Systems for Machine Learning (CMSC828G)

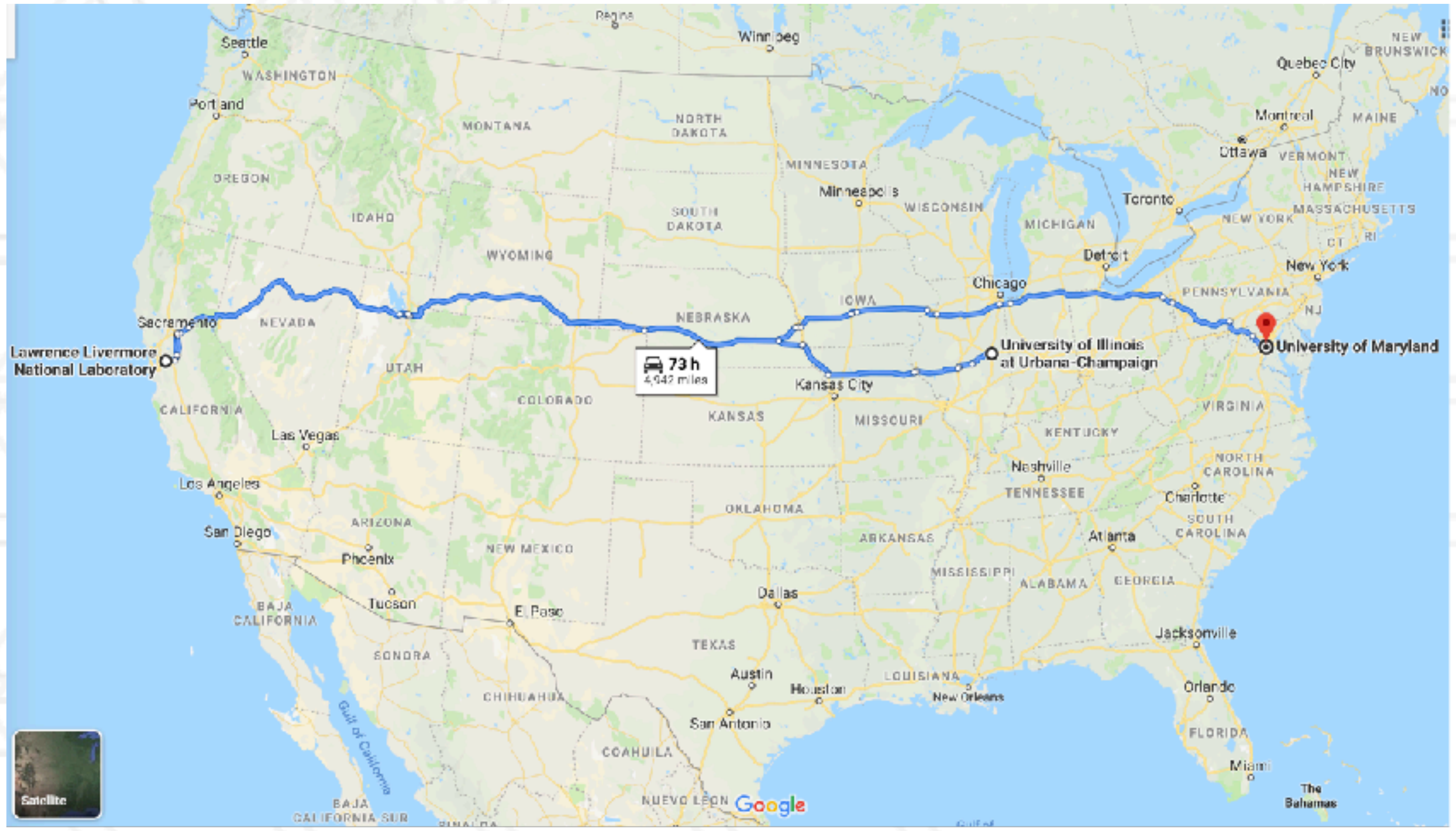
CMSC828G: Course Overview

Abhinav Bhatele, Daniel Nichols

UNIVERSITY OF
MARYLAND

Instructors: Dr. Bhatele

- Ph.D. from the University of Illinois at Urbana-Champaign (midwest)
- Spent eight years at Lawrence Livermore National Laboratory (SF bay area)
- Sixth year at the University of Maryland
- Research areas:
  - High performance computing
  - Distributed AI


Lawrence Livermore National Laboratory
73 h
4,942 miles
University of Illinois at Urbana-Champaign
University of Maryland
Satellite
Google

Instructors: (soon-to-be Dr.) Nichols

- Originally from Philadelphia, PA
- 5th and final year PhD student
- Research:
  - The intersection of systems and ML
  - Code LLMs

Student introductions

- Name
- PhD/MS/undergraduate student
  - Mention department if not in computer science
- Something interesting/unique about yourself
- *(optional) Why this course?*

This course is

- Seminar course on recent advances in systems for machine learning (SysML)
- Qualifying course for MS/PhD: Computer Systems and Artificial Intelligence
- Work expected and grading:
  - Two to three programming assignments: 30%
  - Class participation:
    - Submit questions/discussion topics on assigned paper readings: 10%
    - Present an overview of one paper (in groups of two): 5%
  - Midterm exam: in class on April 10: 30%
  - Final (group) project: 25%

Course topics

- Introduction to high performance computing (2 weeks)
- Introduction to deep learning (1 week)
- Challenges in high performance DL (1 week)
  - Profiling DL workloads
- Distributed training (1.5 weeks)
- On-node performance optimizations (1 week)
- ML optimizations for systems (1.5 weeks)
- Inference (1 week)
- Data movement and I/O (2 weeks)

Tools we will use for the class

- Syllabus, lecture slides, assignment/project descriptions on course website:
  - https://www.cs.umd.edu/class/spring2025/cmsc828g
- All student submissions will be on gradescope:
  - https://www.gradescope.com/courses/924314
- Discussions on Piazza:
  - https://piazza.com/umd/spring2025/cmsc828g
- If you want to contact the course staff outside of piazza, send an email to: cmsc828g@cs.umd.edu

Zaratan accounts

- Zaratan is the UMD DIT cluster we’ll use for the programming assignments
- You should receive an email when your account is ready for use
- Do NOT use the class allocation for research unrelated to the course
- Helpful resources:
  - https://hpcc.umd.edu/hpcc/help/usage.html
  - https://missing.csail.mit.edu
  - https://www.cs.umd.edu/~mmarsh/books/cmdline/cmdline.html
  - https://www.cs.umd.edu/~mmarsh/books/tools/tools.html

Programming assignments

- You can write and debug most of your assignment locally
  - If you have access to GPUs, you do not need to start on zaratan
- On zaratan:
  - vim, emacs
  - Do not use VSCode to ssh into zaratan
- Eventually, you should ensure that your code runs correctly on zaratan

Excused absence

Any student who needs to be excused for an absence from a single lecture, due to a medically necessitated absence shall make a reasonable attempt to inform the instructor of his/her illness prior to the class. Upon returning to the class, present the instructor with a self-signed note attesting to the date of their illness. Each note must contain an acknowledgment by the student that the information provided is true and correct. Providing false information to University officials is prohibited under Part 9(i) of the Code of Student Conduct (V-1.00(B) University of Maryland Code of Student Conduct) and may result in disciplinary action.

Self-documentation may not be used for Major Scheduled Grading Events (midterm exam, project presentation) and it may only be used for one class meeting during the semester. Any student who needs to be excused for a prolonged absence (two or more consecutive class meetings), or for a Major Scheduled Grading Event, must provide written documentation of the illness from the Health Center or from an outside health care provider. This documentation must verify dates of treatment and indicate the timeframe that the student was unable to meet academic responsibilities. In addition, it must contain the name and phone number of the medical service provider to be used if verification is needed. No diagnostic information will ever be requested.

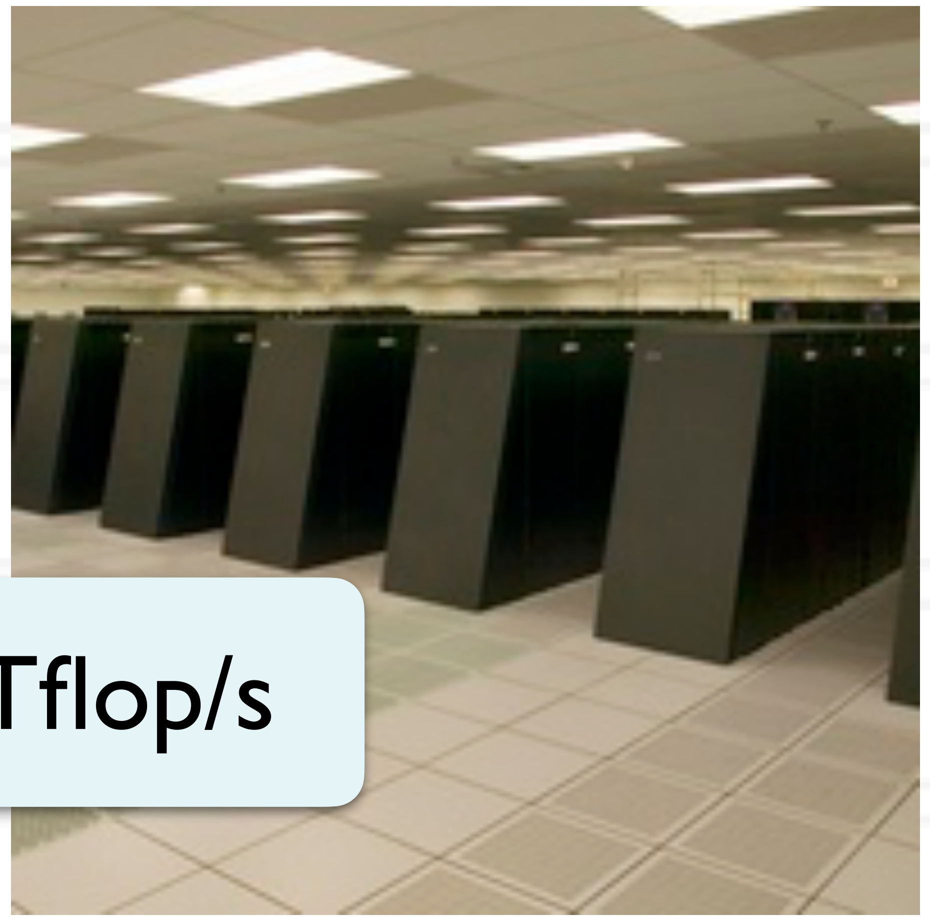
Use of LLMs

AI assistance (ChatGPT, Copilot, DALL-E, etc.) is not permitted for coding, writing, editing, or any other part of the class participation tasks and programming assignments. Even though we expect you will use these tools in the future, this approach will help you build a solid understanding of the subject matter, which will benefit your future career.

