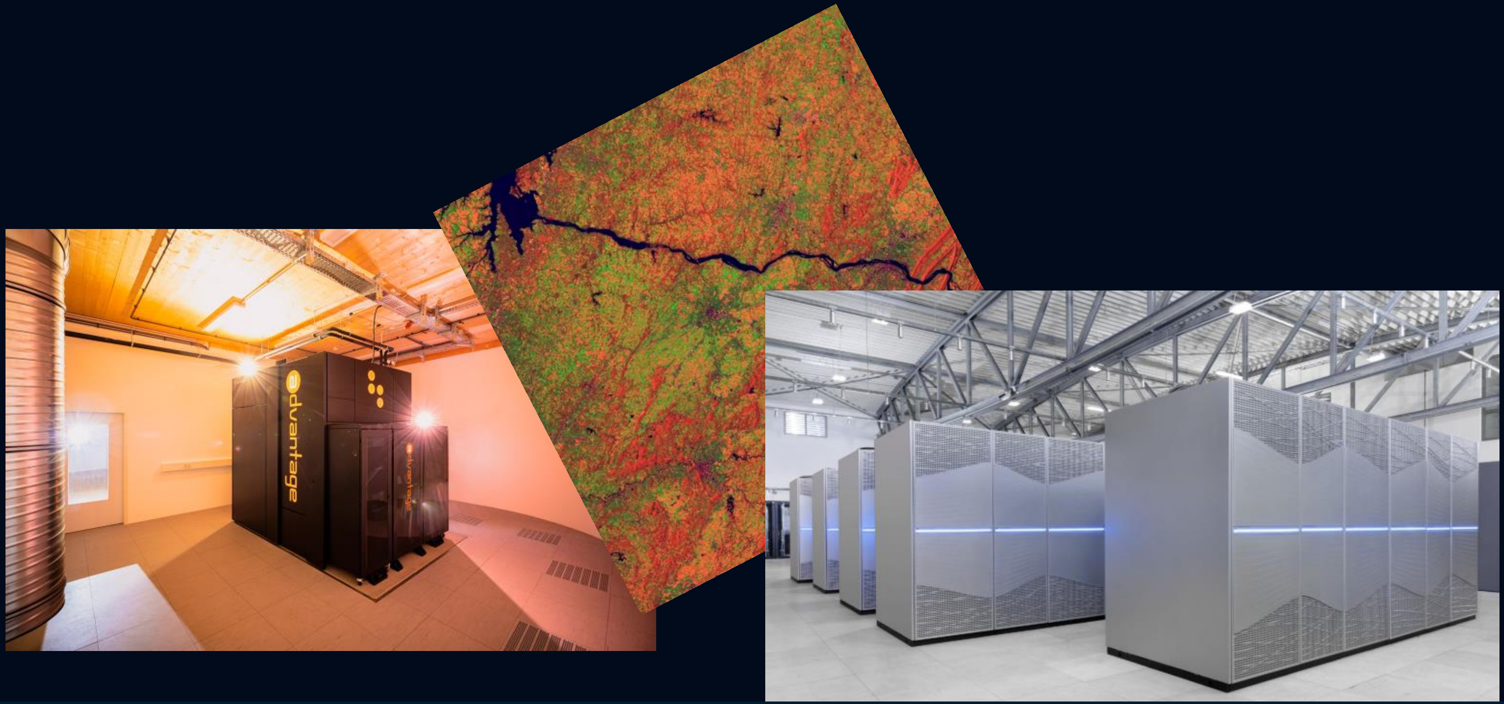


LECTURE 3: LEVELS OF PARALLELISM AND HIGH PERFORMANCE COMPUTING

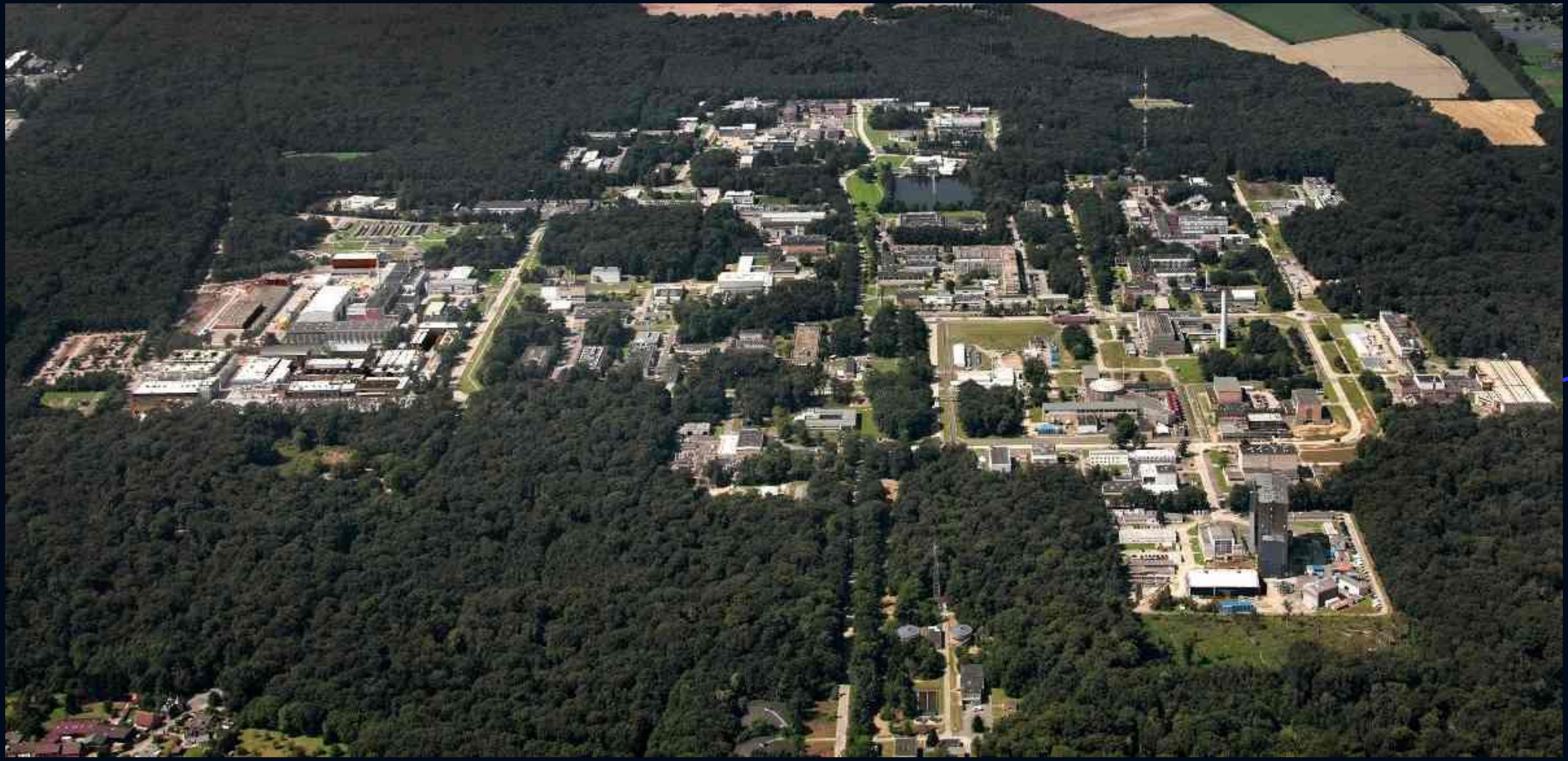
End-to-End Machine Learning with High Performance and Cloud Computing - Tutorial IGARSS 2022 – 17 July, 2022

