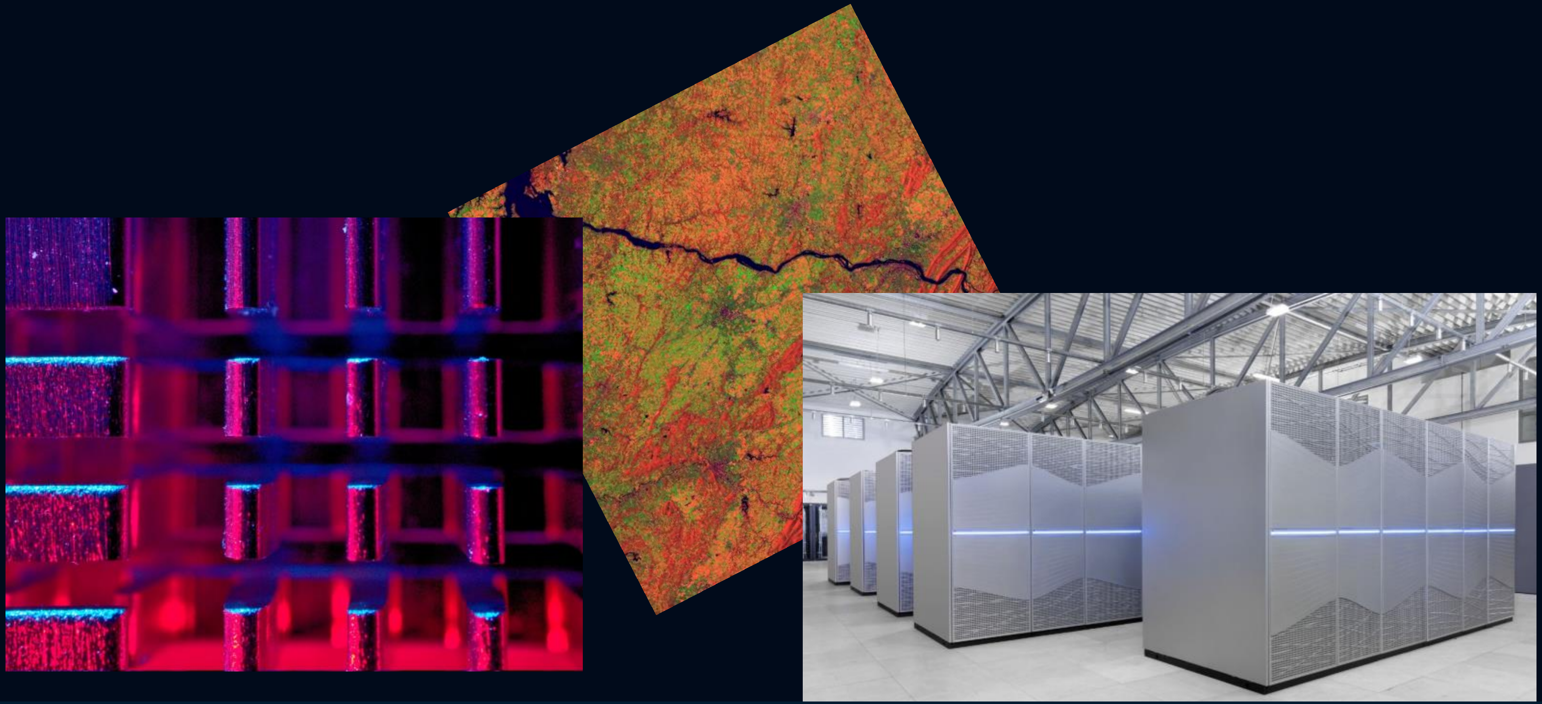


LEVELS OF PARALLELISM AND HIGH PERFORMANCE COMPUTING

IEEE IGARSS 2021 Tutorial on Scalable Machine Learning with High Performance and Cloud Computing

DR. – ING. GABRIELE CAVALLARO
HIGH PRODUCTIVITY DATA PROCESSING RESEARCH GROUP
JÜLICH SUPERCOMPUTING CENTRE
WWW.GABRIELE-CAVALLARO.COM

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE   UNIVERSITY OF ICELAND   HDCRS   ESI   GRSS   igarss BRUSSELS 2021

IT ENVIRONMENTS

Key Priorities

Embedded system

Power consumption and price

Desktop Computing

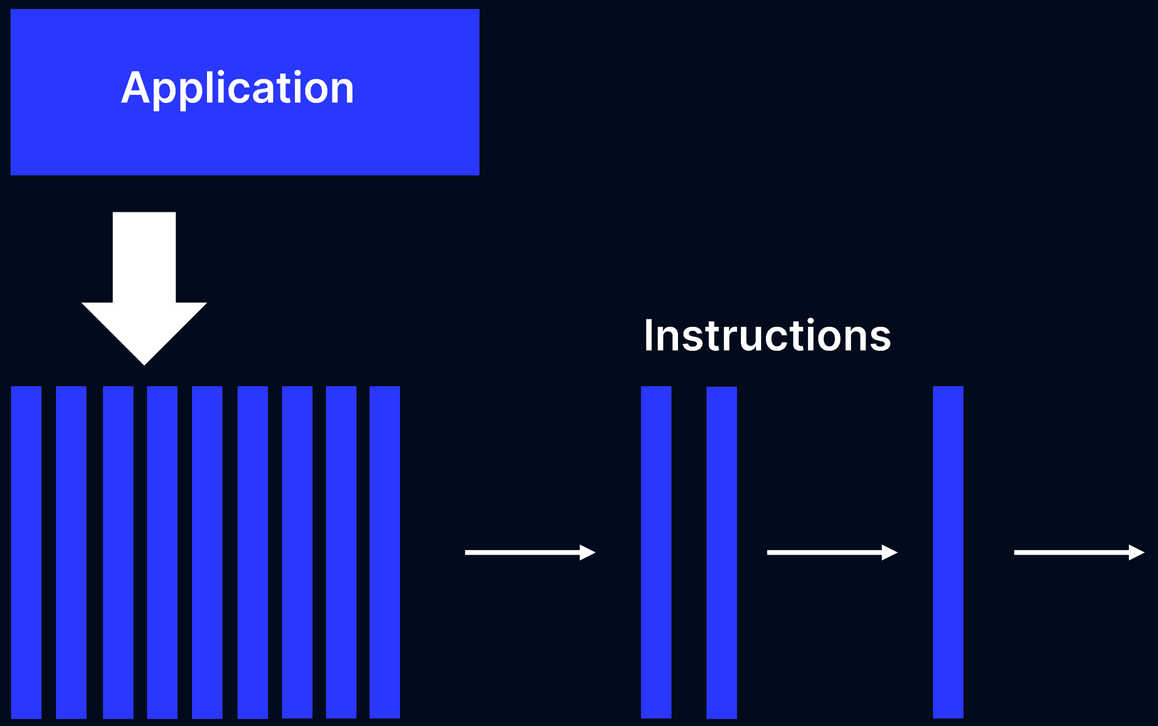
Performance / price ration

Servers

Maximum performance and maintainability reliability

HOW TO BECOME FASTER?

Split into instructions



HOW TO BECOME FASTER?