You can use AI tools such as ChatGPT as you would use Google for research. However, you cannot generate your solutions using ChatGPT. You must demonstrate independent thought and effort. If you use any AI tools for anything class related, you must mention that in your answer/report. Please note that LLMs provide unreliable information, regardless of how convincingly they do so. If you are going to use an LLM as a research tool in your submission, you must ensure that the information is correct and addresses the actual question asked.

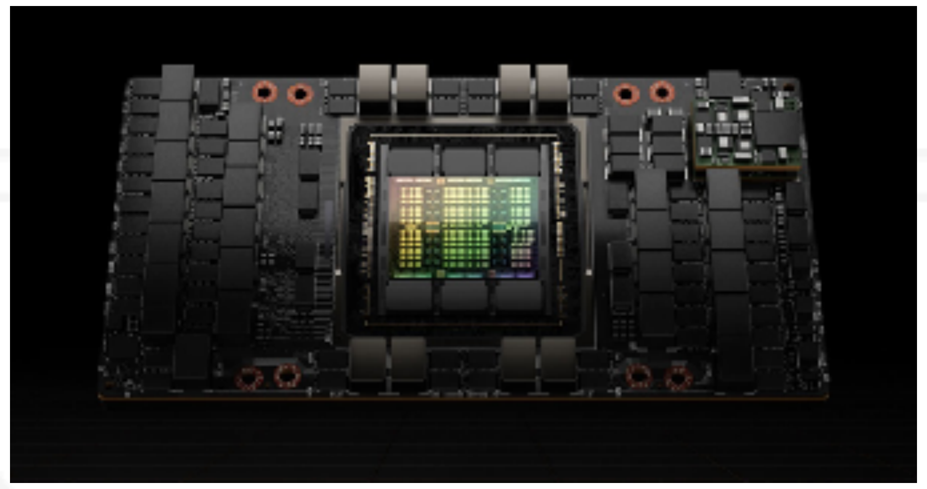
The evolution of HPC systems and rise of a new revolution in AI

- In the last two decades, an enormous amount of compute power has become available
- Large datasets and open source software such as PyTorch have also emerged
- Led to a frenzy in the world of AI and the effects are being felt in almost every other domain



Top500 Rpeak - 91.75 Tflop/s

IBM Blue Gene/L, 2004

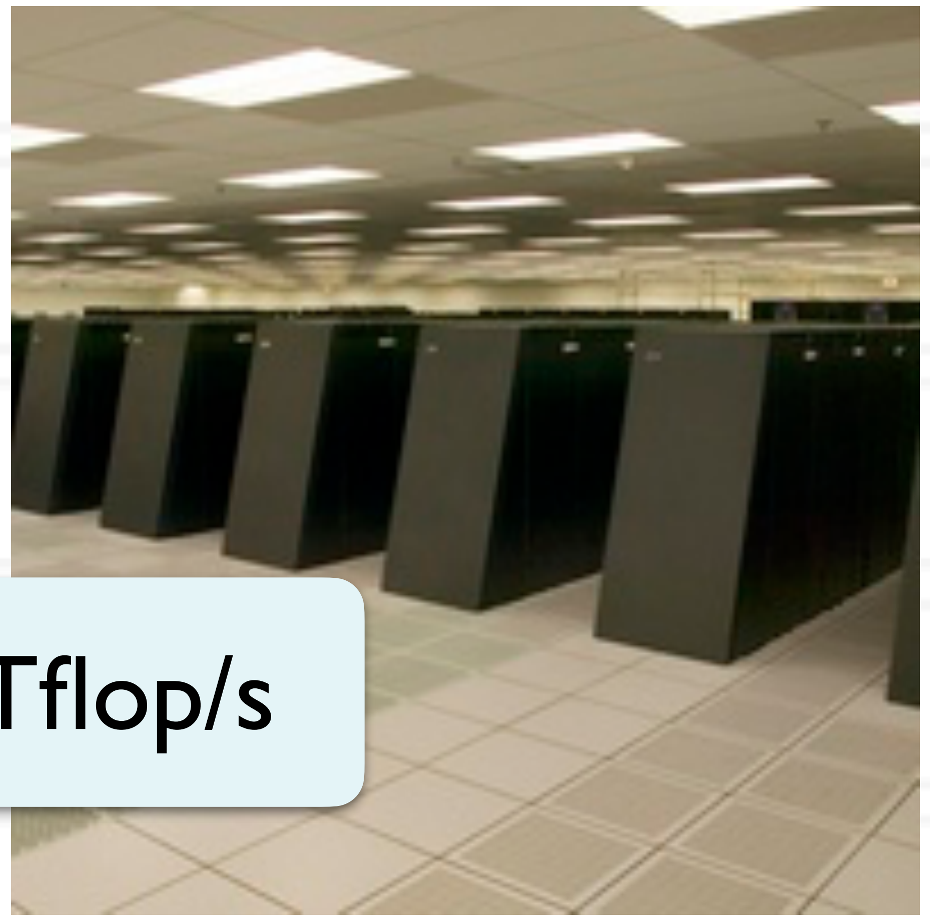


FP64 - 34 Tflop/s

NVIDIA H100, 2024

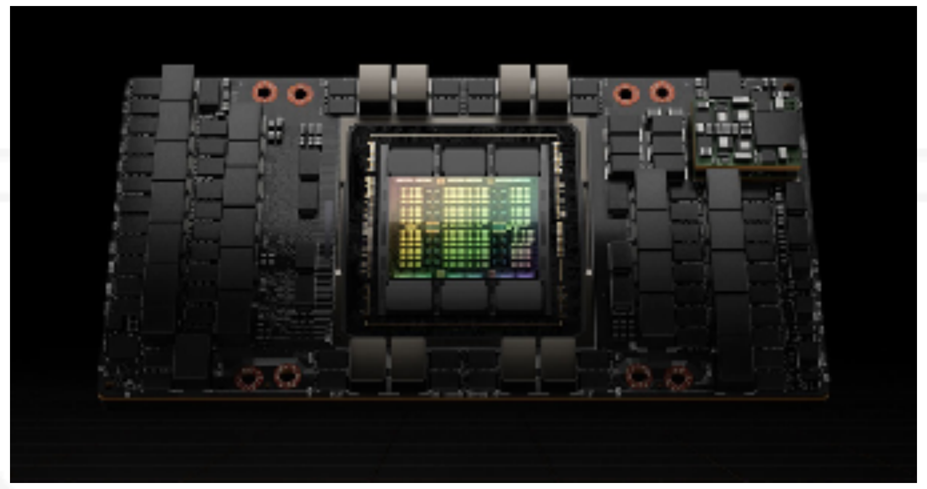
The evolution of HPC systems and rise of a new revolution in AI

- In the last two decades, an enormous amount of compute power has become available
- Large datasets and open source software such as PyTorch have also emerged
- Led to a frenzy in the world of AI and the effects are being felt in almost every other domain