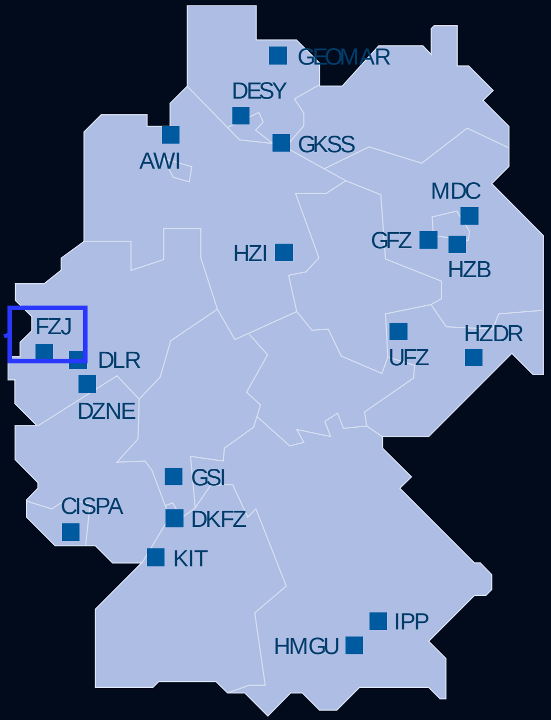
PROF. DR. –ING. GABRIELE CAVALLARO (WWW.GABRIELE-CAVALLARO.COM)
HEAD OF SIMULATION AND DATA LAB "AI AND ML FOR REMOTE SENSING", JÜLICH SUPERCOMPUTING CENTRE
ADJUNCT ASSOCIATE PROFESSOR, SCHOOL OF ENGINEERING AND NATURAL SCIENCES, UNIVERSITY OF ICELAND

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE
ESI
HDCRS
NASA
IMPACT Interagency Implementation and Advanced Concepts Team
RAISE Center of Excellence
UNIVERSITY OF ICELAND

FORSCHUNGSZENTRUM JÜLICH

Helmholtz Association (Germany)