Three Alternative Ways

Work Harder

Clock Speed

Work Smarter

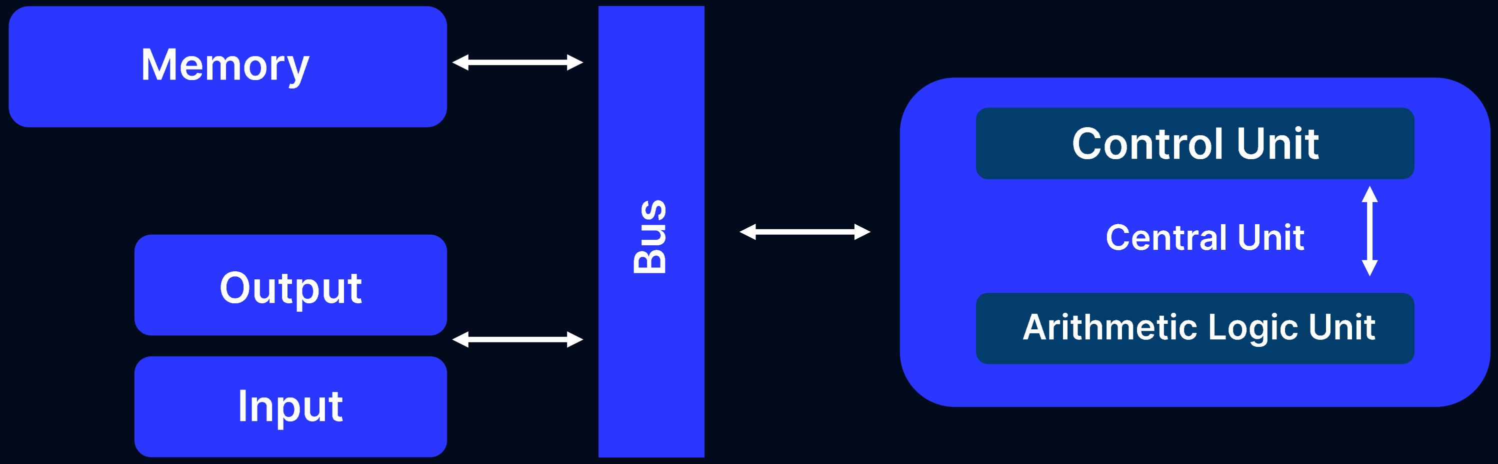
Optimization, Caching

Get Help

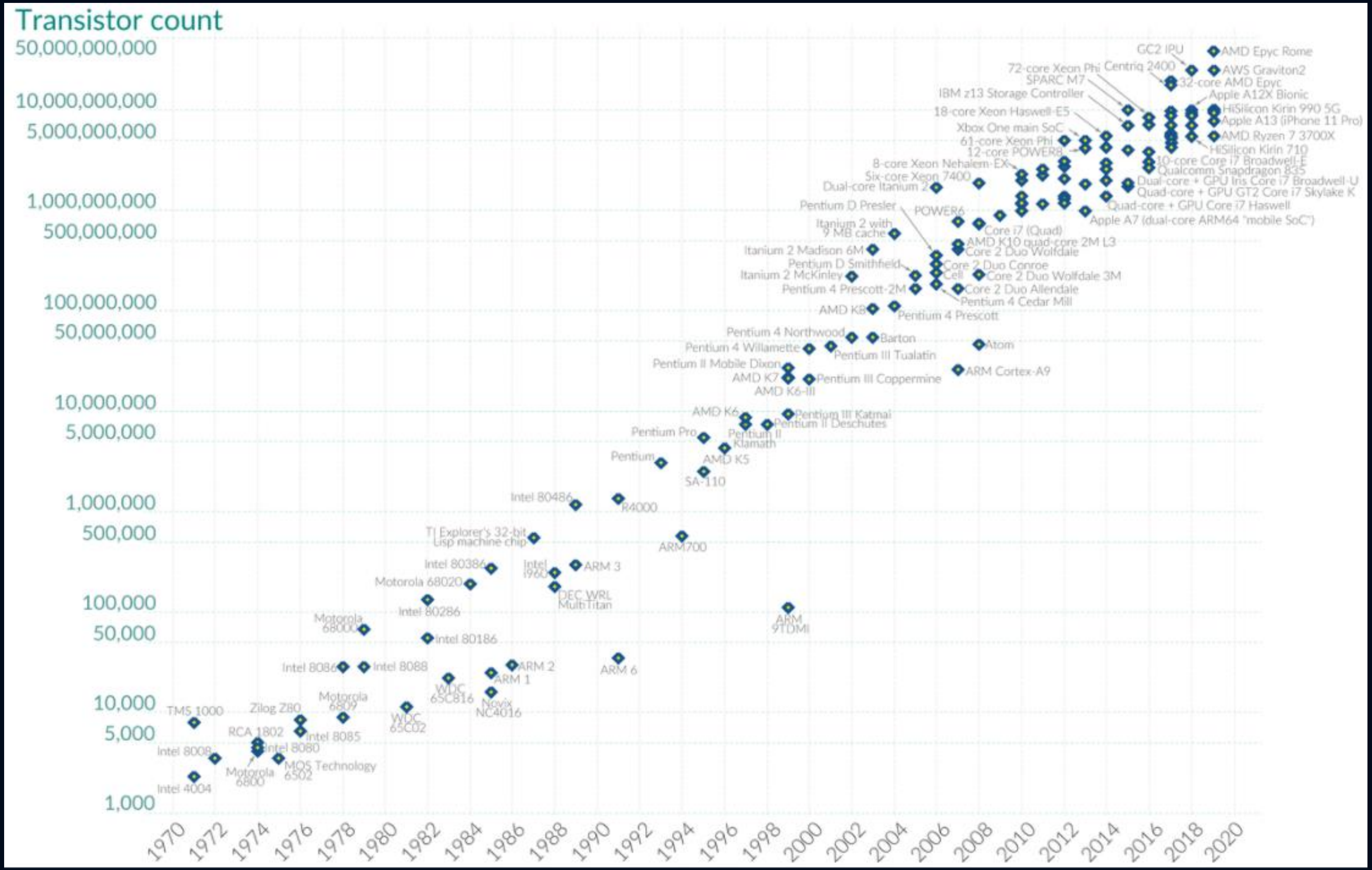
Parallelization

VON NEUMANN ARCHITECTURE

Machine Model



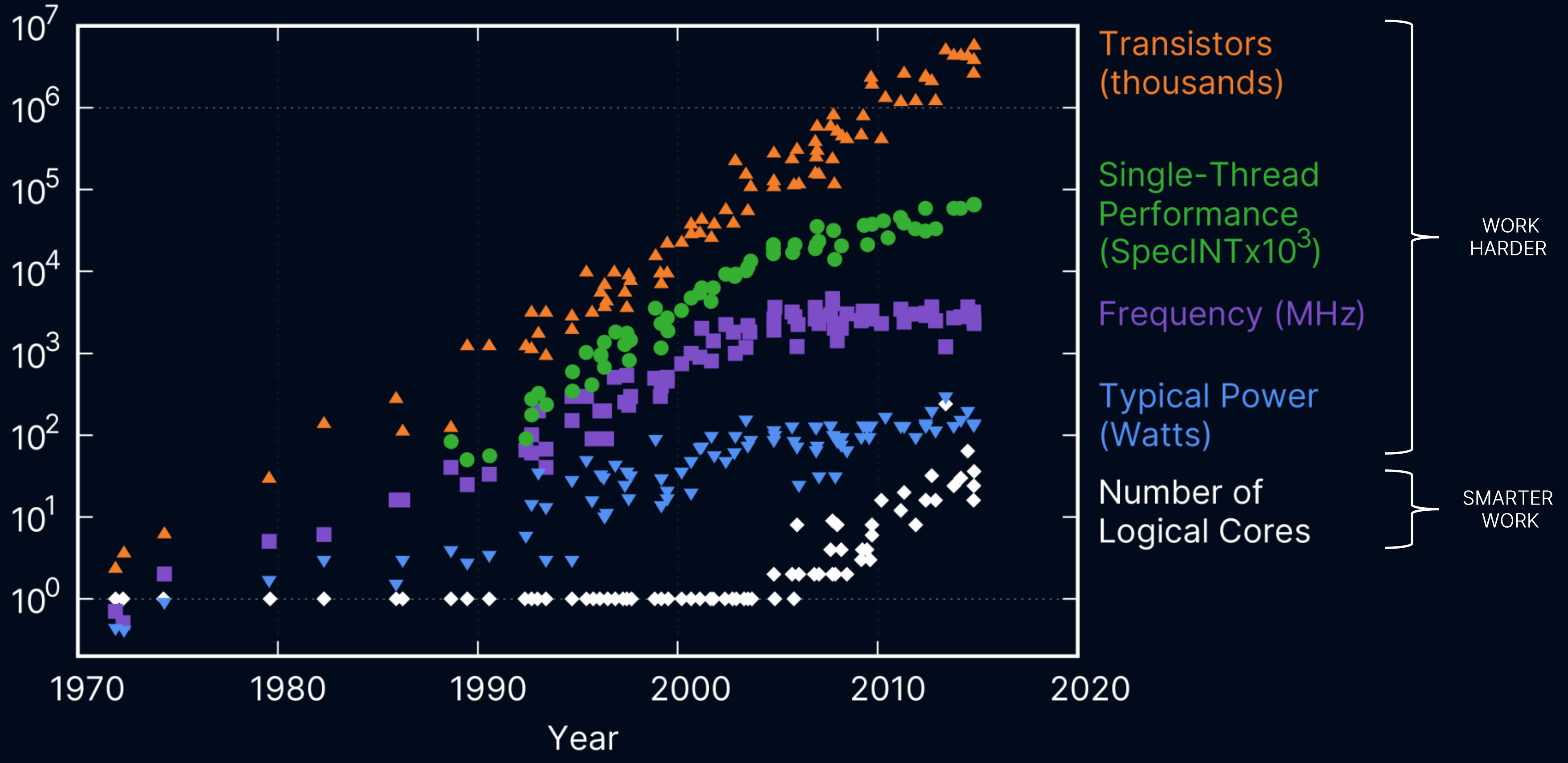
MOORE'S LAW



Wikipedia: Transistor count

THE END OF DENNARD SCALING

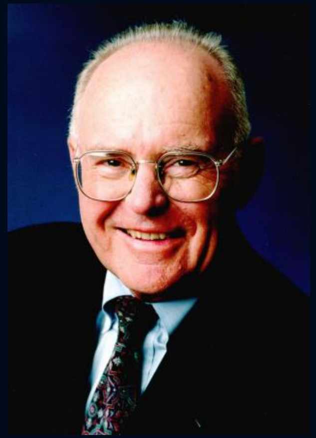
Why haven't clock speeds increased, even though transistors have continued to shrink?



Karl Rupp, 40 Years of Microprocessor Trend Data
https://www.karlrupp.net/2015/06/40-years-of-microprocessor-trend-data/

WILL MOORE'S LAW END?
Paradigm Shift

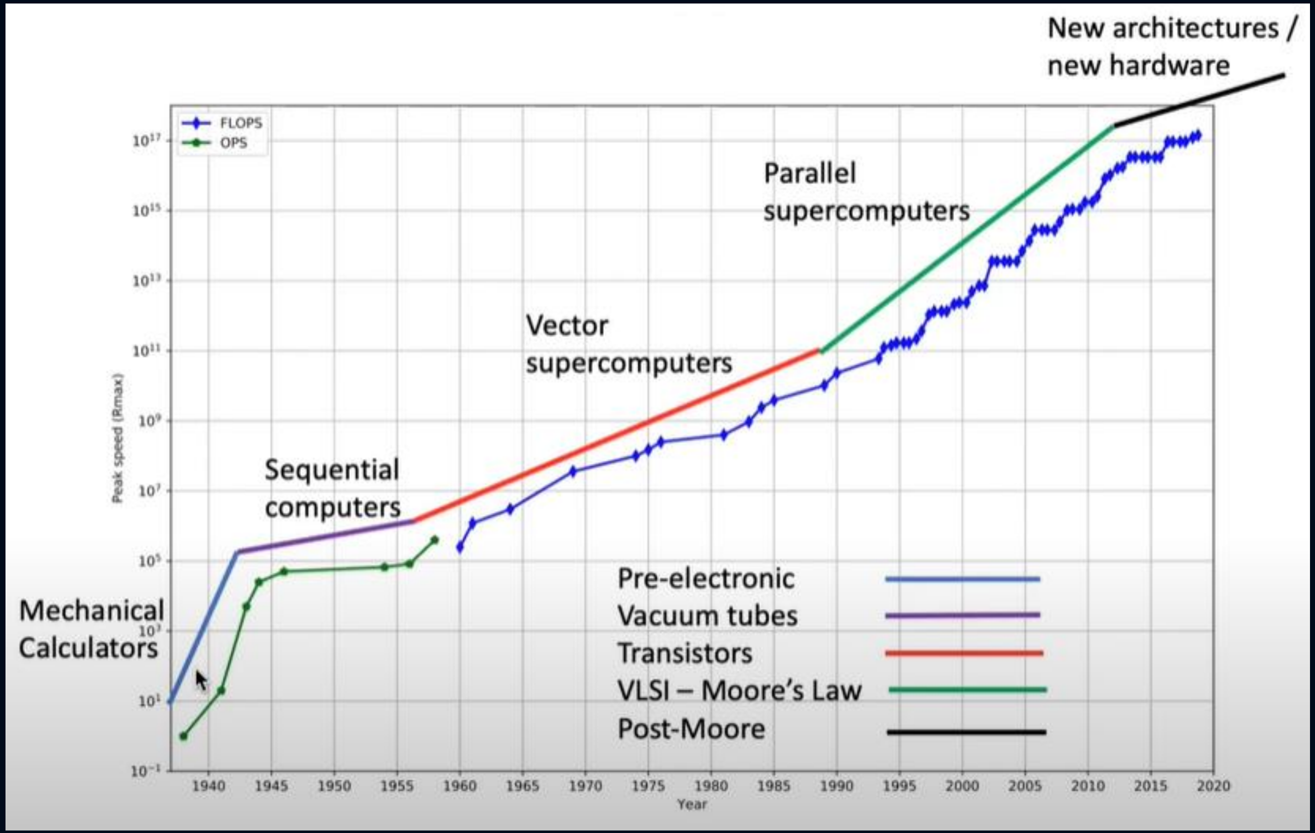
"There's no getting around the fact that we build these things out of atoms"



Gordon Moore

WHY SUPERCOMPUTER PERFORMANCE KEEP INCREASING?

