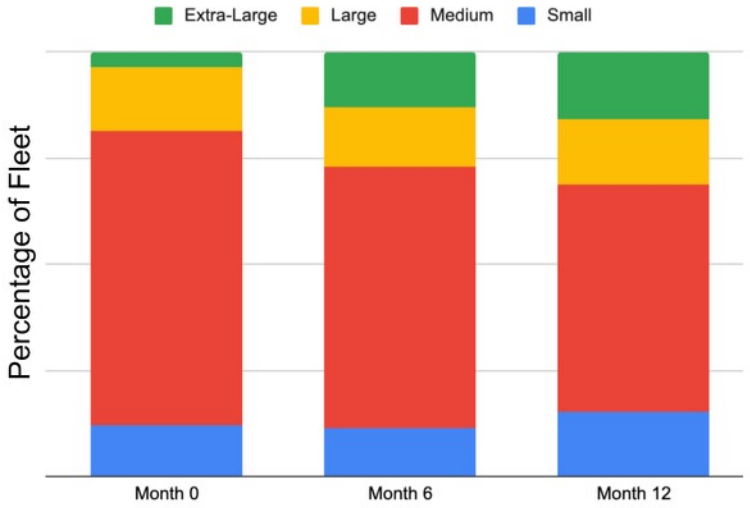


Figure 4: A sample breakdown of Google's ML fleet for internal workloads, segmenting on workload topology size (the number of accelerators requested by a given job). Progressive snapshots over the course of one year illustrate the ML fleet's growing share of jobs using an "extra-large" number of accelerators. This demonstrates how an ML fleet scheduler must be able to adapt to changing conditions, as the evolution of job sizes and topologies in response to shifting ML workloads presents unique challenges for the entire fleet.

Anatomy of an ML Fleet: Programming

Runtime/Compiler

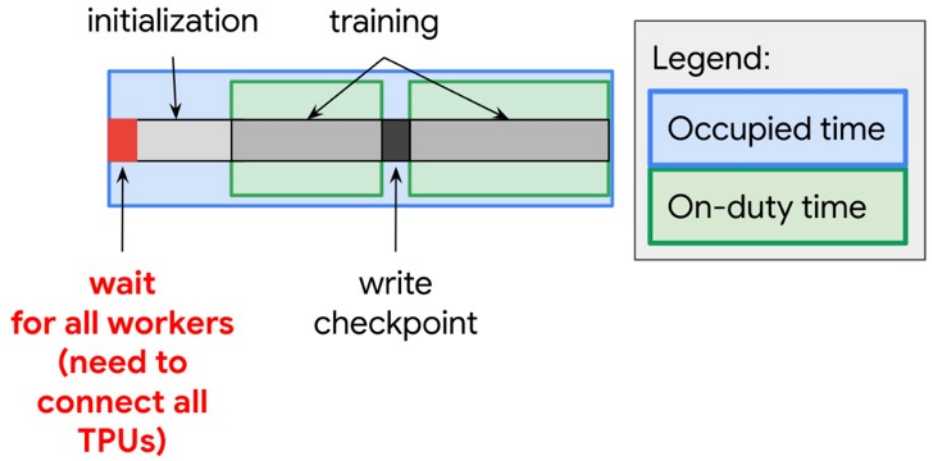


Figure 5: An ML workload requires all requested TPUs to be allocated before the task can start. In this example of a training workload, forward progress is saved via checkpoints. Delays during workload initialization and checkpoint writing, which are part of the Runtime and Framework layers, can reduce overall system efficiency.