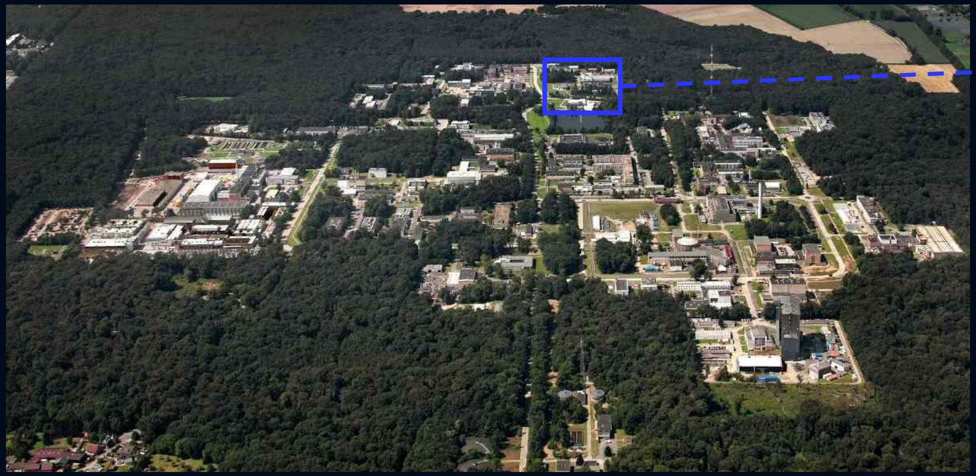




https://www.fz-juelich.de/portal/EN/Home/home_node.html

JÜLICH SUPERCOMPUTING CENTRE

Tier-0 Supercomputing Institution



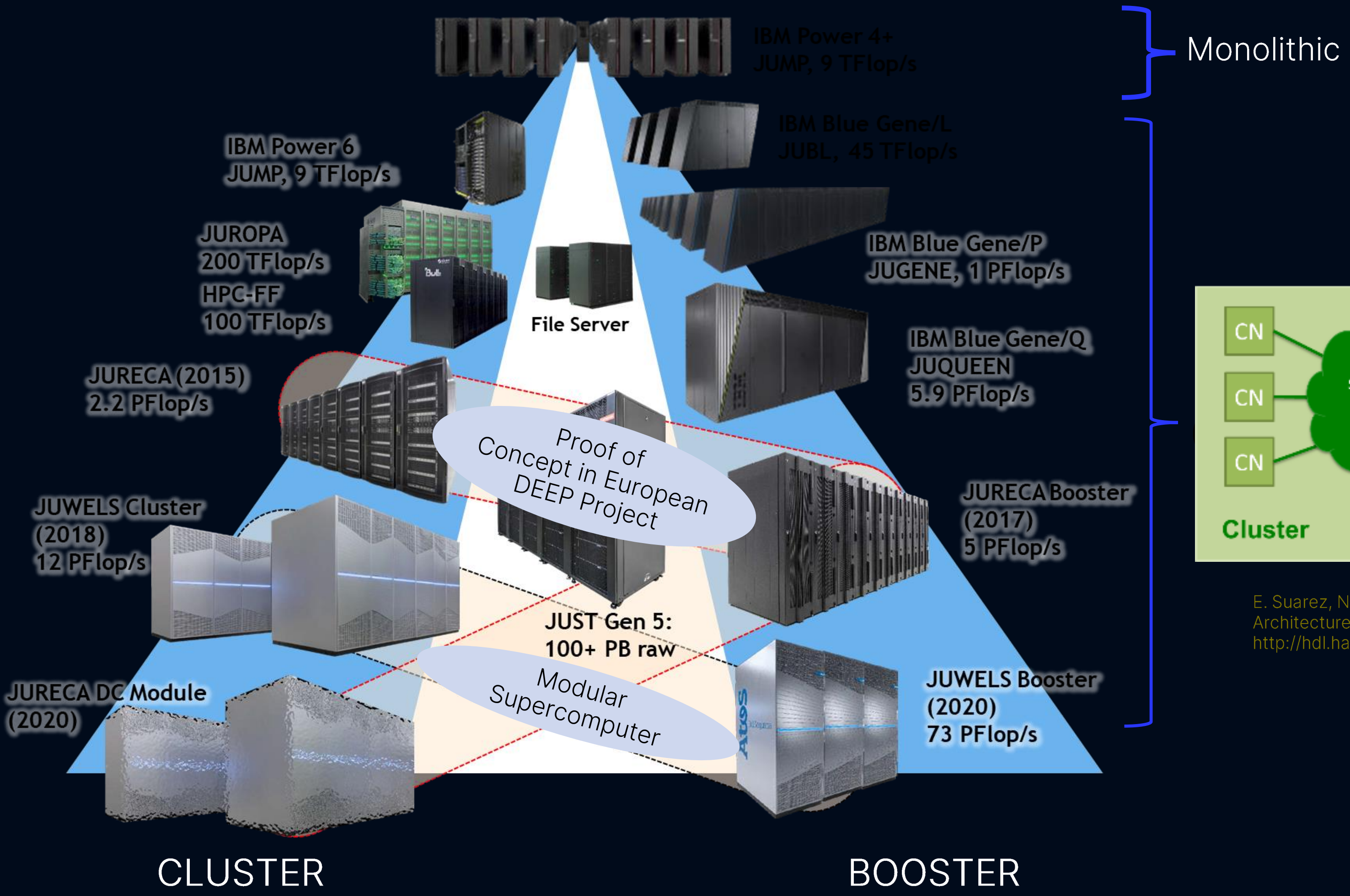
https://www.fz-juelich.de/ias/jsc/EN/Home/home_node.html

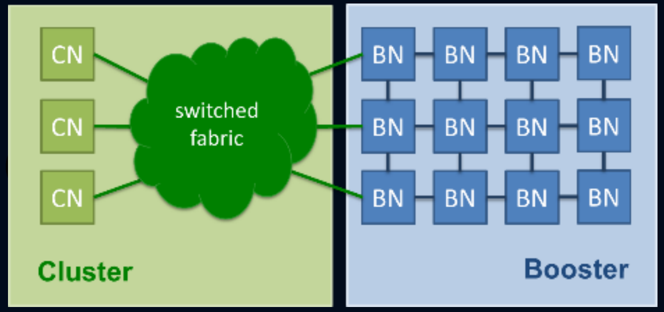
GCS
Gauss Centre for Supercomputing

https://www.gauss-centre.eu/

SUPERCOMPUTERS

https://www.top500.org/





E. Suarez, N. Eicker, and T. Lippert, “Modular Supercomputing Architecture: from Idea to Production”, 2019
http://hdl.handle.net/2128/22212

https://www.deep-projects.eu/

JUWELS

Exascale Pathfinder

- Ranking in November 2020
  - TOP500 (7 World, 1 Europe)
  - Green500 (1 in TOP100)
  - TOP10 AI (4)

ParTec CLUSTER COMPETENCE CENTER Mellanox TECHNOLOGIES Atos Bull NVIDIA

Julich Supercomputing Centre, "JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Julich Supercomputing Centre," Journal of large-scale research facilities, vol. 7, no. A138, 2021.

Designed for simulation and large-scale machine learning

Funded through SiVeGCS (BMBF, MWK-NRW)

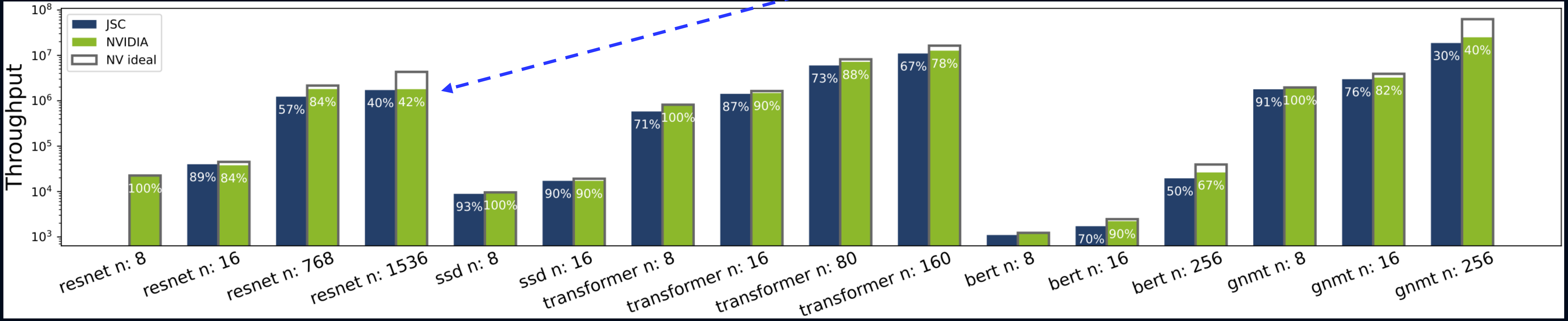
JUWELS BOOSTER – A “DUAL USE” SYSTEM

Benchmark result

- Benchmark: NVIDIA’s submission to MLPerf training v0.7
- Metrics: Throughput in Samples/sec
- 5 Benchmarks on up to 1536 GPUs
- Reference: NVIDIA’s results on specific AI system Selene