Parallel computing is going mainstream

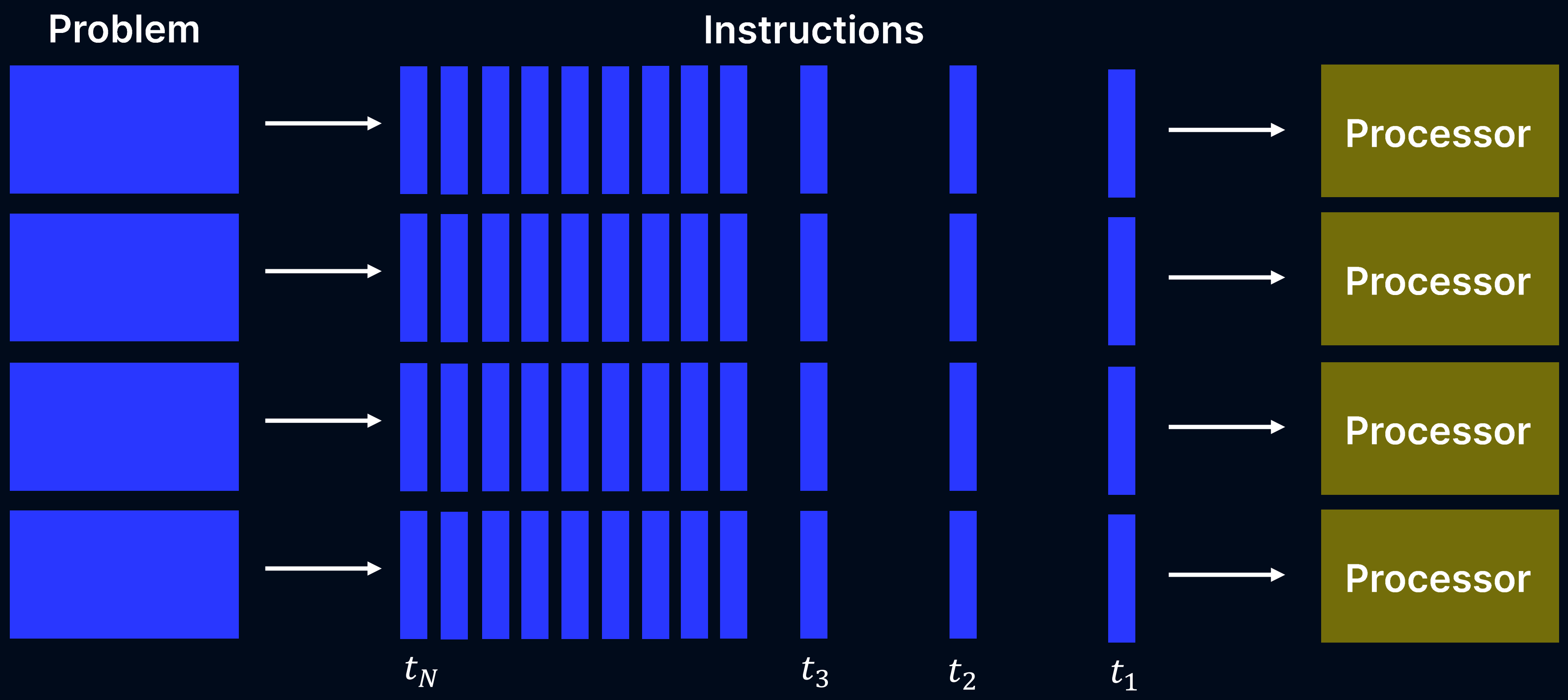


Rob Schreiber, High Performance Computing: Beyond Moore's Law
https://www.youtube.com/watch?v=LOf57fdxln4&t=258s

THE EASY TIMES HAVE GONE

Responsibility for better performance is on the software developers

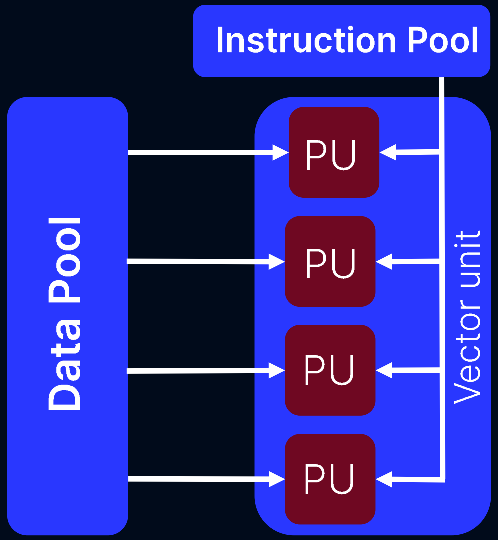
PARALLEL COMPUTING



SINGLE INSTRUCTION MULTIPLE DATA (SIMD)

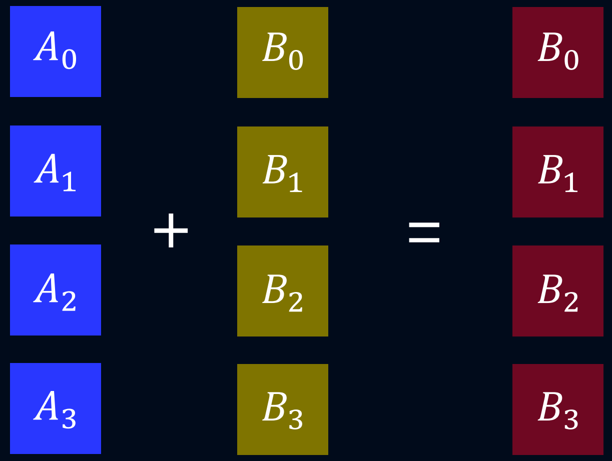
In-core parallelism

SIMD



Parallel execution within
a single CPU core

Operation with Vectors

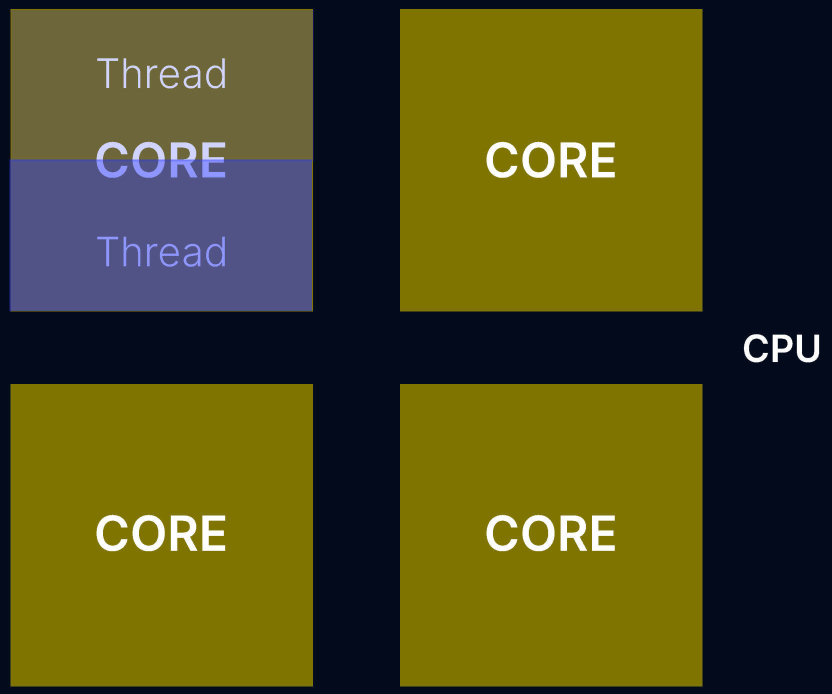


Elements of the vector
are sum in parallel

SIMULTANEOUS MULTITHREADING (SMT)

In-core parallelism

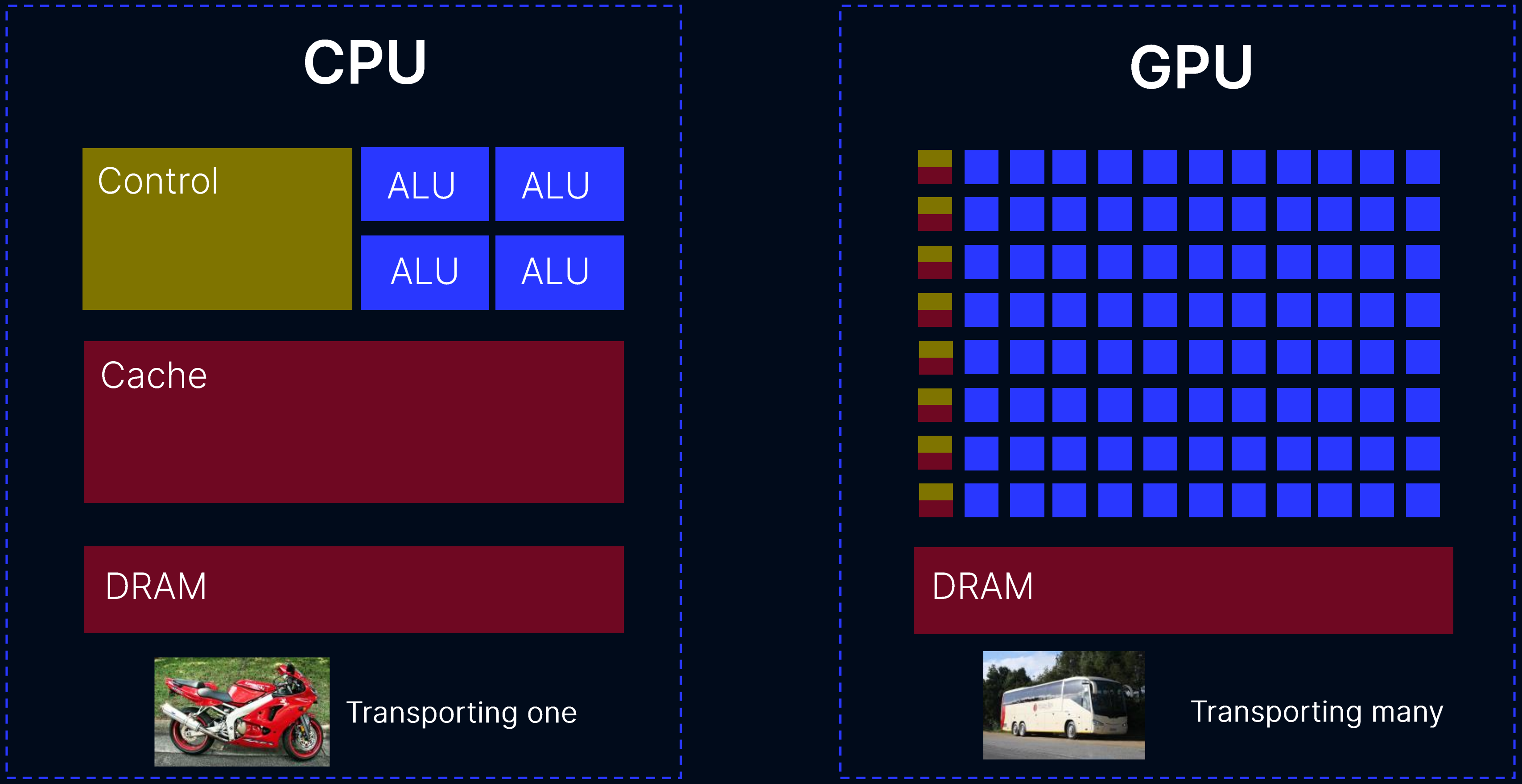
Split physical cores into virtual cores (threads)



Multiple threads with different tasks are executed simultaneously on one CPU core