Top500 Rpeak - 91.75 Tflop/s

IBM Blue Gene/L, 2004

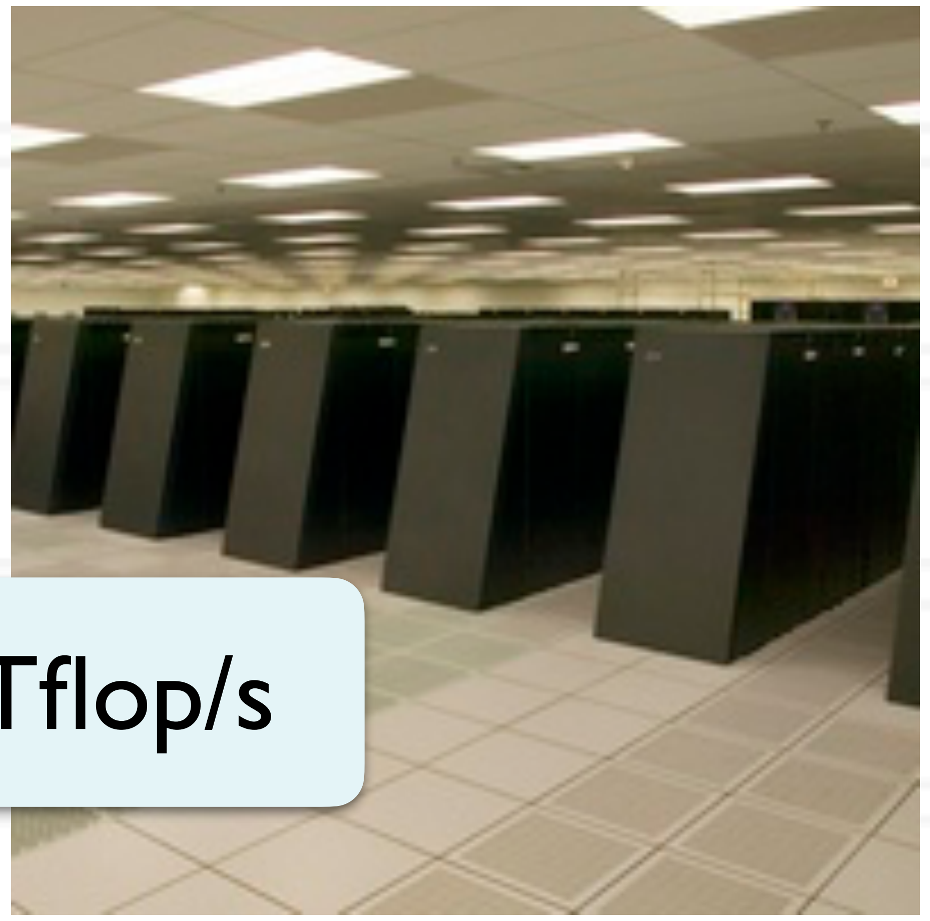


FP16 - 989 Tflop/s

NVIDIA H100, 2024

The evolution of HPC systems and rise of a new revolution in AI

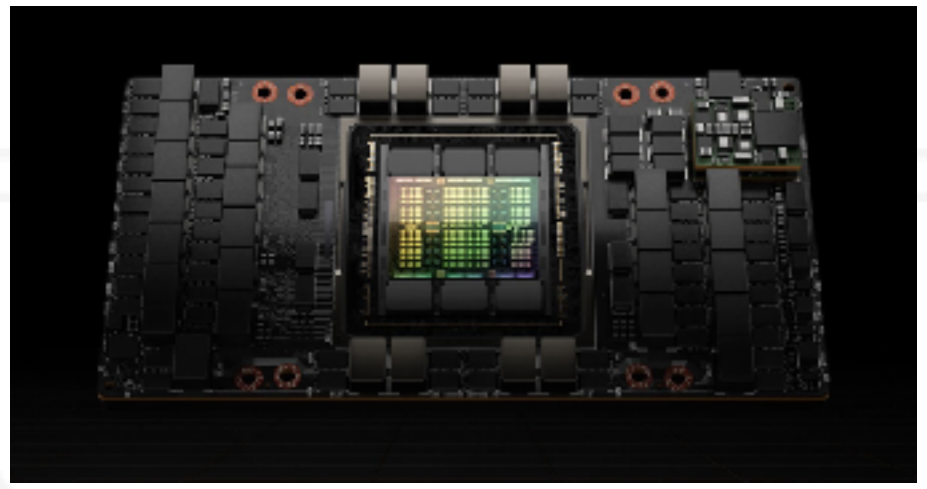
- In the last two decades, an enormous amount of compute power has become available
- Large datasets and open source software such as PyTorch have also emerged
- Led to a frenzy in the world of AI and the effects are being felt in almost every other domain



Top500 Rpeak - 91.75 Tflop/s

IBM Blue Gene/L, 2004

10.63 Exaflop/s!!



NVIDIA H100, 2024

A timeline of evolution of LLMs

○ Papers
● Open Source Models
● Commercial Models
SENNNA
word2vec
Transformer model
Bert
GPT-2
GPT-3
Lamda
GPT - Neox
Chinchilla
InstructGPT
PaLM
BLOOM
ChatGPT
MED-PaLM
RT-1
Jasper
LLaMA
PaLM-E
GPT-4
MED-PaLM2
Vicuna
Alpaca
Dolly
Bard
Cerebras
Koala
Open Assistant
GPT4All
Guanaco
Falcon
Llama2
Claude 2
GPT4 Turbo
2011
2013
2017
2018
2019
June 2020
January 2021
March 2021
June 2021
Sept 2021
January 2022
March 2022
June 2022
Sept 2022
January 2023
March 2023
June 2023
Sept 2023

https://www.ml6.eu/resources/large-language-models

Computer systems and ML

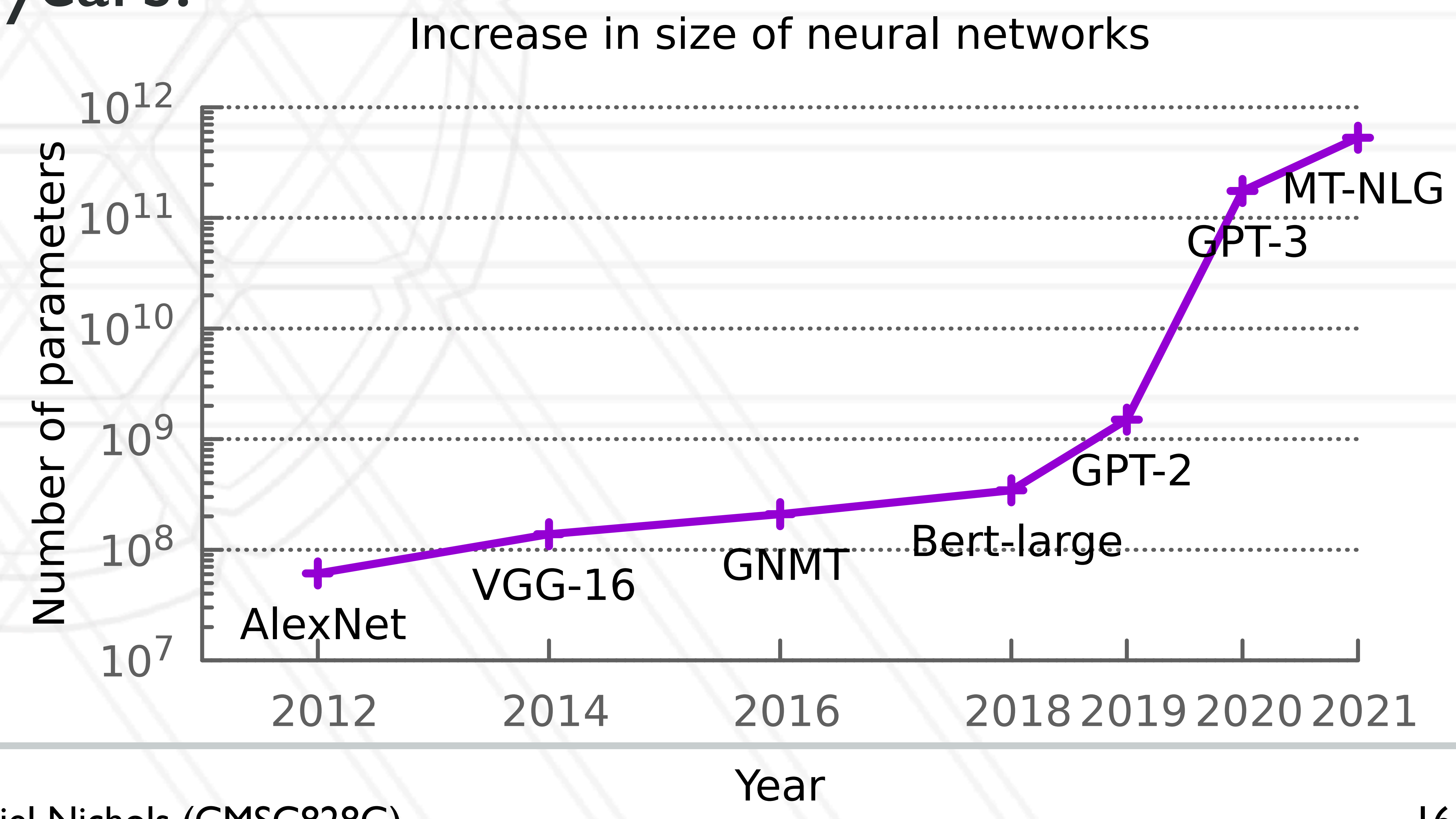
- Architecture: GPUs, TPUs, …
- Memory management and optimization
- Networks: communication on data center networks, cloud servers, HPC systems
- Storage: File input/output
- Performance engineering and optimization: compute kernels

Parallel computing and ML

- Large models do not fit on a single GPU
- Training: with a large amount of data, it can take too long on a single GPU
- Inference: large models and/or serving a large number of users can require multiple GPUs

Do we really need parallel resources?

- The largest model you can run on an H100 96 GB GPU is around 3.5-4 billion parameters
- On a single node (with four H100 GPUs): around ~16 billion parameters model
- Training a 16B parameter would take 33 years!
- OpenAI's GPT 4.0 is estimated to have 1.8 trillion parameters
- Meta's Llama-3.1-405B has more than 400 billion parameter