Example

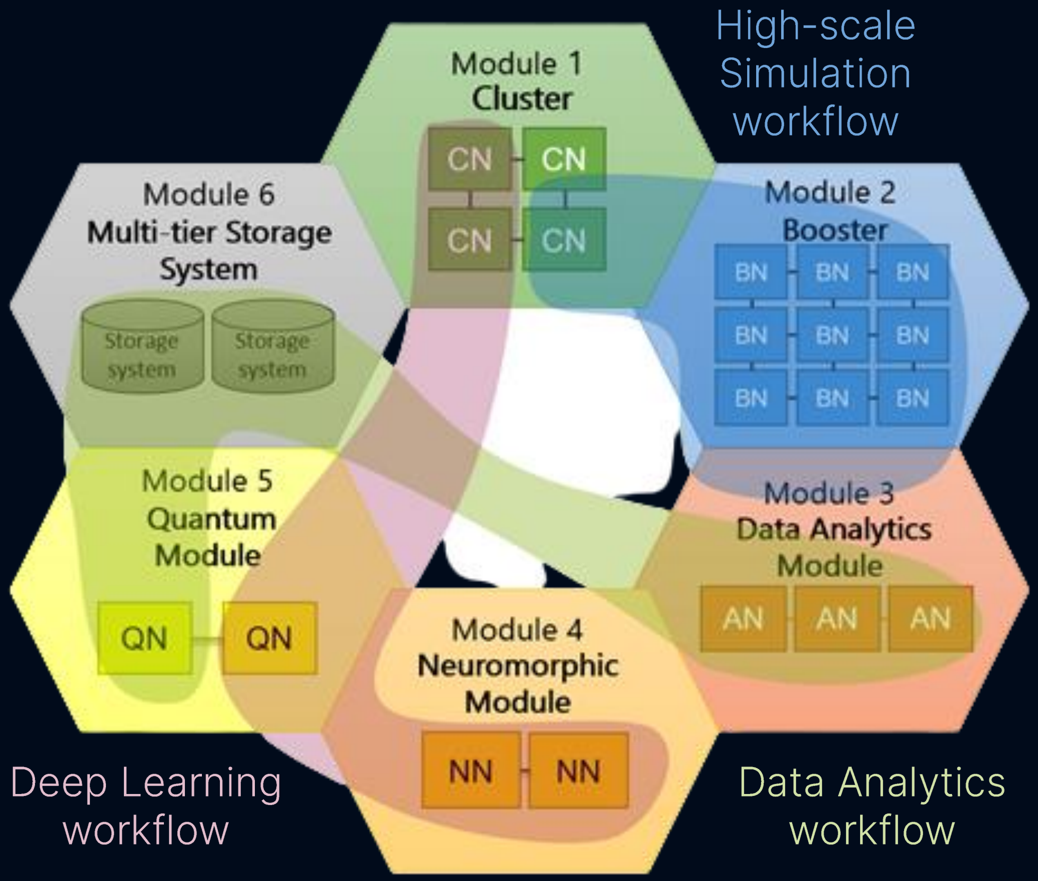
- Task: Train Resnet50 on ImageNet
- GPUs: 1536
- Throughput: 1.7 Million Images / Sec
- Training complete after 43 seconds!
- Parallelization efficiency: 40%



S. Kesselheim, A. Herten, K. Krajsek, J. Ebert, J. Jitsev, M. Cherti, M. Langguth, B. Gong, S. Stadtler, A. Mozaffari, G. Cavallaro, R. Sedona, A. Schug, A. Strube, R. Kamath, M. G. Schultz, M. Riedel, and T. Lippert, “JUWELS Booster – A Supercomputer for Large-Scale AI Research,” in High Performance Computing (H. Jagode, H. Anzt, H. Ltaief, and P. Luszczek, eds.), (Cham), pp. 453–468, Springer International Publishing, 2021.

MODULAR SUPERCOMPUTING

- Cost-effective scaling
- Effective resource-sharing
- Fit application diversity
  - Large-scale simulations
  - Data analytics
  - Machine- and Deep Learning
- Composability of heterogeneous resources



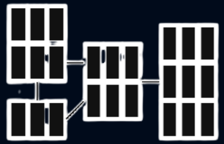
Modular Supercomputing Architecture (MSA)

MISSION OF JSC

Develop supercomputing towards Exascale

Introduce future computing technologies

- Quantum Computing, Neuromorphic Computing



Create exemplary federated infrastructures for data analytics and AI

Provide innovative support structures and tools

Educate a new generation of simulation and data science specialists

Thomas Lippert "Exascale Ante Portas: Machines for Mankind",
Luxembourg Supercomputing Day 2022, https://youtu.be/nw4rVr4yChg