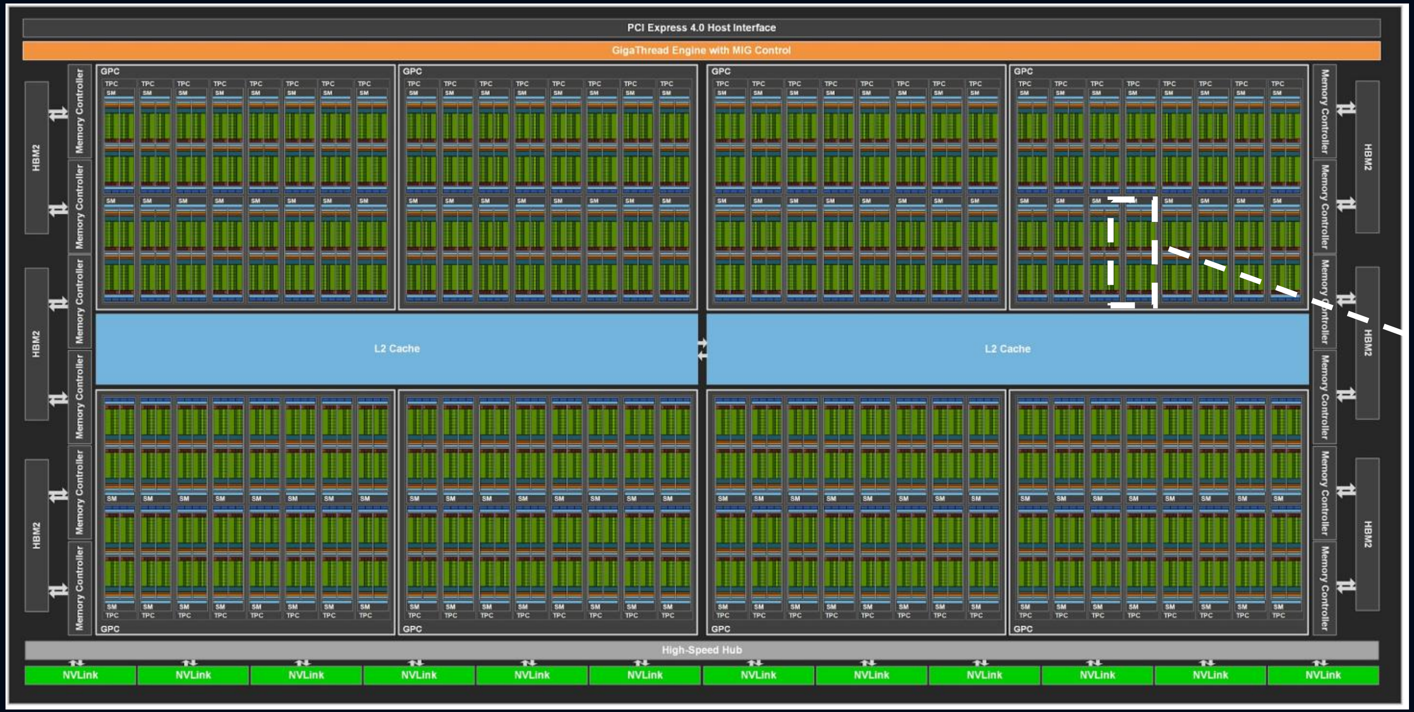
GRAPHICS PROCESSING UNIT (GPU) VS CPU

Made of many simple Cores



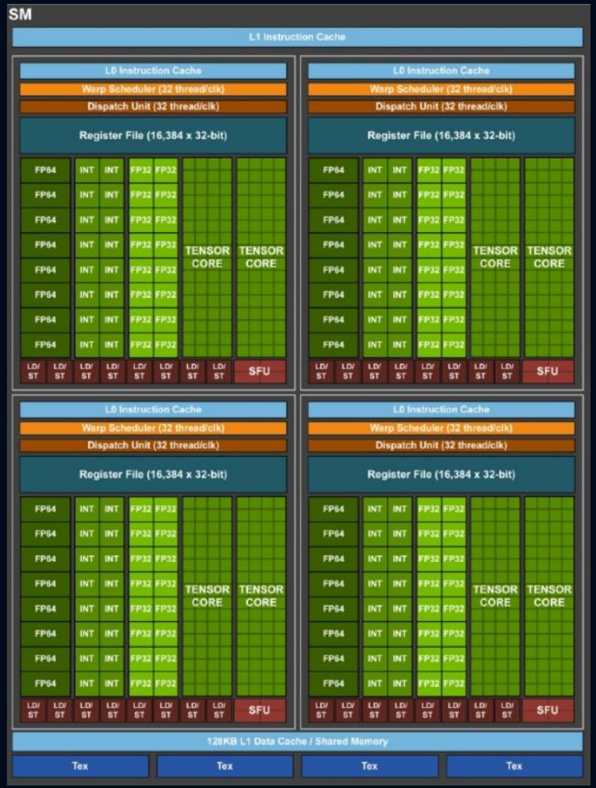
NVIDIA AMPERE GPU ARCHITECTURE

Streaming Multiprocessors (SMs)



NVIDIA Ampere Architecture
(128 SMs)

SM ≅ CPU core

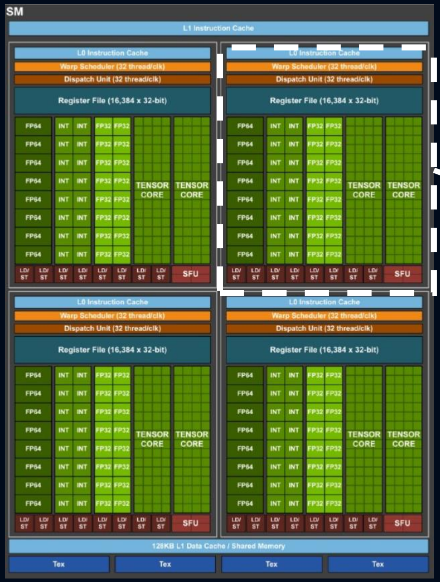


Ronny Krashinsky, et al., NVIDIA Ampere Architecture In-Depth
https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

NVIDIA AMPERE GPU ARCHITECTURE

Streaming Processors (SPs)

SM



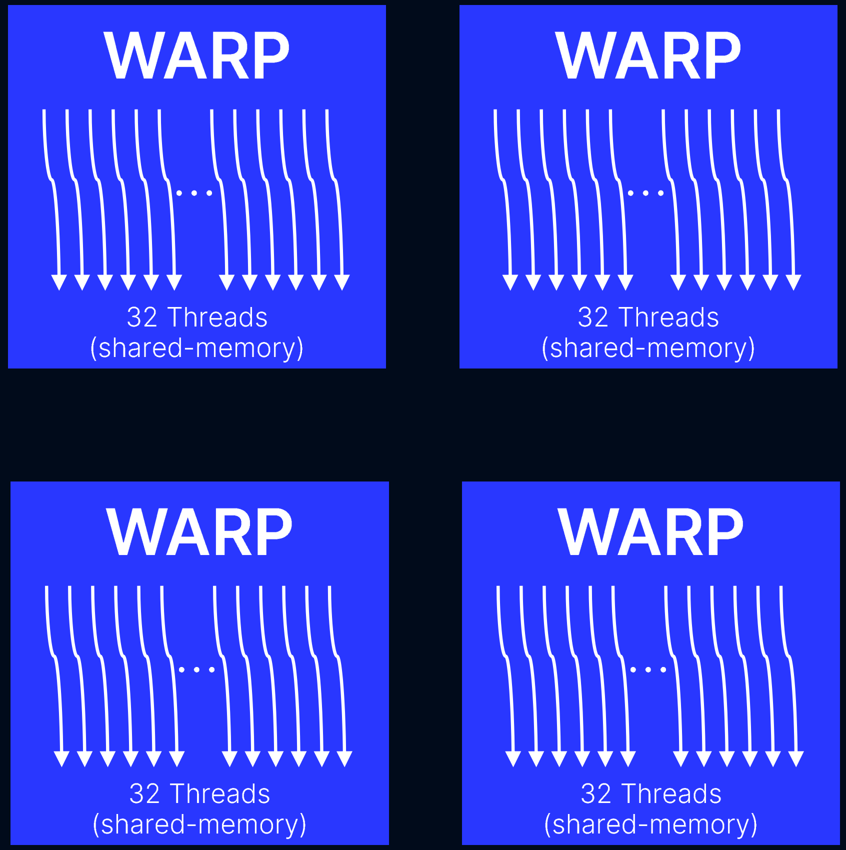
SP (CUDA core)

Compute elements:
16 32-bit integer point units
16 32-bit floating point units
8 64-bit floating point units

Ronny Krashinsky, et al., NVIDIA Ampere Architecture In-Depth
https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

SINGLE INSTRUCTION MULTIPLE THREADS (SIMT)

In-processor parallelism - SIMT= SIMD + SMT

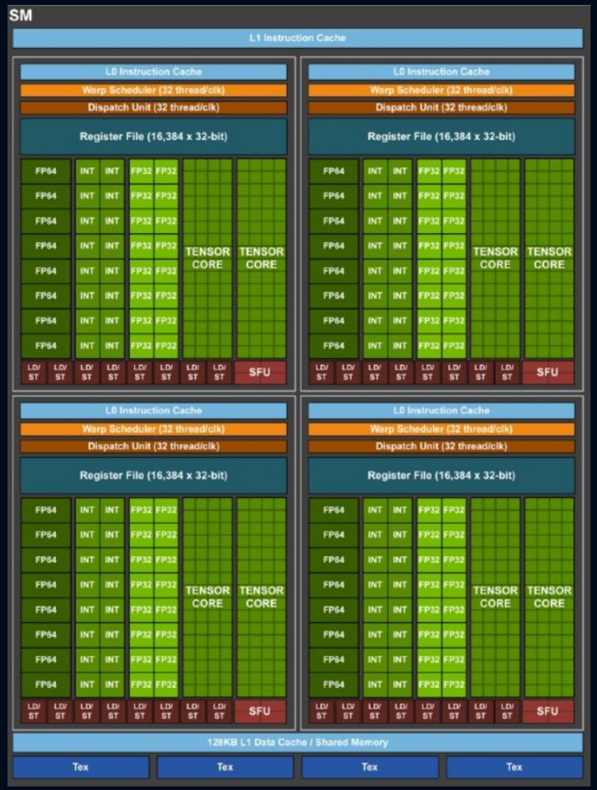


Thread block

The WARPS are executed simultaneously by a SM

The threads of a WARP execute the same instruction (as SIMD)

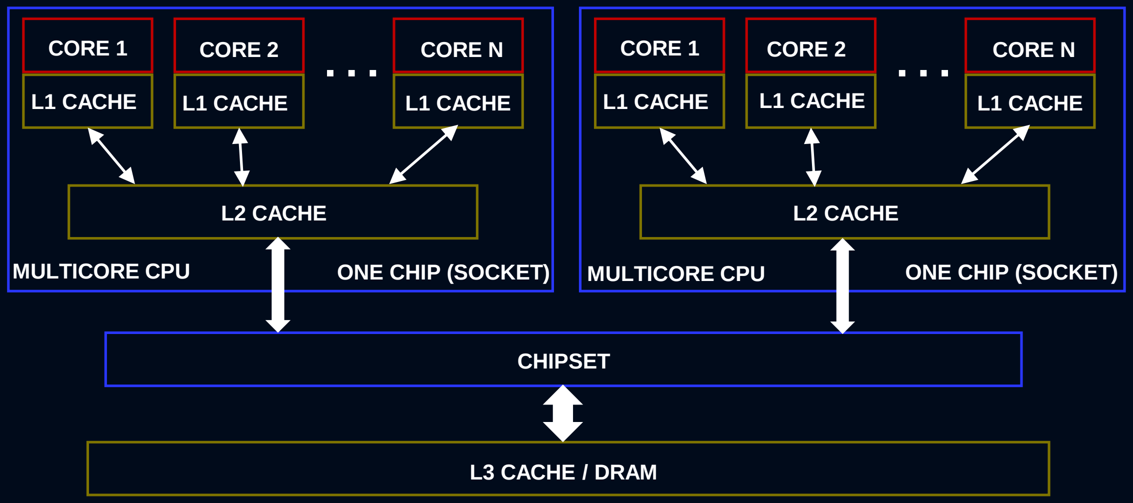
Streaming Multiprocessor (SM)



Ronny Krashinsky, et al., NVIDIA Ampere Architecture In-Depth
https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