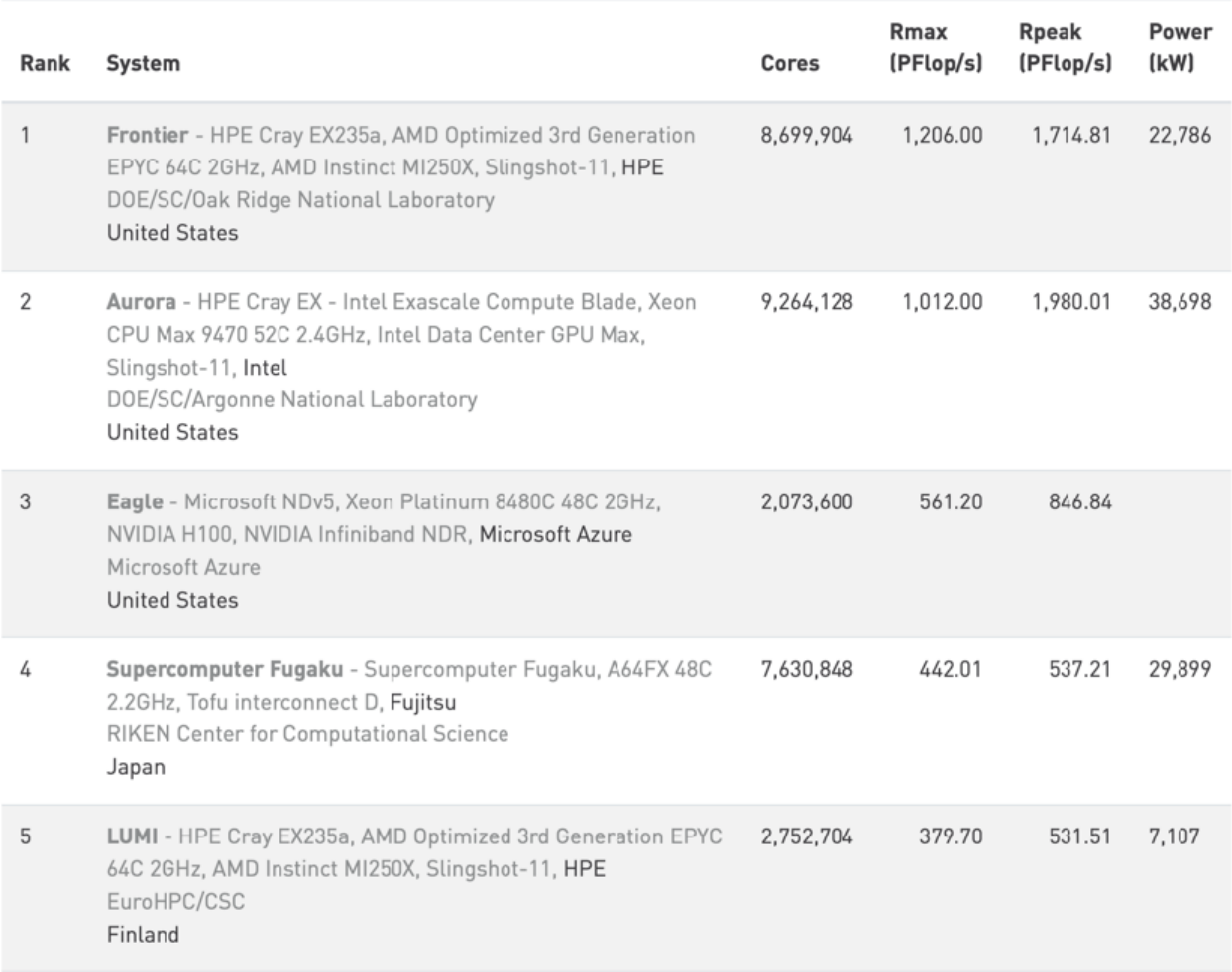


Terms and definitions

- Model training: process of adjusting a model's parameters using input data (and correct output) to accurately predict the output for unseen data
- Inference: using a trained model to make predictions for new inputs
- Fine-tuning: starting with a pre-trained model and adapt it to a specific task

Large supercomputers

- Top500 list: https://top500.org/lists/top500/2024/06/

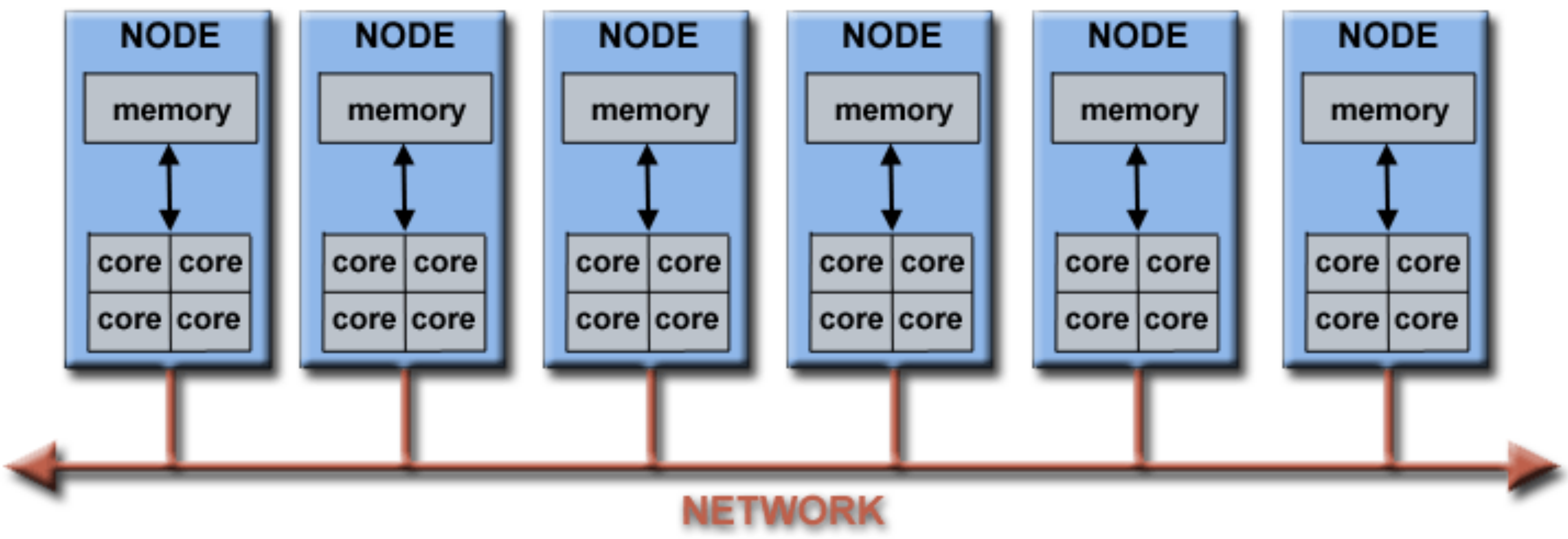


Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,206.00	1,714.81	22,786
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	
4	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107

https://www.olcf.ornl.gov/frontier

Architecture of a parallel cluster

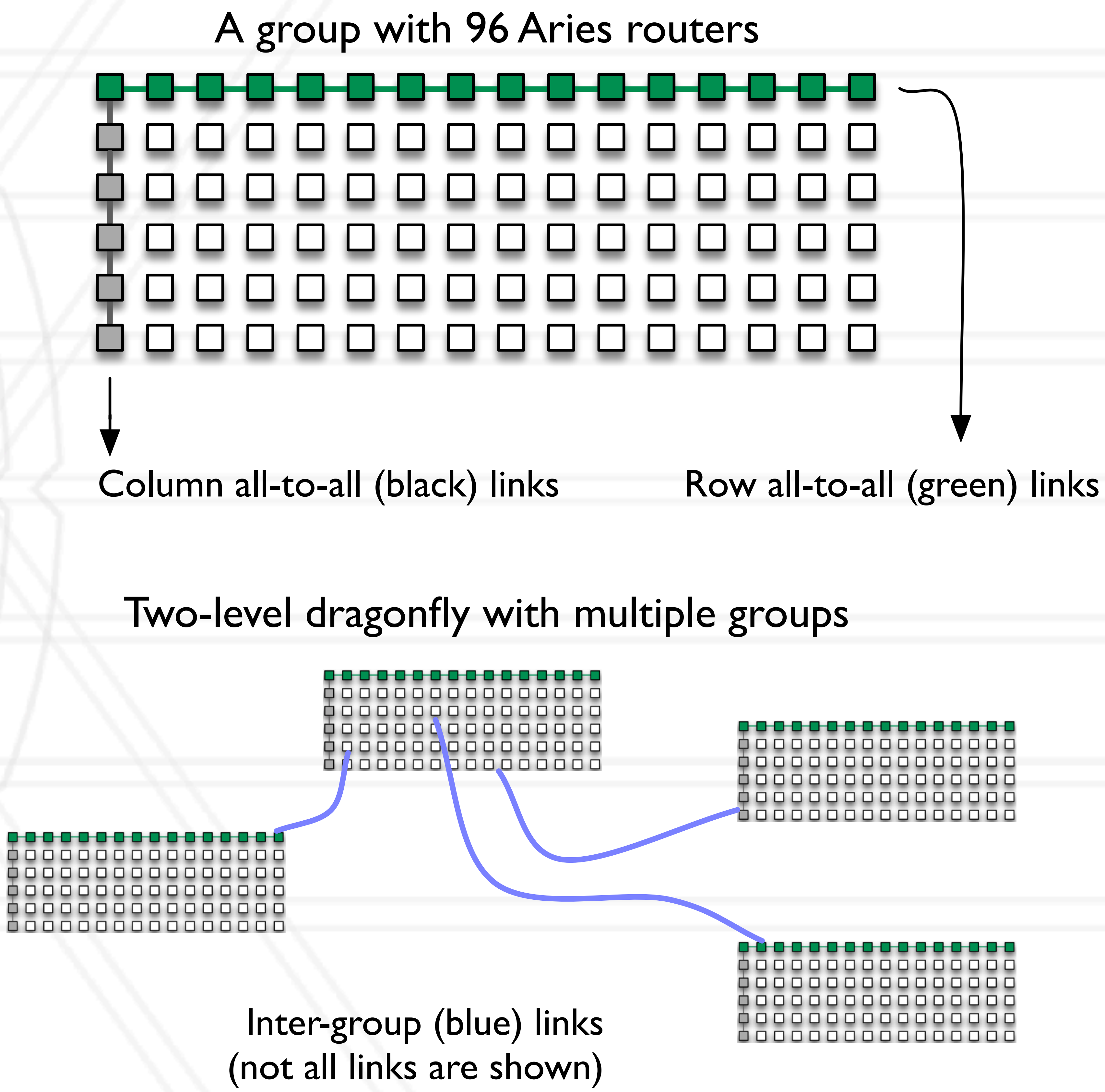
- A set of nodes or processing elements connected by a network.



https://computing.llnl.gov/tutorials/parallel_comp

Interconnection networks

- Different topologies for connecting nodes together
- Used in the past: torus, hypercube
- More popular currently: fat-tree, dragonfly