IT ENVIRONMENTS

Key Priorities

Embedded system

Power consumption and price

Desktop Computing

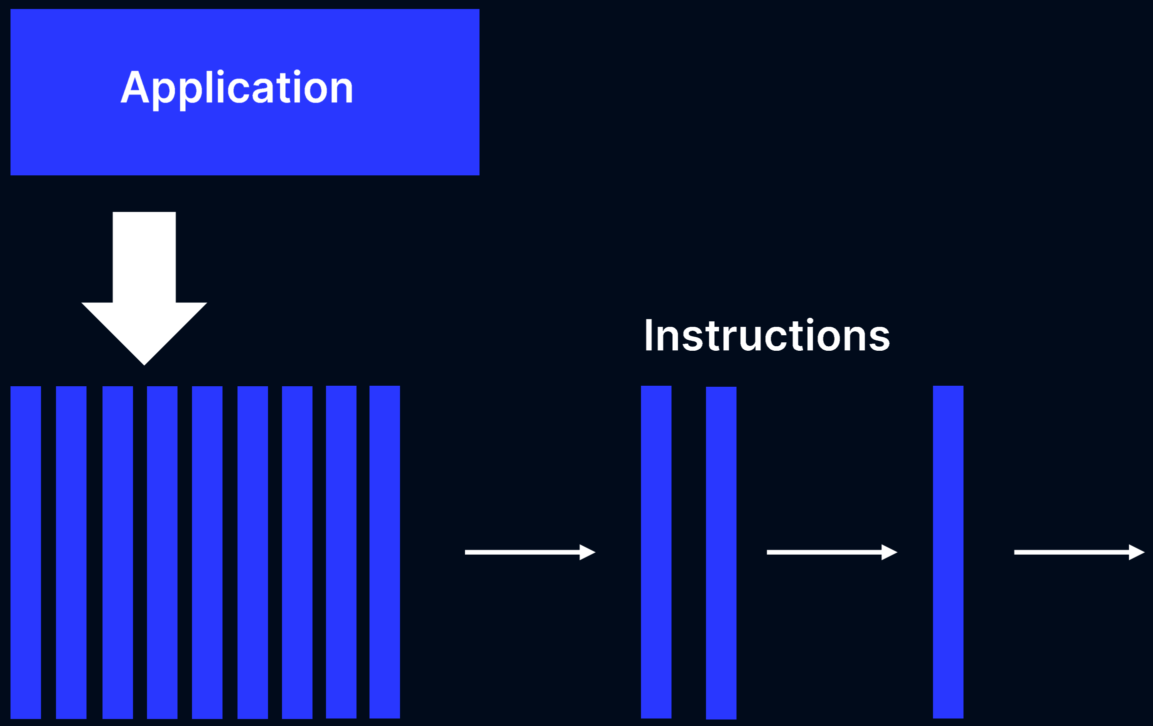
Performance / price ration

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

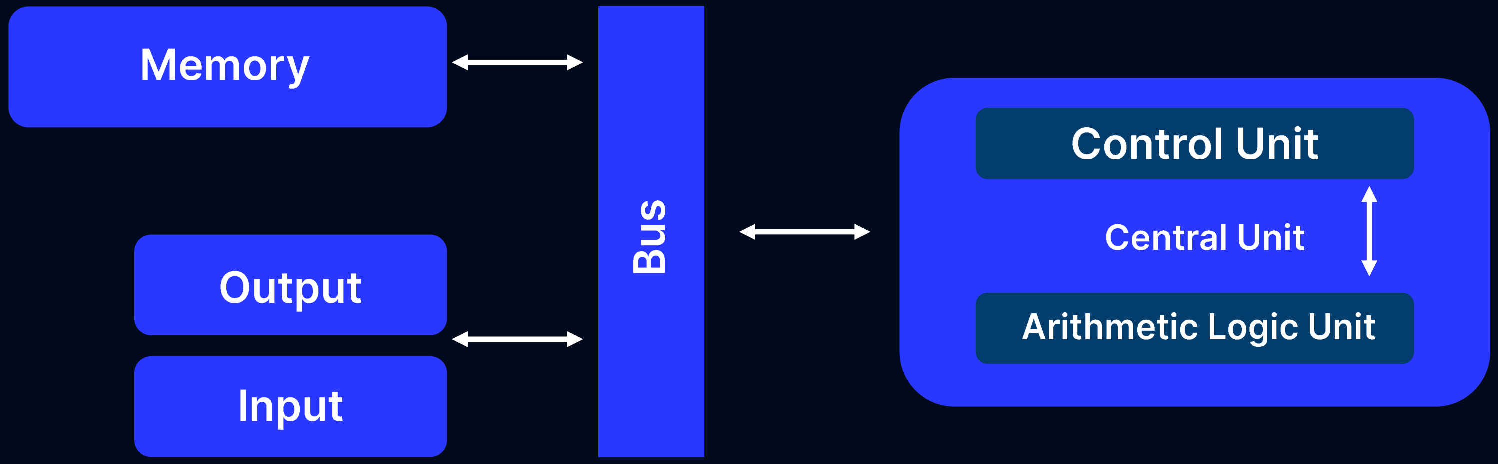
Optimization, Caching

Get Help

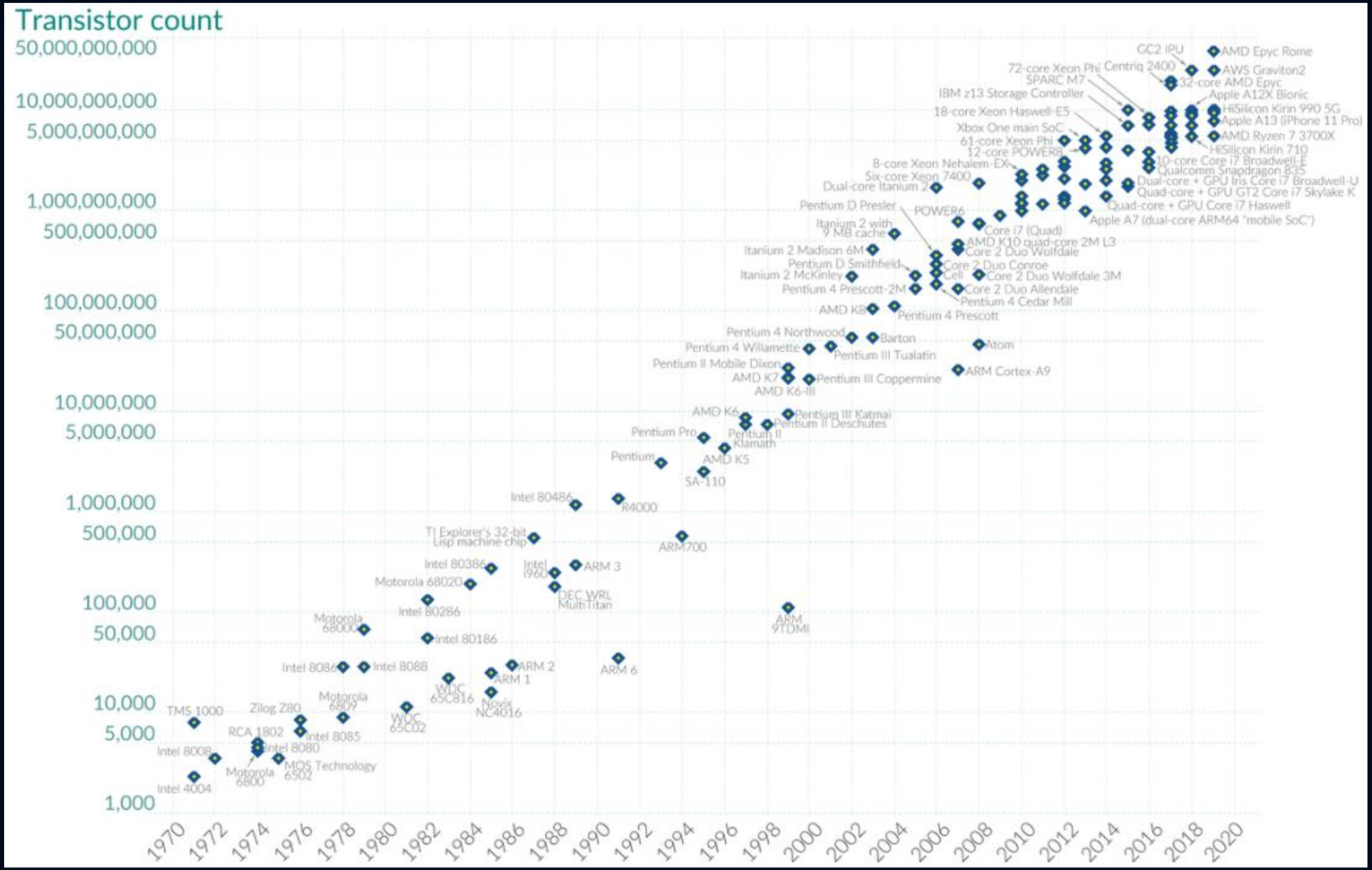
Parallelization

VON NEUMANN ARCHITECTURE

Machine Model



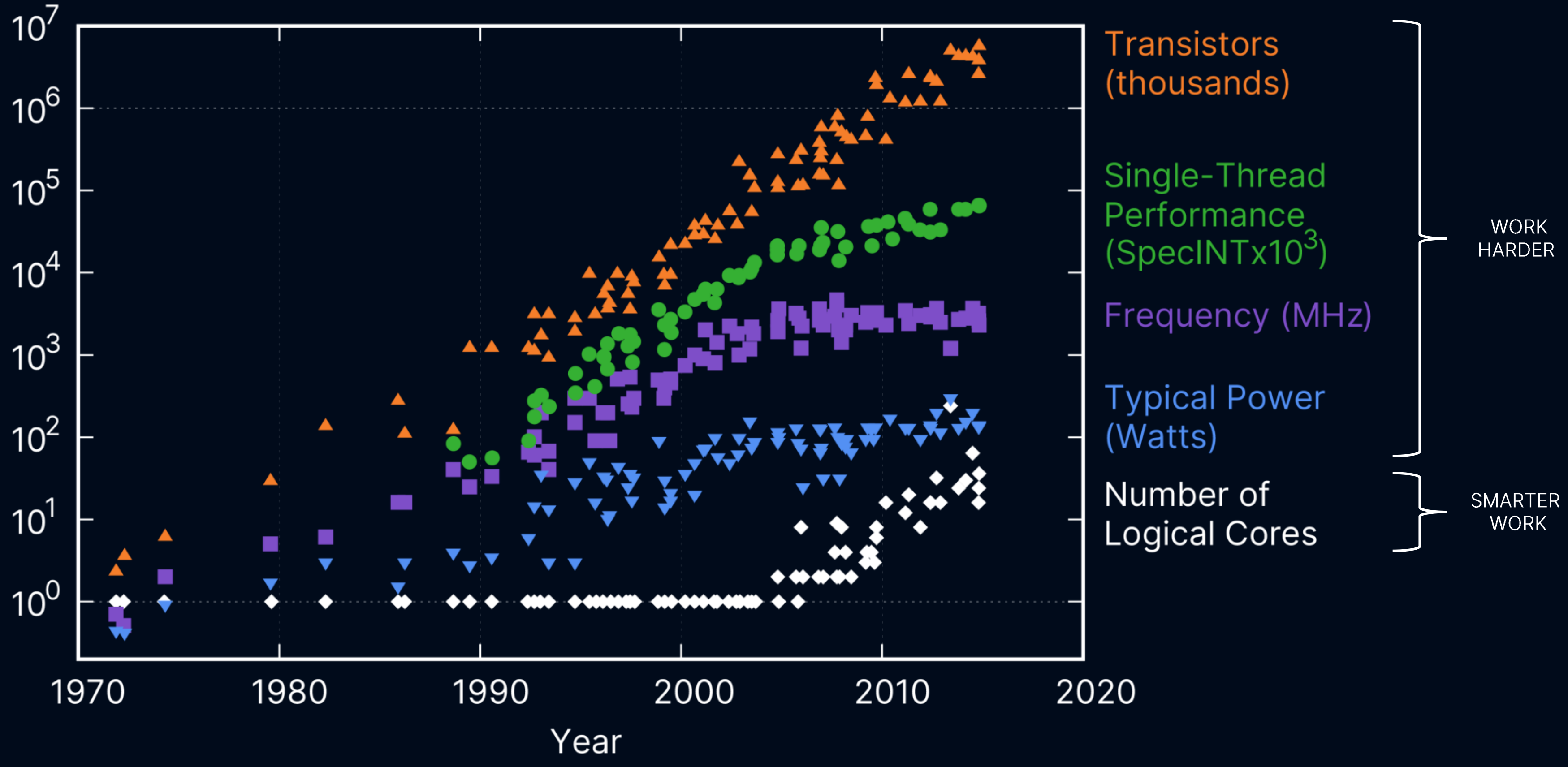