SHARED-MEMORY ARCHITECTURE

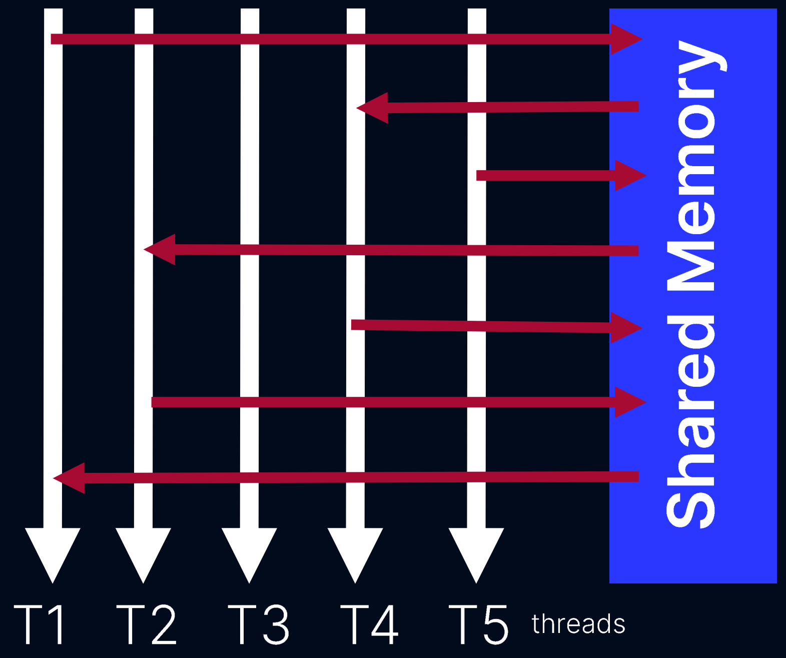
Single computer



PARALLEL PROGRAMMING MODEL: OPENMP

Single Computer

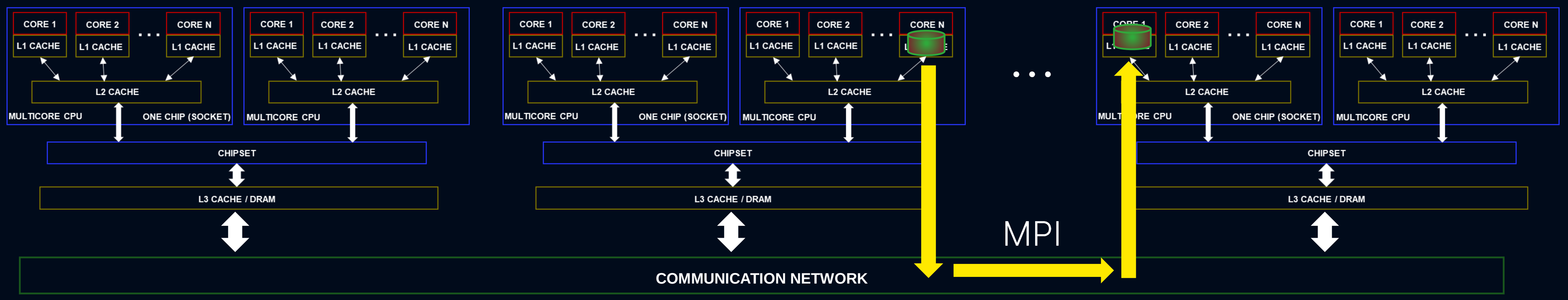
OpenMP



The programmer work on the parallelism, leaving the data shuffling to the hardware system

DISTRIBUTED-MEMORY ARCHITECTURE

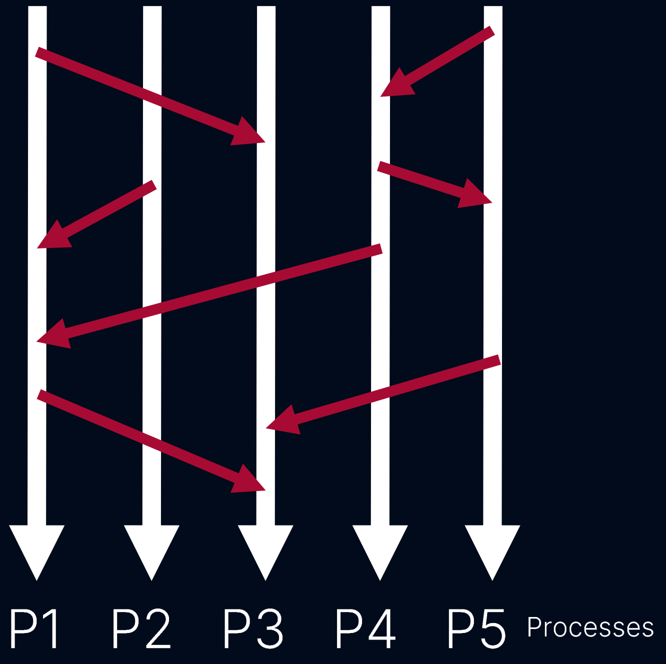
Multiple Computers



PARALLEL PROGRAMMING MODEL: MPI

Multiple Computers

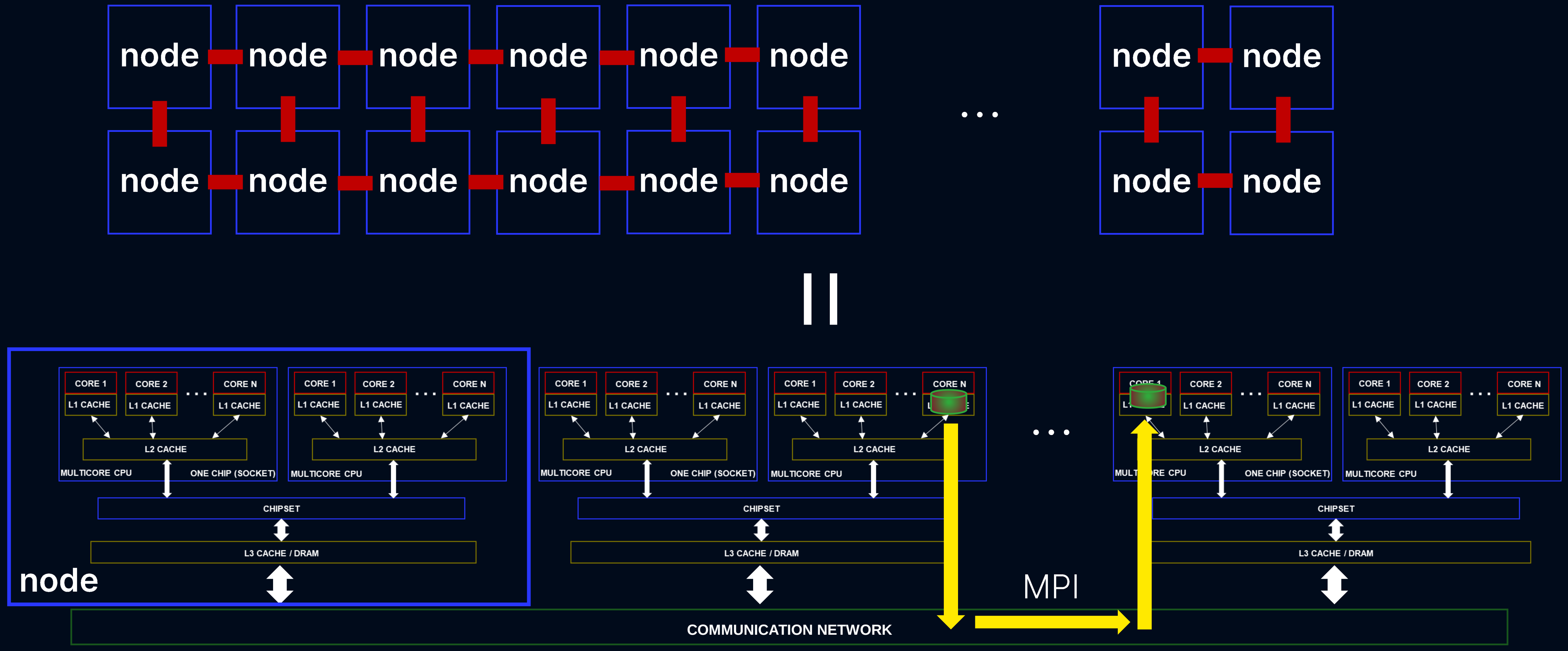
Message Passing Interface (MPI)



Forces the programmer to consider the distribution of the data as a first-class concern

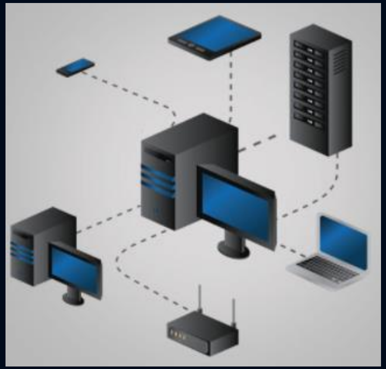
WHAT IS A SUPERCOMPUTER?

Mixture of shared-memory and distributed-memory architectures



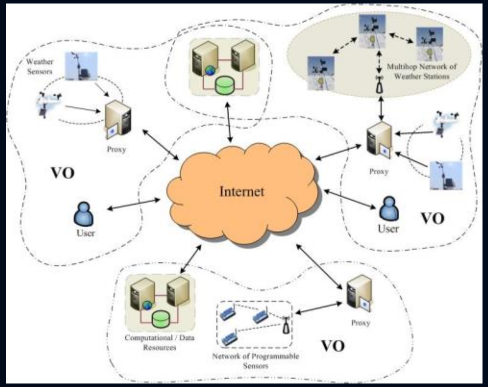
WHAT IS NOT A SUPERCOMPUTER?

Computer cluster



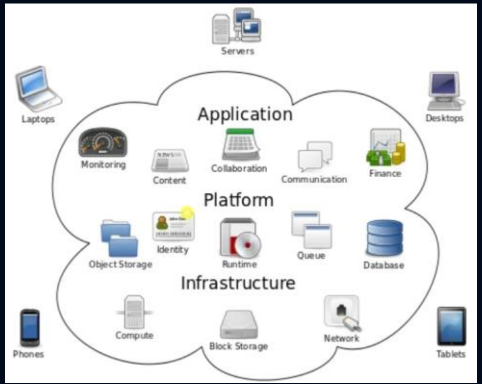
Many computers bound together locally

Grid computing



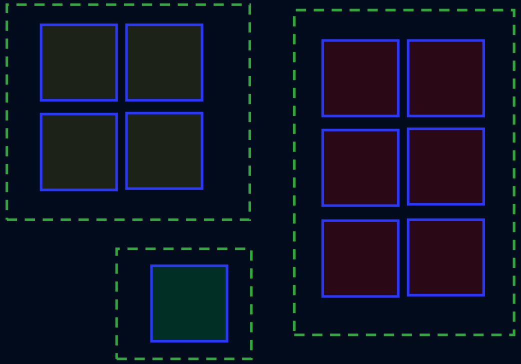
Many clusters working as some kind of supercomputer

Cloud computing



Virtual machines in compute center(s)

High Throughput Computing



Many independent computations

Difference between Grid Computing and Cloud Computing
http://www.differencebetween.net/