Frameworks

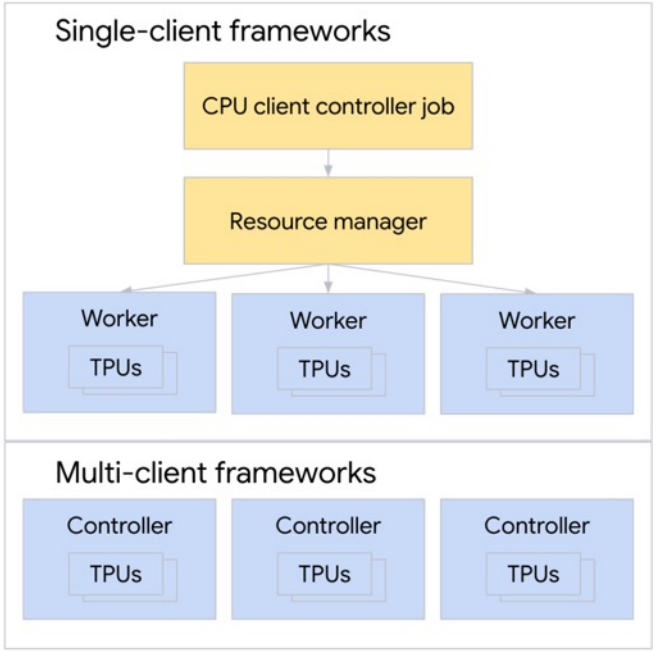


Figure 7: Comparing single-client frameworks with multi-client frameworks.

Anatomy of an ML Fleet: Workloads

Model

- Drives computation demands and patterns, both over time and spatially
- Changes frequently as new models, training paradigms, etc. emerge and become popular

Data

- Data pipeline/IO (system memory, storage, networking) can easily bottleneck the overall fleet
- Both quality and quantity matters
- Both pre- and post-processing matters

MPG: ML Productivity Goodput

- **Scheduling Goodput (SG)**
  - How often does an ML application have all necessary resources to make progress?
- **Runtime Goodput (RG)**
  - When it does, how often does it make progress?
- **Program Goodput (PG)**
  - When it's progressing, how close is it to maximum achievable efficiency?

$$\text{ML Productivity Goodput (MPG)} = \text{Scheduling Goodput (SG)} \times \text{Runtime Goodput (RG)} \times \text{Program Goodput (PG)}$$

$$\text{SG} = \frac{\text{all-allocated}}{\text{capacity}} \qquad \text{RG} = \frac{\text{productive}}{\text{all-allocated}} \qquad \text{PG} = \frac{\text{predicted}}{\text{productive}}$$