MOORE'S LAW



Wikipedia: Transistor count

THE END OF DENNARD SCALING

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

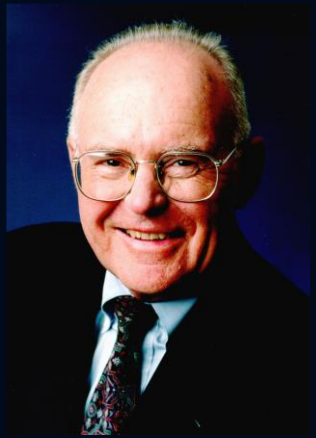


Karl Rupp, 40 Years of Microprocessor Trend Data
https://www.karlrupp.net/2015/06/40-years-of-microprocessor-trend-data/

WILL MOORE'S LAW END?

Paradigm Shift

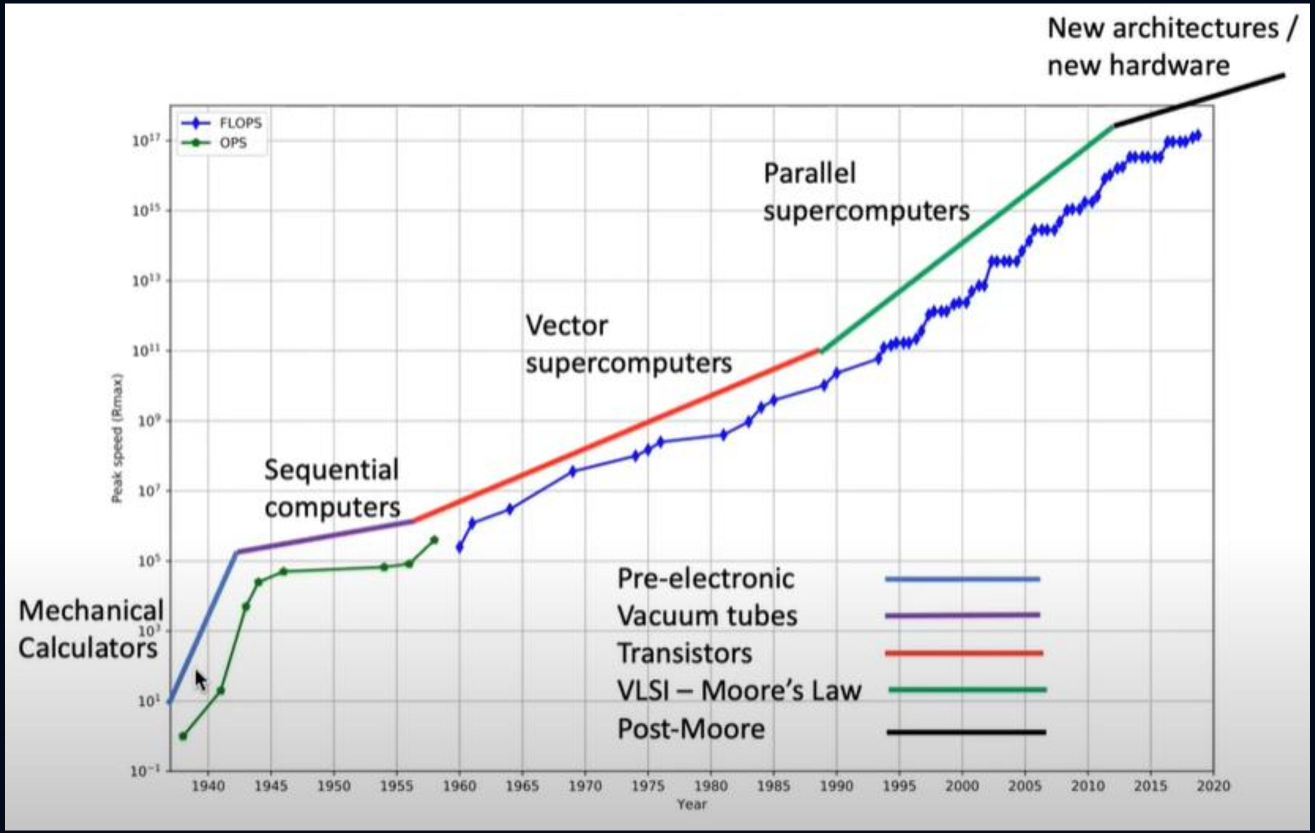
"There's no getting around the fact that we build these things out of atoms"