TOP500 LIST

November 2020

https://www.top500.org/

Fugaku
442,010 Tflops/s

1

Summit
148,600 Tflop/s

2

Sierra
94,640 Tflop/s

3

TOP500

NVIDIA DGX
SuperPOD
26.195 Gflops/watts

1

MN-3
26.039 Gflops/watts

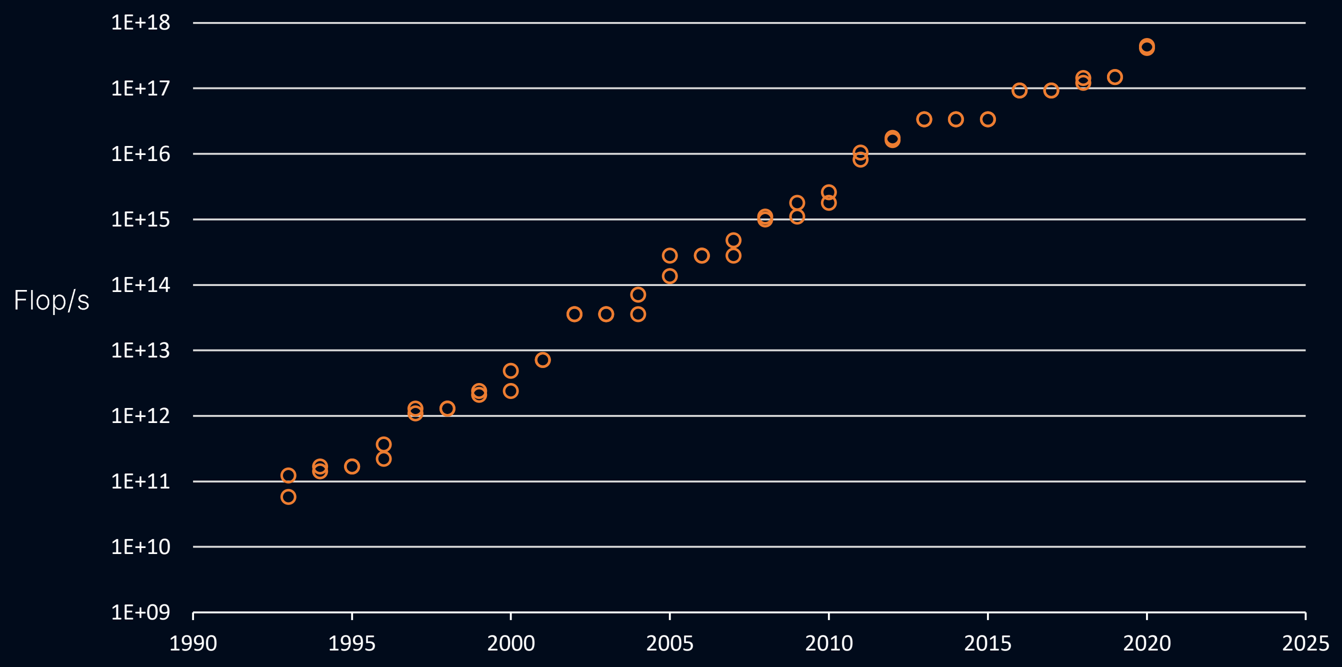
2

JUWELS
Booster Module
25.008 Gflops/watts

3

GREEN500

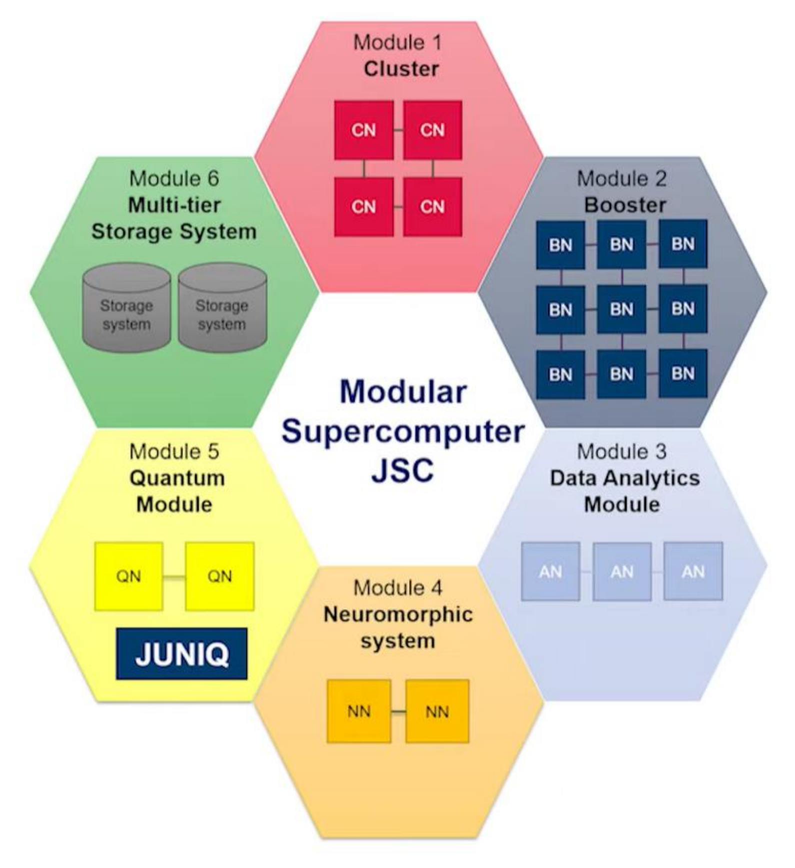
A RACE TOWARD EXASCALE COMPUTING



TOP500, Performance Development
https://www.top500.org/statistics/perfdevel/

MODULAR SUPERCOMPUTING ARCHITECTURE (MSA)

Heterogeneous HPC clusters (modules) within a single system



REVIEW ON HARDWARE LEVELS OF PARALLELISM

Best performance is achieved with a combination of them!

SIMD

In-core
parallelism

SIMT

In-processor
parallelism

Many threads
on many cores

SMT

Single Computer

Simultaneous Multithreading
Cross-core, Cross-socket
OpenMP, pthreads

MPI

Multiple "Computers"