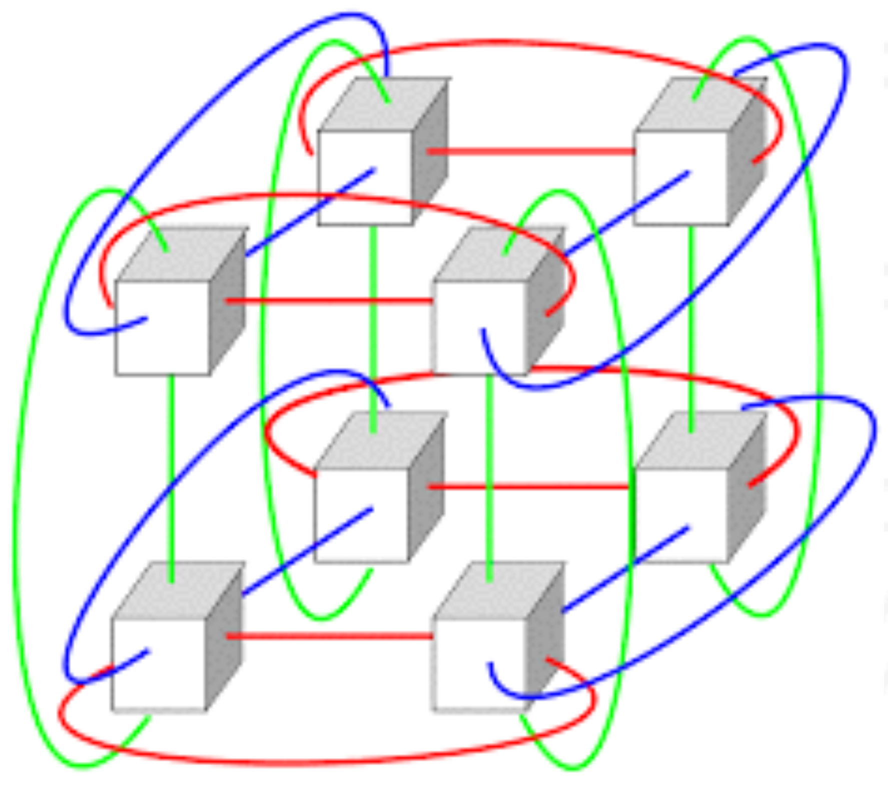
A group with 96 Aries routers

Column all-to-all (black) links

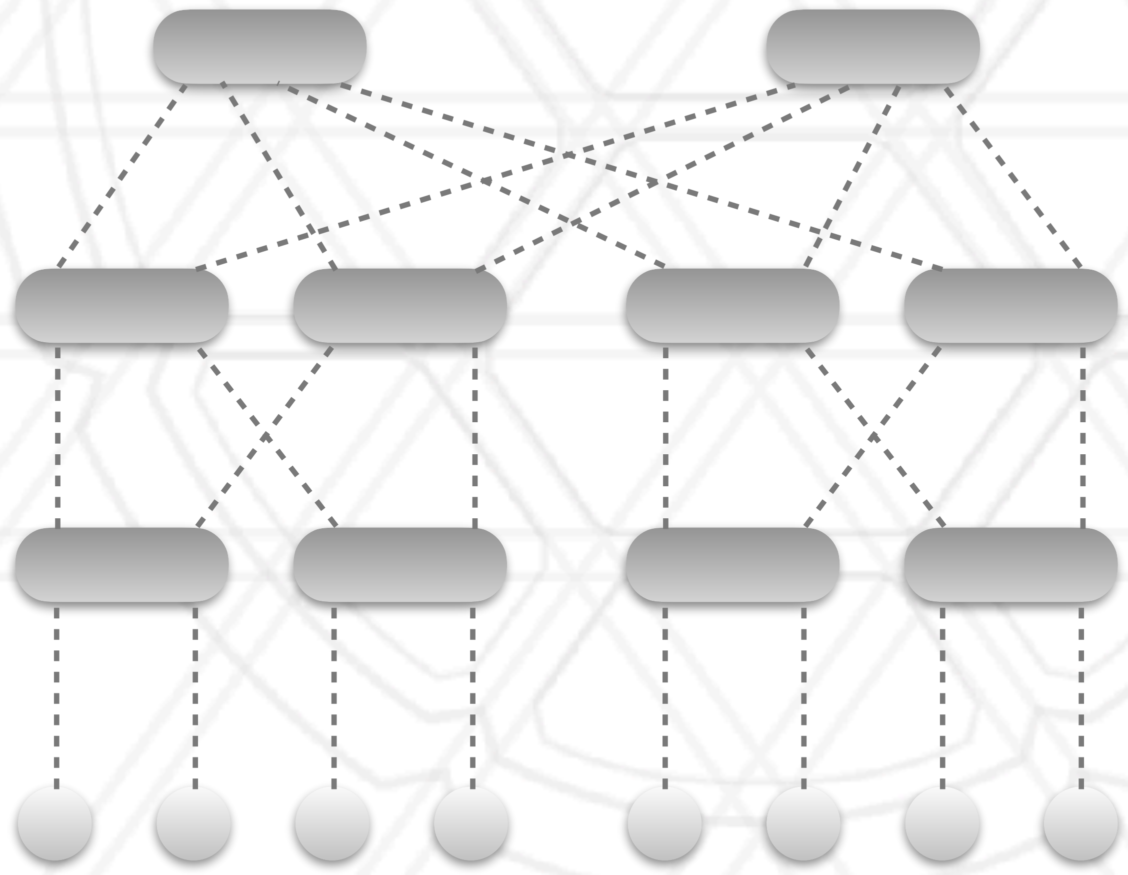
Row all-to-all (green) links

Two-level dragonfly with multiple groups

Inter-group (blue) links
(not all links are shown)



Torus

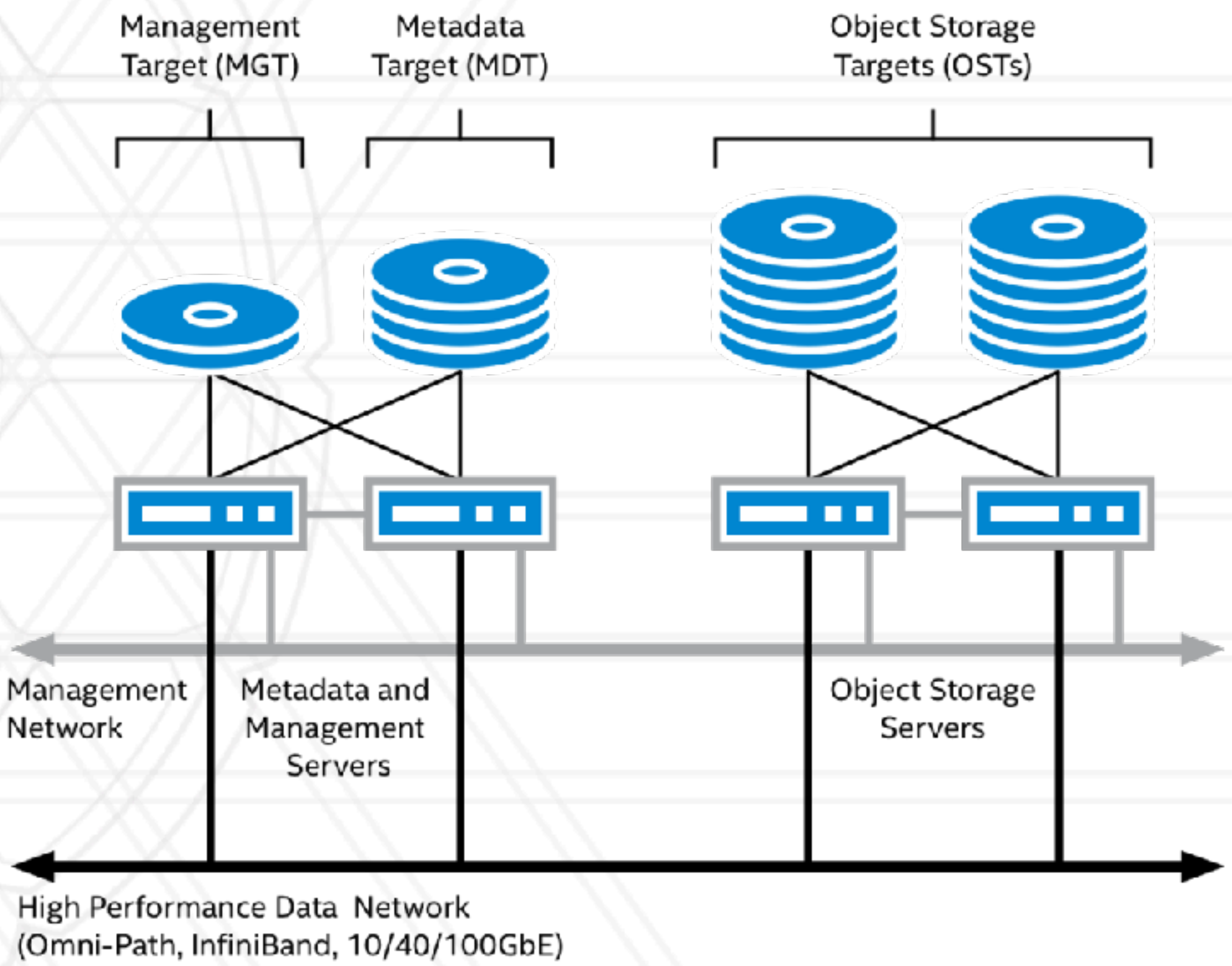


Fat-tree

Dragonfly

I/O sub-system / Parallel file system

- Home directories and scratch space on clusters are typically on a parallel file system
- Compute nodes do not have local disks
- Parallel filesystem is mounted on all login and compute nodes