MPG: ML Productivity Goodput

Utilization

≠

Productivity

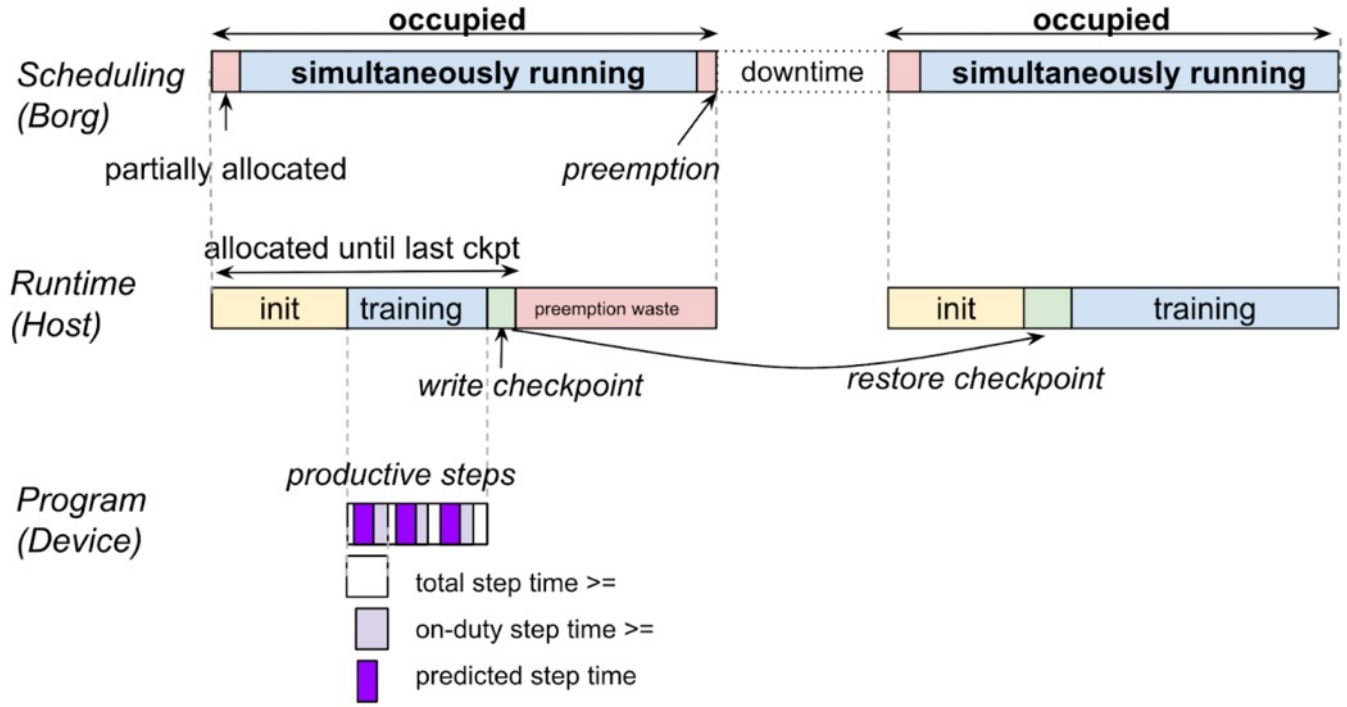


Figure 9: Breakdown of a ML workload using ML Productivity Goodput.

Projects

Research-Oriented Course!

- The final project accounts for **60%** of total grades
- What can and cannot be a project?
  - Just surveys are not allowed
  - Measurements of new environments or of existing solutions on new environments are acceptable
  - Reproducing results from existing solutions is also acceptable
- An ideal project should answer the questions you asked during paper reviews and points you cared about for presentations

How to Approach it?

1. Find a problem and motivate why this is worth solving
2. Quickly survey background and related work
   - Might require you to go back to the first step
3. Form/update your hypothesis
4. Test your hypothesis
   - Go back to 3 until you are happy
5. Present your findings on poster and in writing
   - Discuss known limitations

Milestones

Date	Milestone	Details
09/04/25	Form Group	Find like-minded students
09/18/25	Submit Proposal	Send your proposal by email to receive feedback either via email or in-person or both
10/21/25 10/23/25	Mid-Semester Presentations	Define and motivate a problem, overview related work, and form initial hypothesis and idea
12/04/25	Poster Presentation	Present your findings
12/15/25	Research paper	Submit a report like the papers you read

Draft Proposal (Sep 18)

- **Two pages including references that must include**
  - What is the problem?
  - Why is it important to solve?
  - Any initial thoughts on what you want to do?
  - How would you evaluate your solution?
- **Include team members**
  - Meaning, form a group ASAP
- **Approved by the instructor and agreed upon by you**
  - Forms the basis of expectation

Mid-Semester Checkpoint (Oct 21,23)

- **In-class short presentation over two days**
  - This is to make sure you are making progress
- **Must include**
  - What is the problem?
  - Why is it important?
  - What are the most related work?
  - What's your hypothesis so far?
  - How are/will you evaluate it?

Presentation & Paper (Dec 4, 15)

- **Research paper**
  - The key part
  - Should be written like the papers you've read
  - As if you'd submit it to a workshop with ~3 more months of work or to a conference after ~6 more months of work
  - How to Write a Great Research Paper by Simon Peyton Jones
- **Extended from the mid-semester checkpoint**

Next Class…

Read the required readings

Form groups of 3-4 and fill out Google Form by *Sep 4*

- Decide if you'll drop, **before** you fill it
- If you are to drop, drop immediately