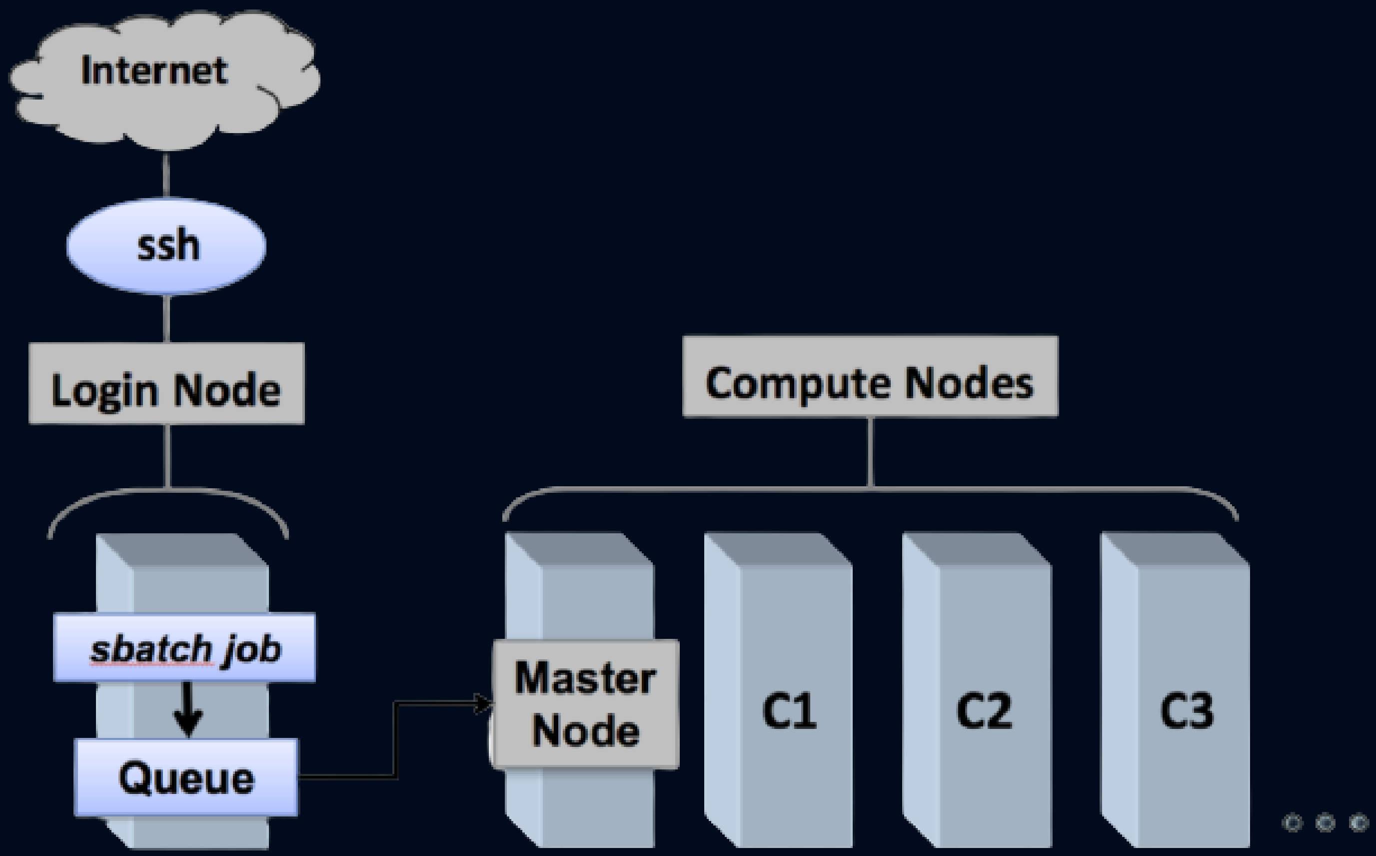
Tightly-coupled
Supercomputing

MPI+MSA

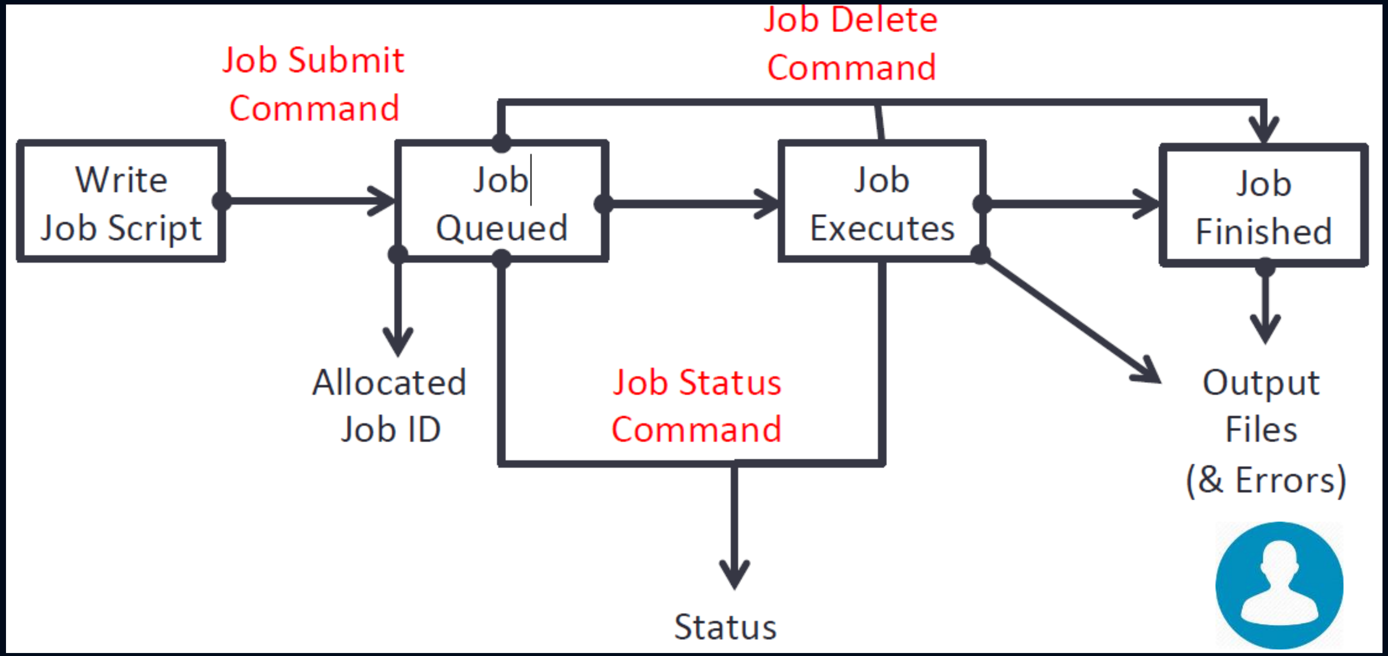
Multiple HPC Systems

Tightly-coupled
Heterogeneous Hardware

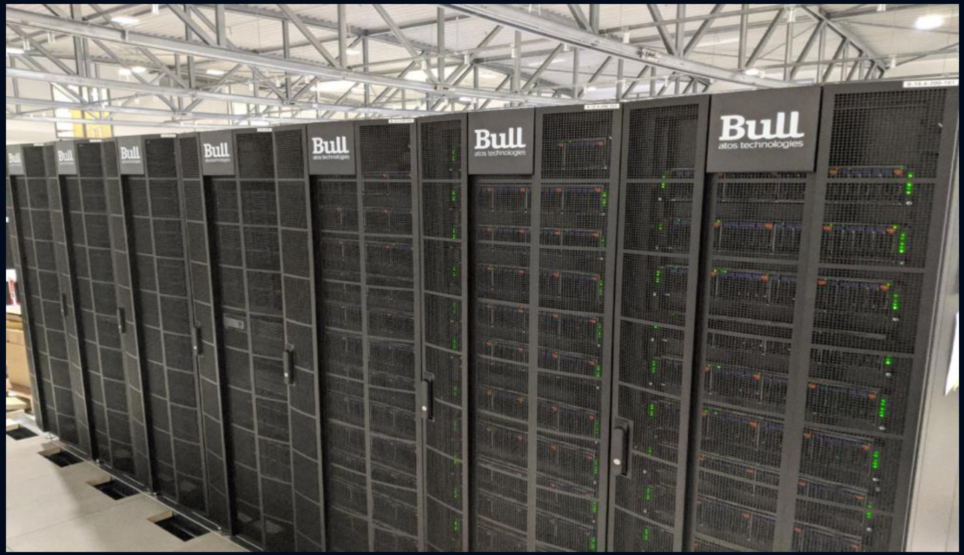
ANATOMY OF A SUPERCOMPUTER



SUPERCOMPUTER USAGE MODEL



JUSUF
Jülich Support for Fenix



https://fenix-ri.eu/

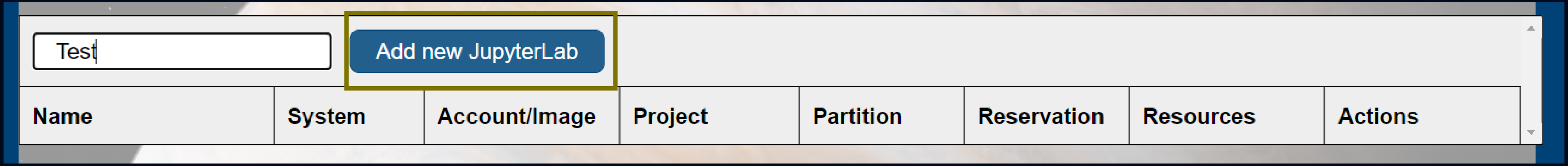
Partition with GPUs

61 NVIDIA V100 GPU with 16 GB Memory

https://apps.fz-juelich.de/jsc/hps/jusuf/cluster/configuration.html

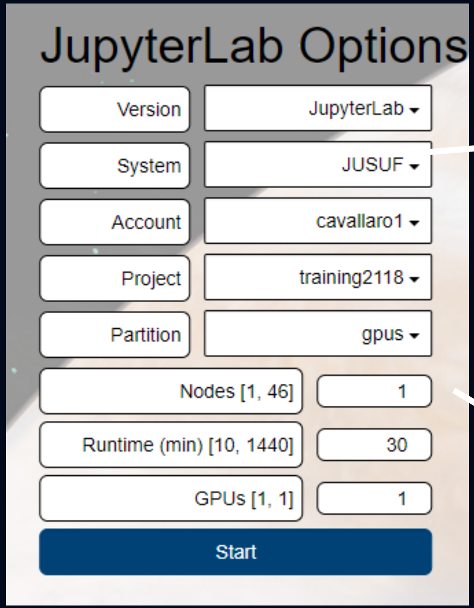
PRE-PRACTICAL – STEP 1

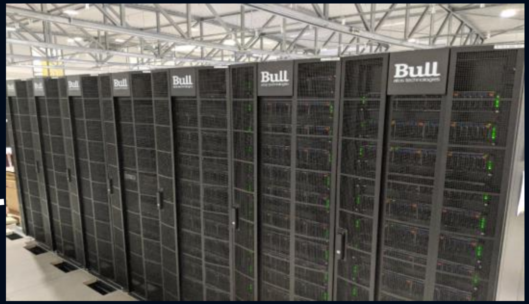
Go to https://jupyter-jsc.fz-juelich.de/, login and add a new JupyterLab

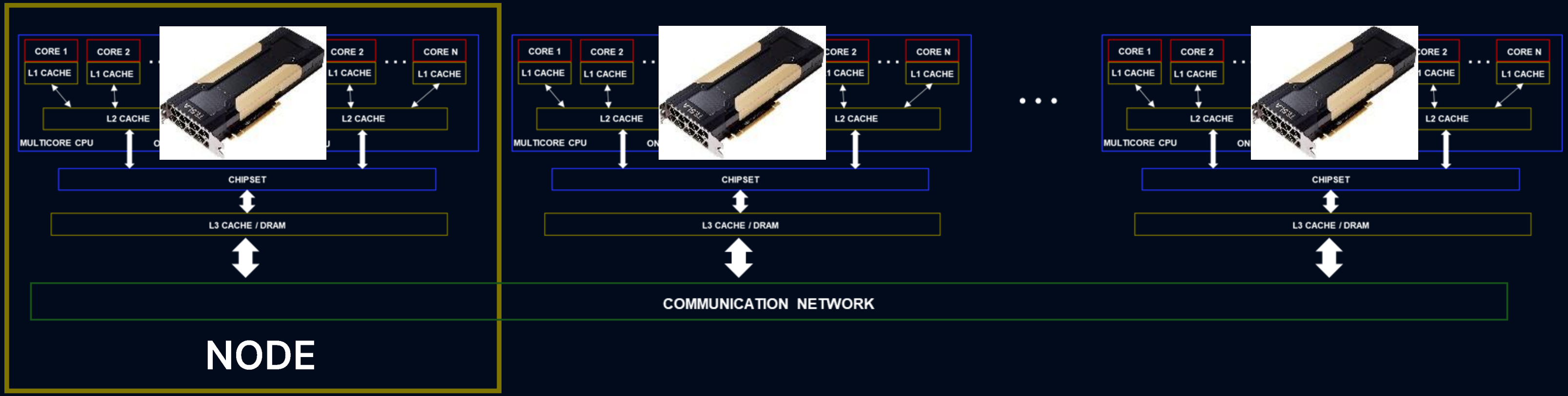


STEP 2

Select the options and Start



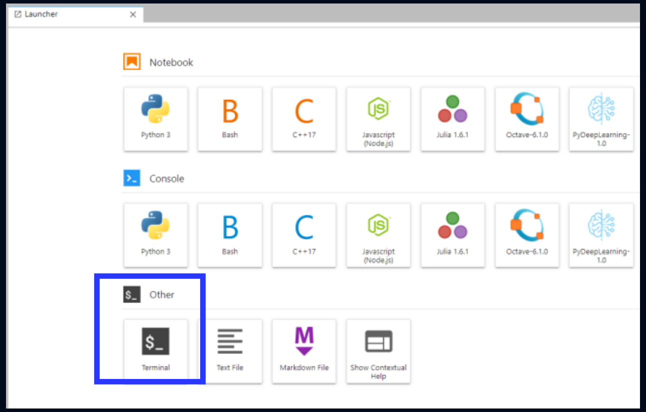




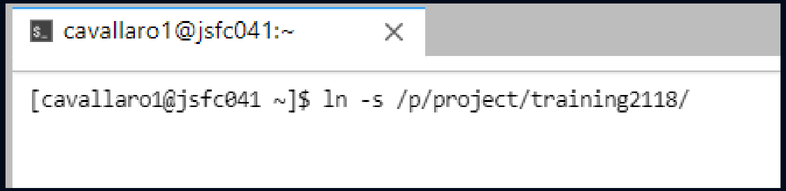
STEP 3

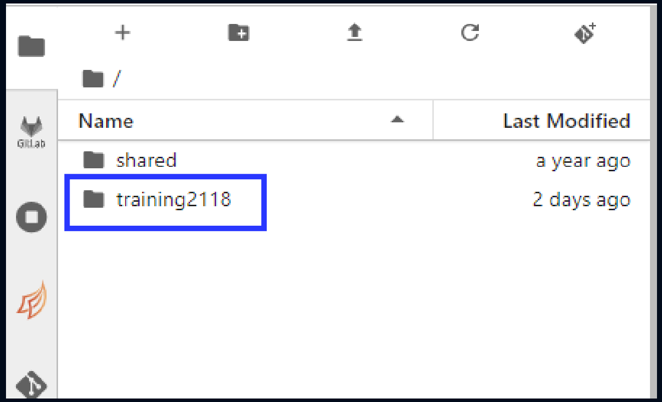
Obtain the tutorial folder

- Open terminal



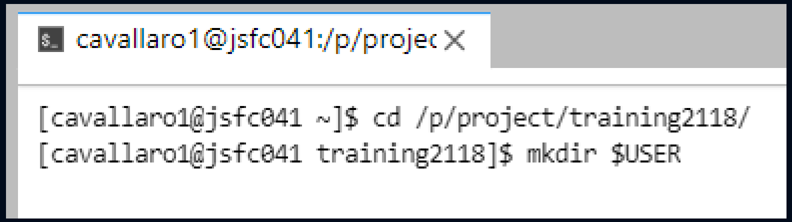
- Run this command and the tutorial folder training2118 will appear in the navigator on the left



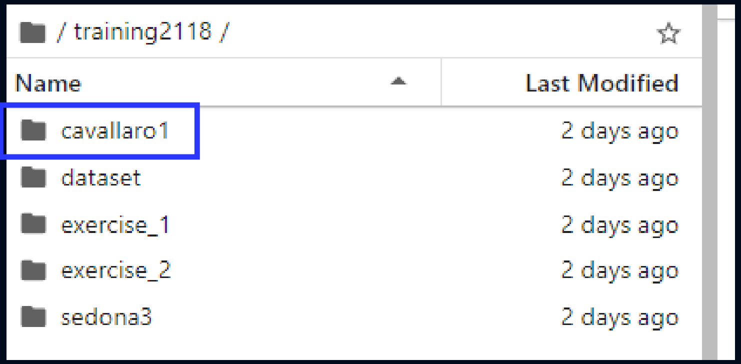


STEP 4
Create your own folder

- Run the following commands in the terminal to navigate to the tutorial folder "training2118" and create your own folder



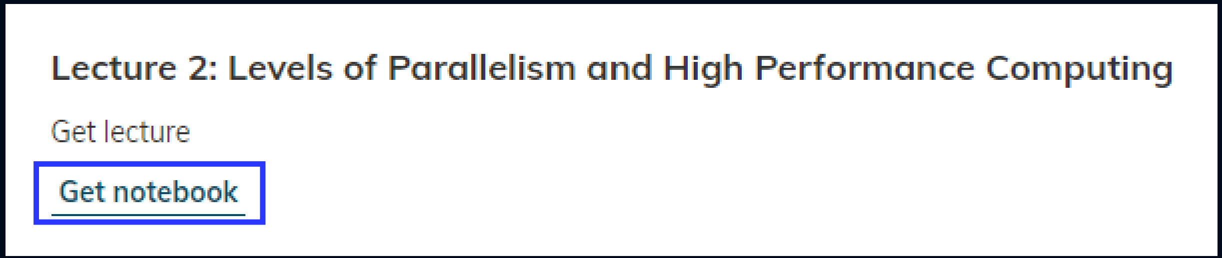
- You can check if the folder was created by looking at the navigator on the left