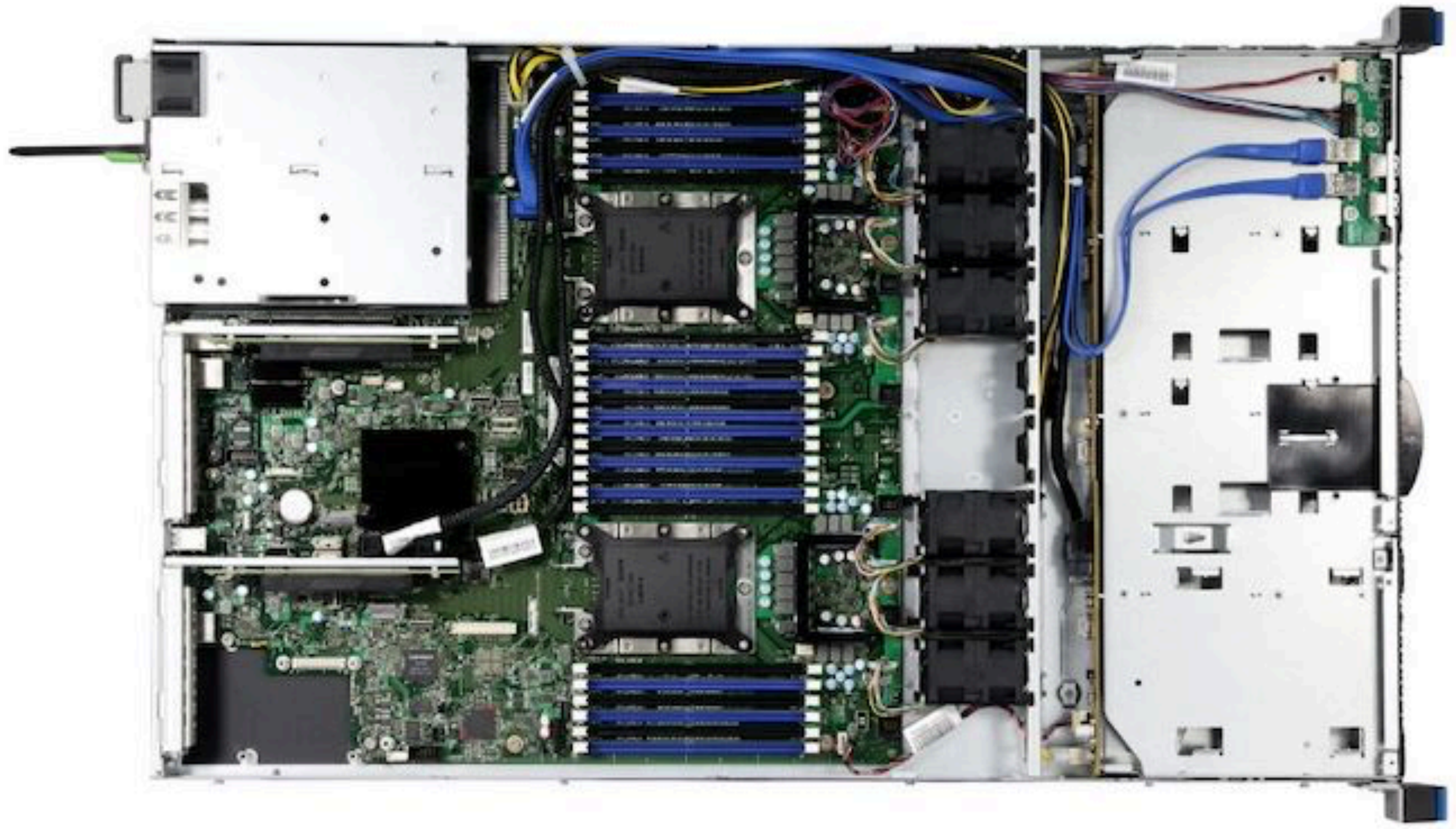

Management Target (MGT)
Metadata Target (MDT)
Object Storage Targets (OSTs)
Management Network
Metadata and Management Servers
Object Storage Servers
High Performance Data Network (Omni-Path, InfiniBand, 10/40/100GbE)