Gordon Moore

WHY SUPERCOMPUTER PERFORMANCE KEEP INCREASING?

Parallel computing is going mainstream

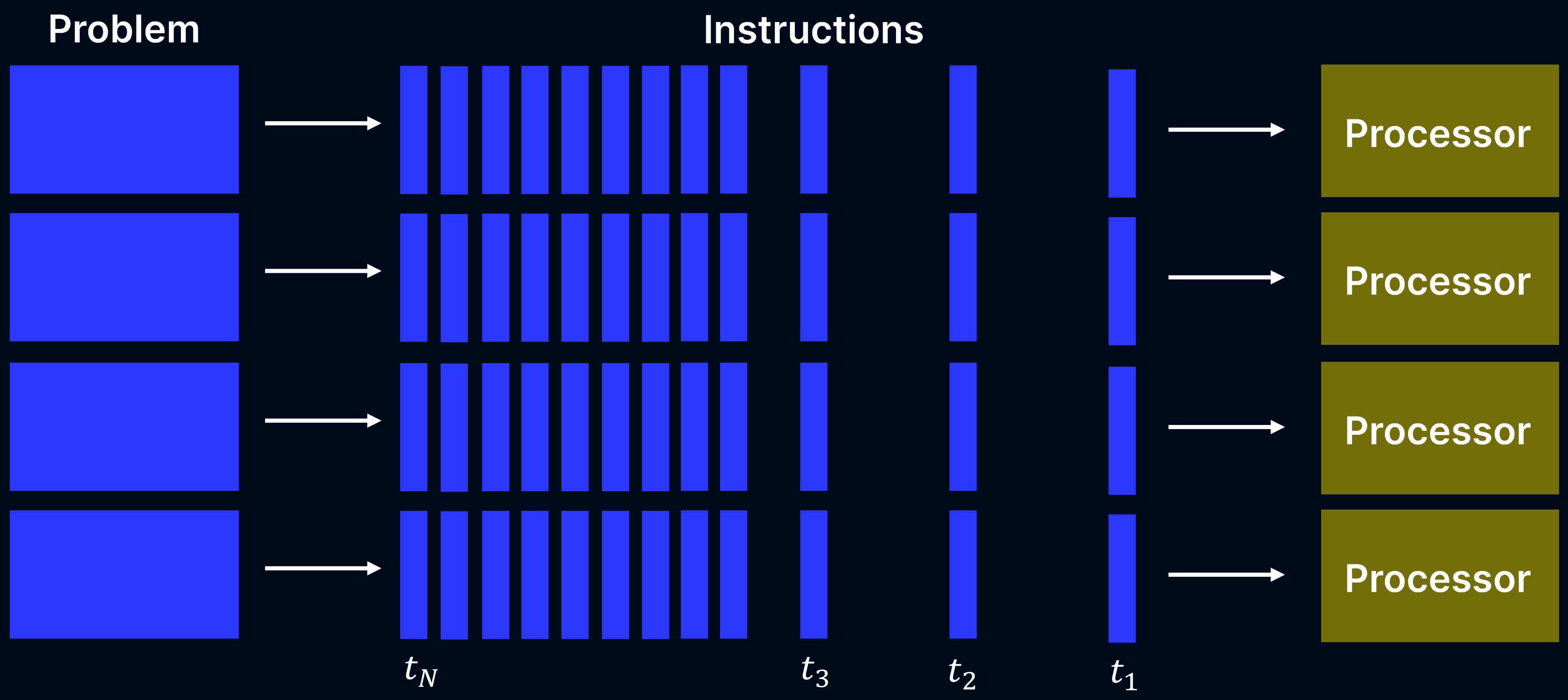


Rob Schreiber, High Performance Computing: Beyond Moore's Law
https://www.youtube.com/watch?v=LOf57fdxln4&t=258s

THE EASY TIMES HAVE GONE

Responsibility for better performance is on the software developers

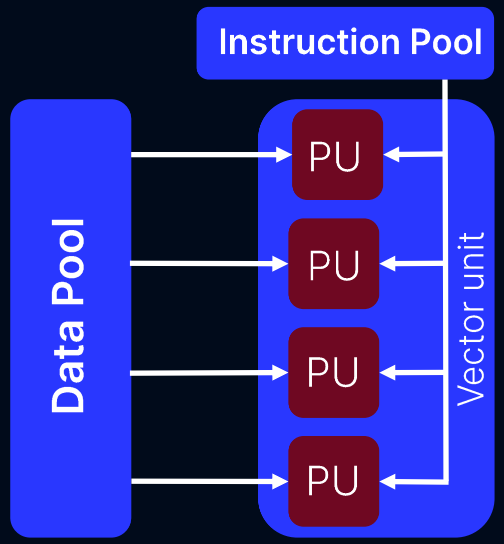
PARALLEL COMPUTING



SINGLE INSTRUCTION MULTIPLE DATA (SIMD)