STEP 5

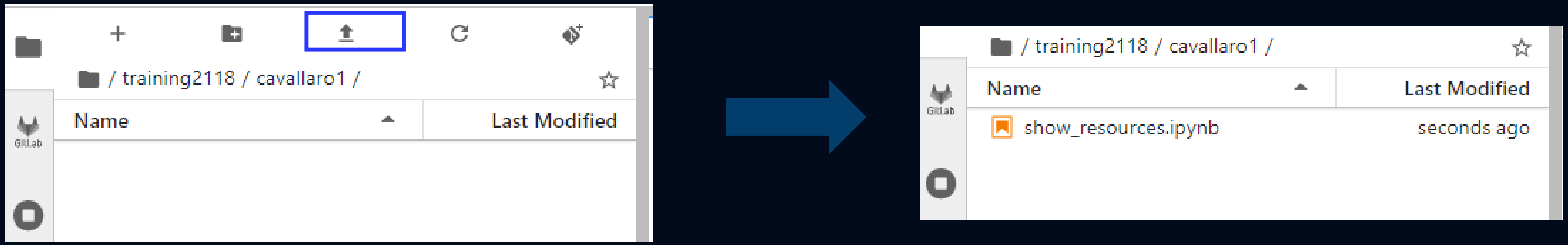
Download Jupyter notebook

- Go to https://www.gabriele-cavallaro.com/teaching/tutorial-igarss2021
- Download the Jupyter notebook ”show_resources.ipynb ”of Lecture 2



STEP 6

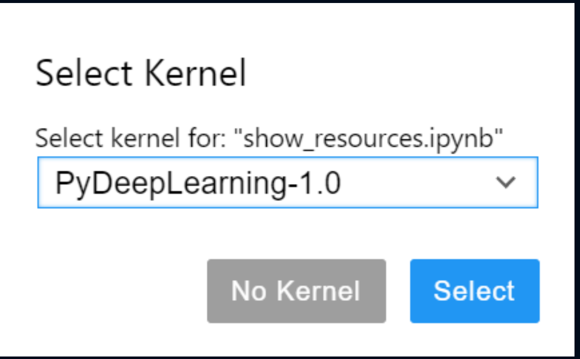
Upload the Jupyter notebook in your own folder

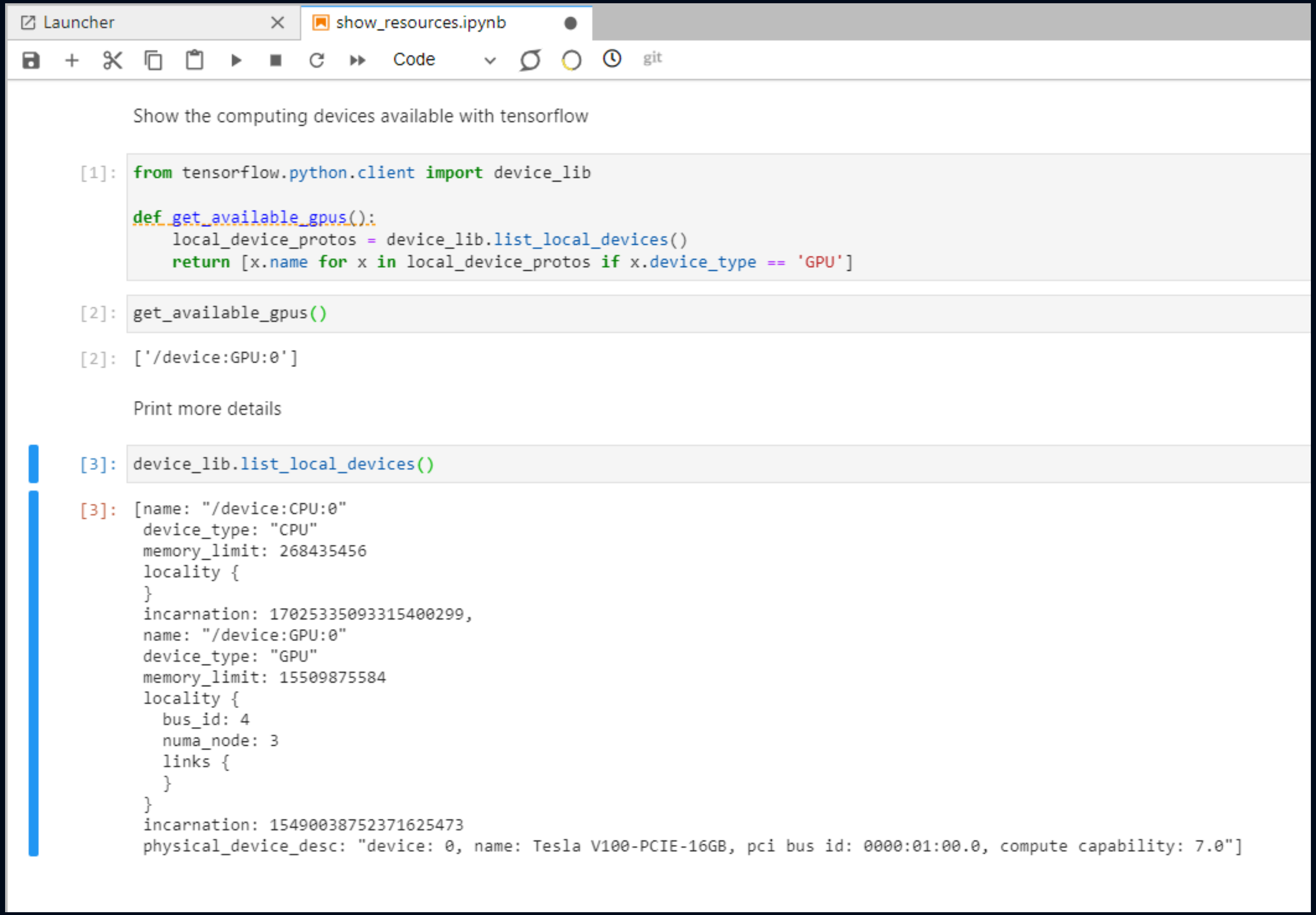


STEP 7

Run the Jupyter notebook

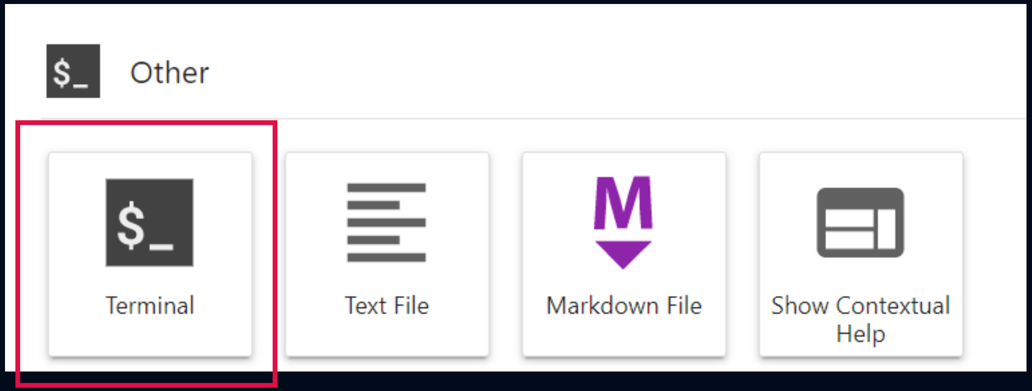
- Select the kernel PyDeepLearning-1.0





STEP 8

TERMINAL / Shell



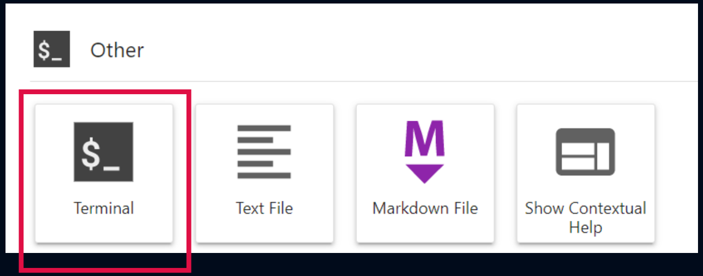
cavallaro1@jsfc041:~

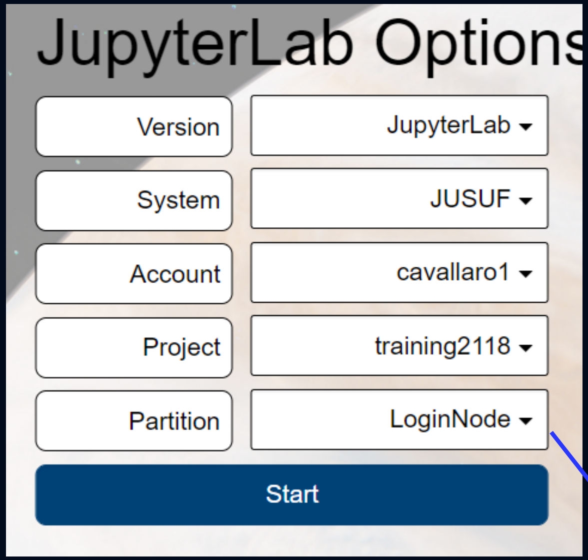
```
[cavallaro1@jsfc041 ~]$ module unload nvidia-driver/.default
[cavallaro1@jsfc041 ~]$ srun nvidia-smi
Fri Jul  9 16:28:37 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  On   | 00000000:01:00.0 Off |                    0 |
| N/A   27C    P0    26W / 250W |      0MiB / 16160MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
[cavallaro1@jsfc041 ~]$
```

STEP 9

Access on the login node

- Starting point, in this node you do not have GPUs

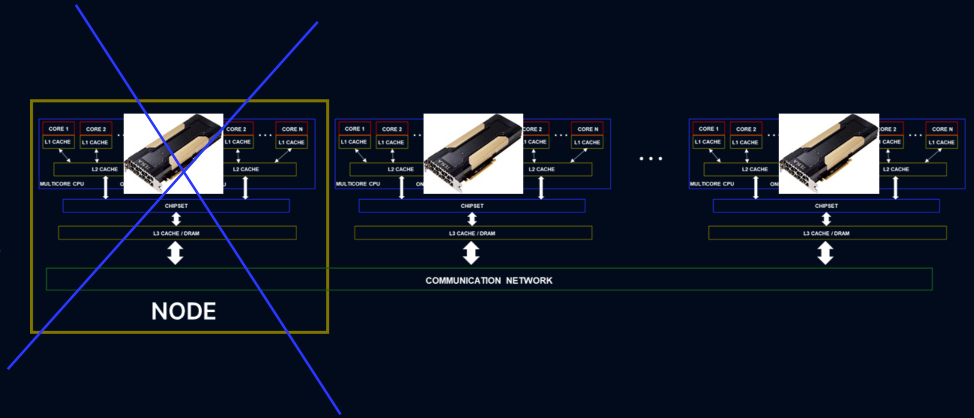




cavallaro1@jsfl03:~

```
[cavallaro1@jsfl03 ~]$ nvidia-smi
NVIDIA-SMI has failed because it couldn't communicate with the NVIDIA driver. Make sure that the latest NVIDIA driver is installed and running.
```

LoginNode



TOMORROW PRACTICALS

Hands-on - Distributed Deep Learning

- Workflow on the batch system
- Use job scripts to execute algorithms with more nodes (i.e., > 1 GPUs)