http://wiki.lustre.org/Introduction_to_Lustre

Rackmount servers

Rackmount servers

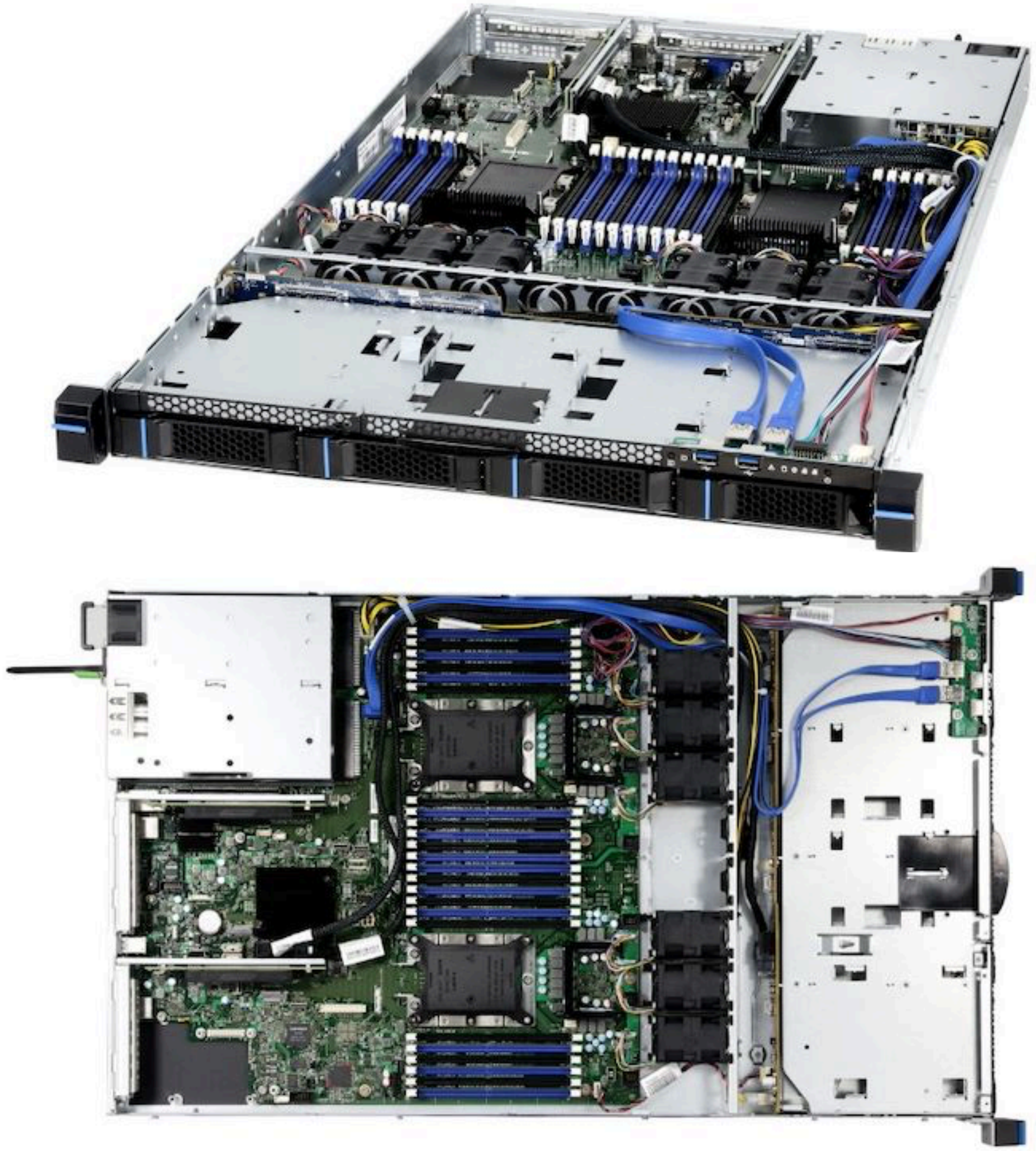
Rackmount server motherboard



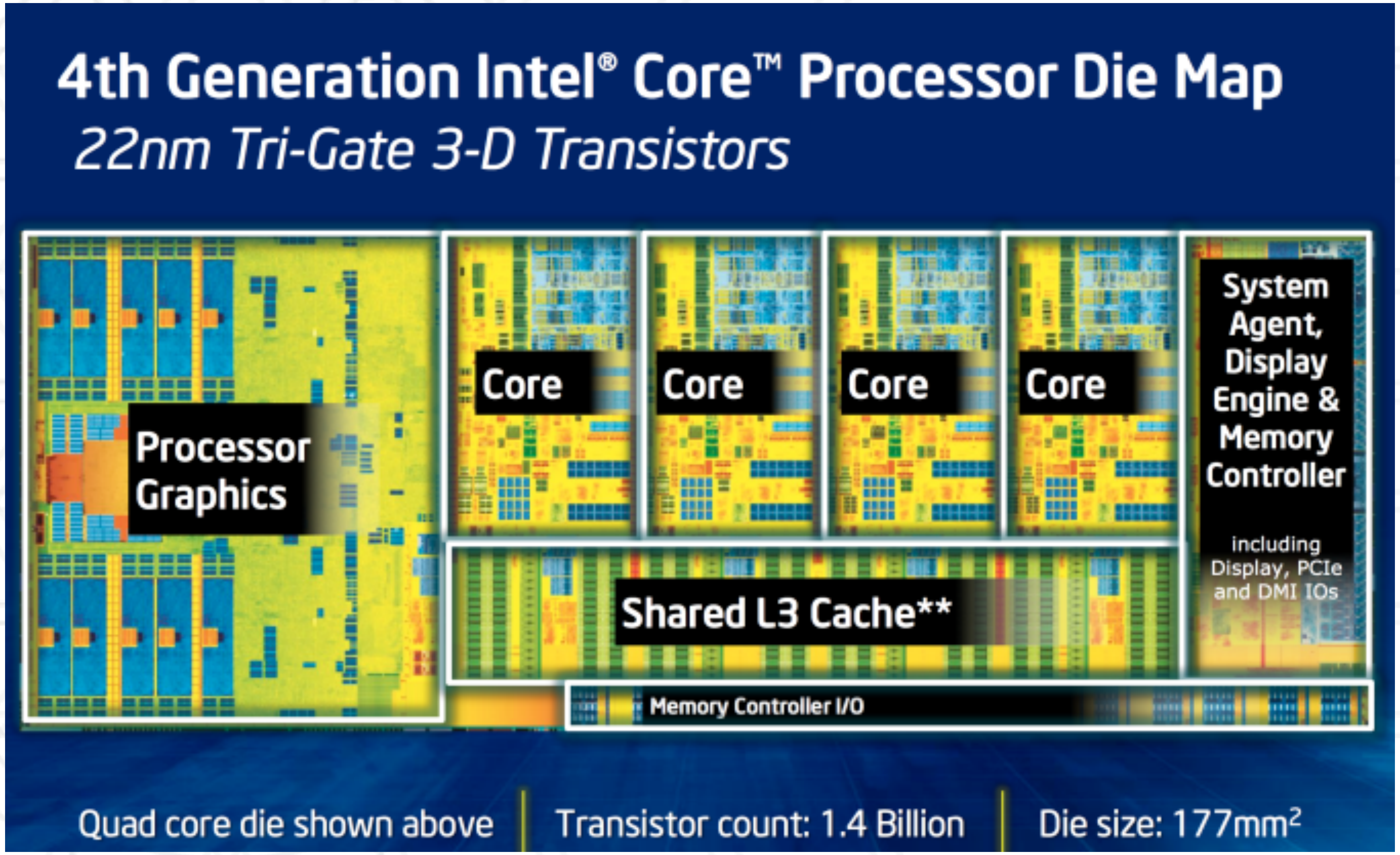
https://www.anandtech.com/show/15924/chenbro-announces-rb13804-dual-socket-1u-xeon-4-bay-hpc-barebones-server

https://www.anandtech.com/show/7003/the-haswell-review-intel-core-i74770k-i54560k-tested

Rackmount server motherboard



https://www.anandtech.com/show/15924/chenbro-announces-rb13804-dual-socket-1u-xeon-4-bay-hpc-barebones-server



4th Generation Intel® Core™ Processor Die Map

22nm Tri-Gate 3-D Transistors

Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O

Quad core die shown above | Transistor count: 1.4 Billion | Die size: 177mm^2

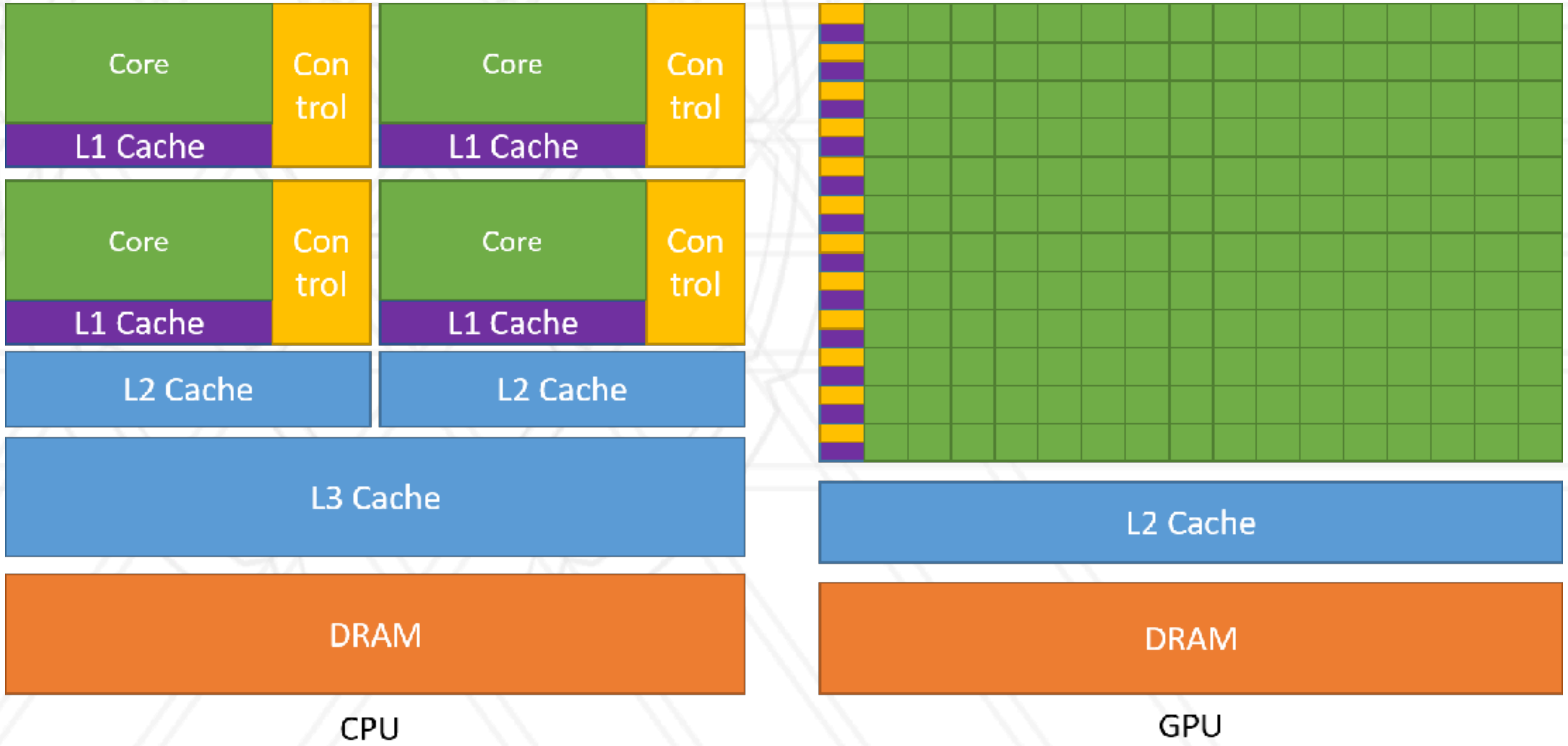
https://www.anandtech.com/show/7003/the-haswell-review-intel-core-i74770k-i54560k-tested

GPGPUs

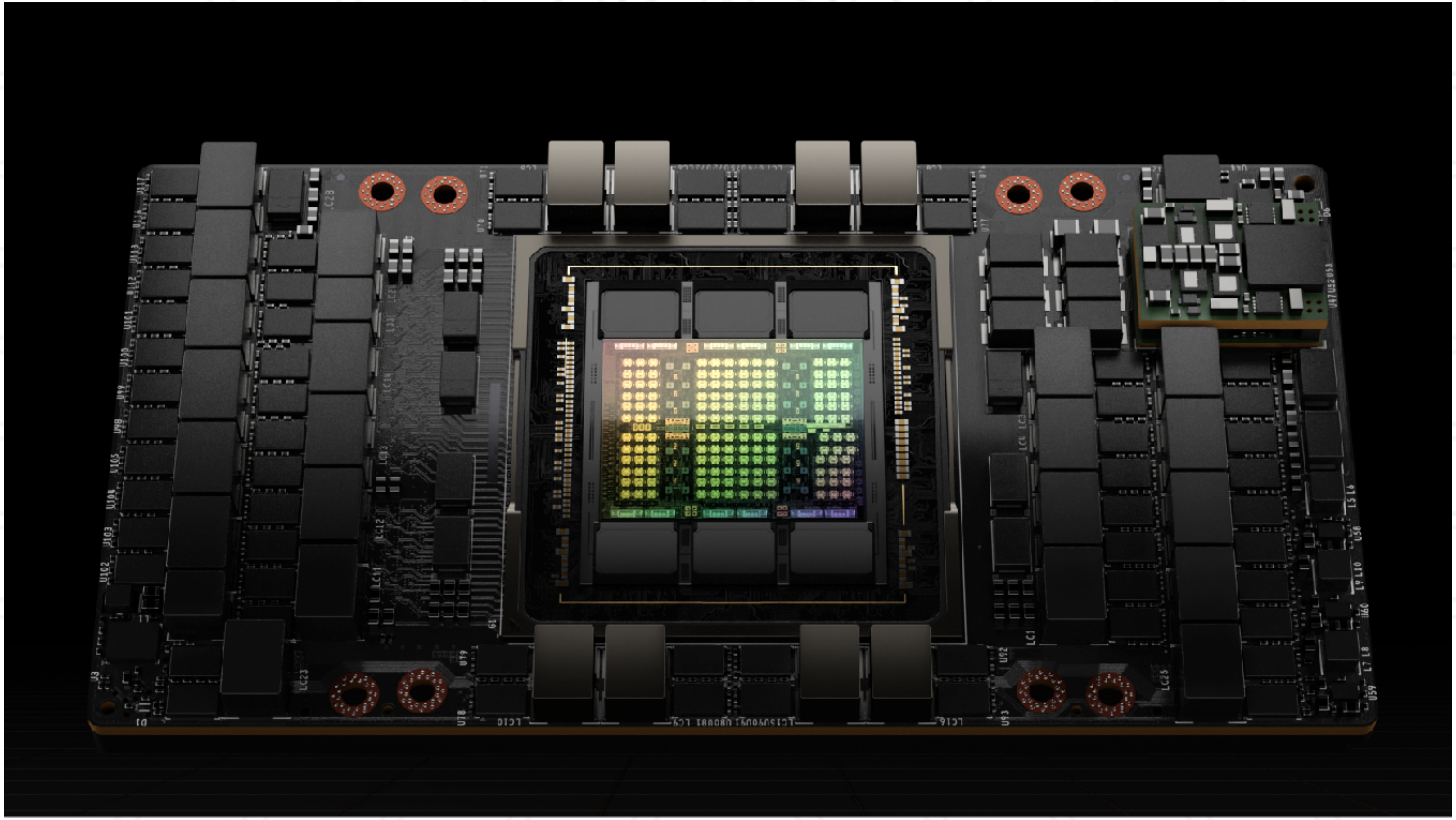
- Originally developed to handle computation related to graphics processing
- Also found to be useful for scientific computing and AI
- Hence the name: General Purpose Graphics Processing Unit

GPGPU Hardware

- Higher instruction throughput
- Hide memory access latencies with computation



NVIDIA H100 chip



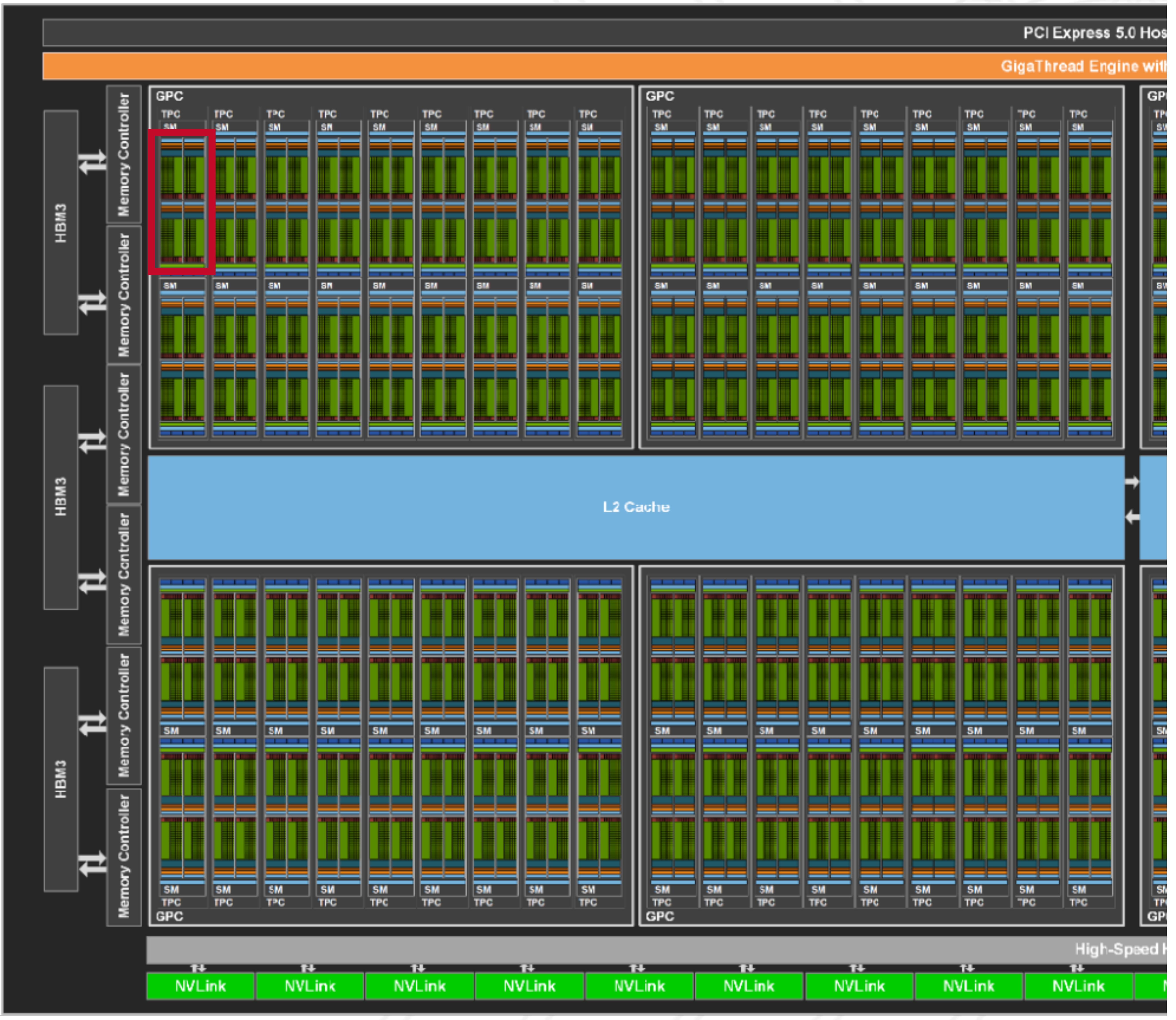
NVIDIA H100 chip

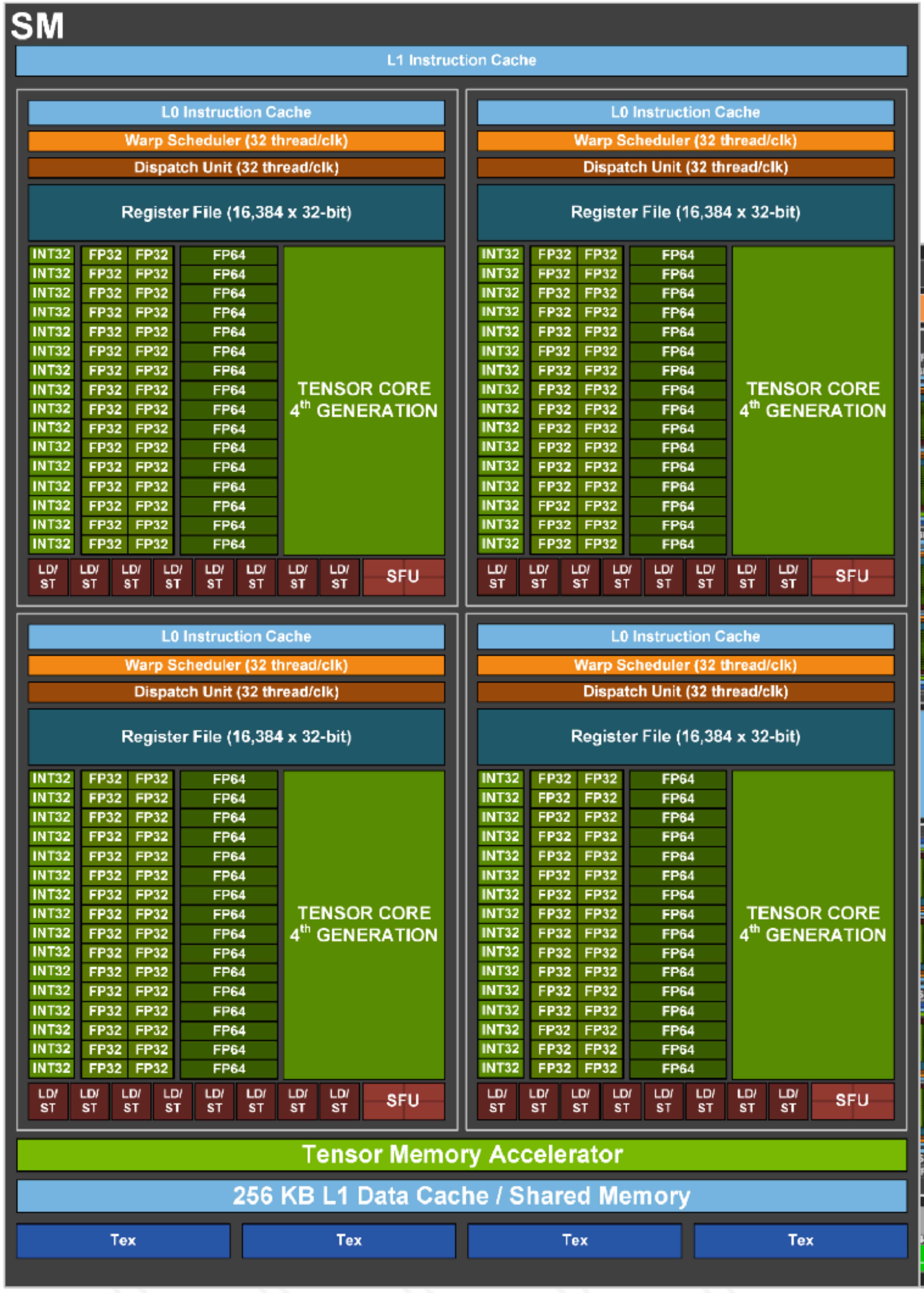
PCI Express 5.0 Host Interface
GigaThread Engine with MIG Control
GPC
TPC
SM
L2 Cache
L2 Cache
Memory Controller
HBM3
High-Speed Hub
NVLink

NVIDIA H100 chip

PCI Express 5.0 Host Interface
GigaThread Engine with MIG Control
GPC
TPC
SM
Memory Controller
HBM3
L2 Cache
High-Speed Hub
NVLink

NVIDIA H100 chip




SM
L1 Instruction Cache
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32
FP32
FP64
TENSOR CORE 4th GENERATION
LD/ ST
SFU
Tensor Memory Accelerator
256 KB L1 Data Cache / Shared Memory
Tex

Terms and definitions

- Model: an overloaded term
- Network architecture: also an overloaded term
- Weights / parameters: floating point numbers that represent the model
  - Used to denote the size of the model

Terms and definitions

- Language model: trained on natural language data for natural language tasks
  - LLMs, Transformer models
- Image model: trained on image data for tasks dealing with images
  - CNNs, ViTs, diffusion models
- Graph neural networks: trained on graph data

Group project

UNIVERSITY OF
MARYLAND