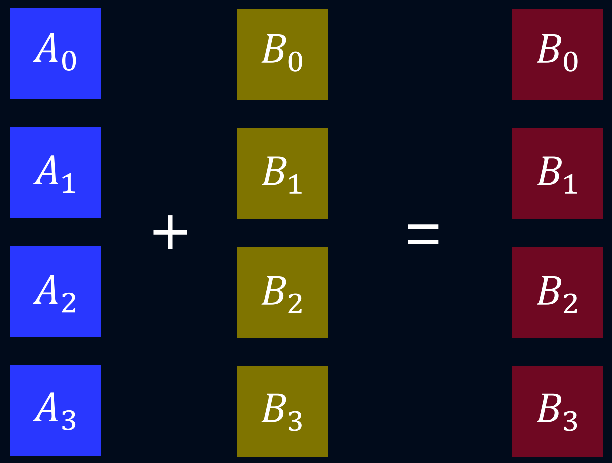
In-core parallelism

SIMD



Parallel execution within
a single CPU core

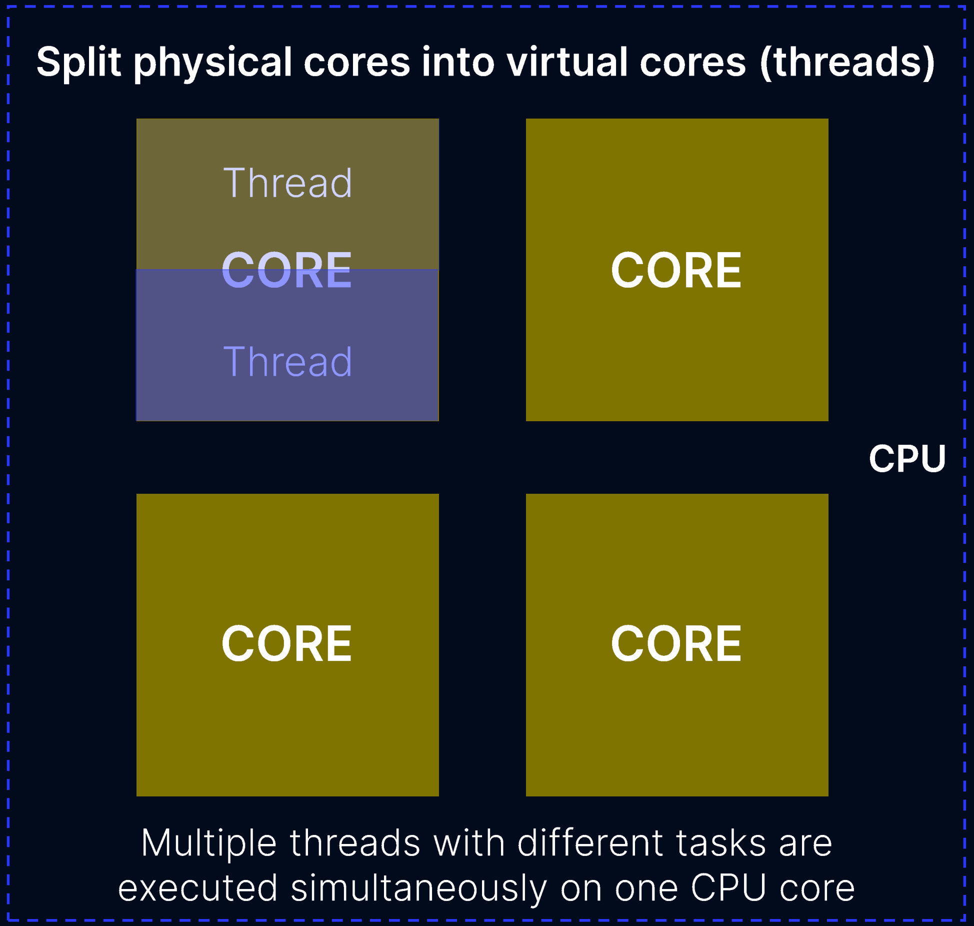
Operation with Vectors



Elements of the vector
are sum in parallel

SIMULTANEOUS MULTITHREADING (SMT)

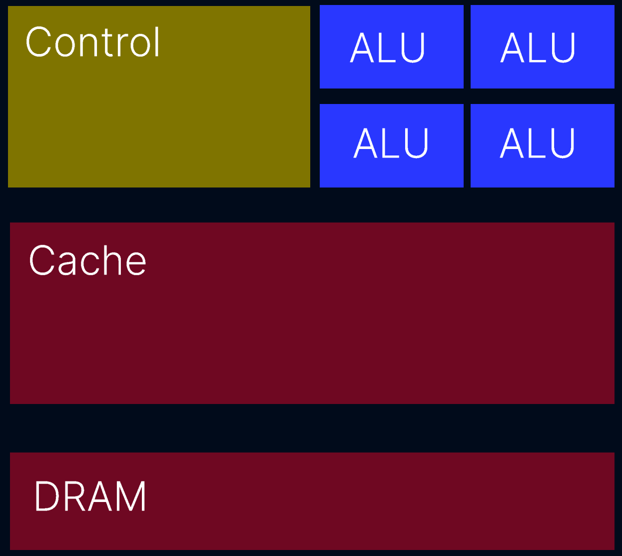
In-core parallelism



GRAPHICS PROCESSING UNIT (GPU) VS CPU

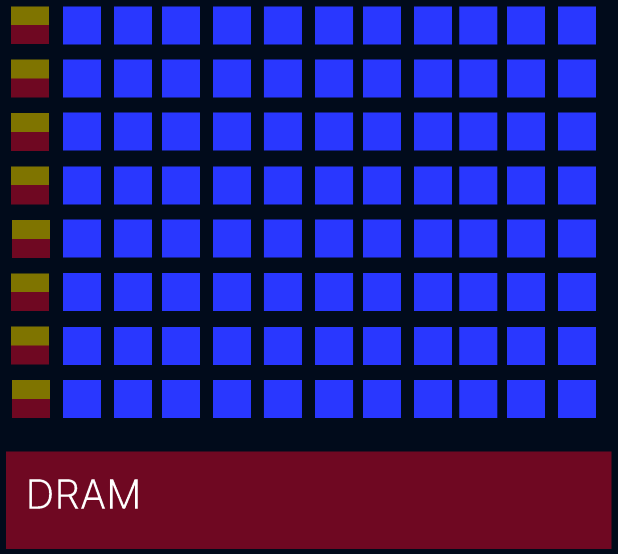
Made of many simple Cores

CPU



Transporting one

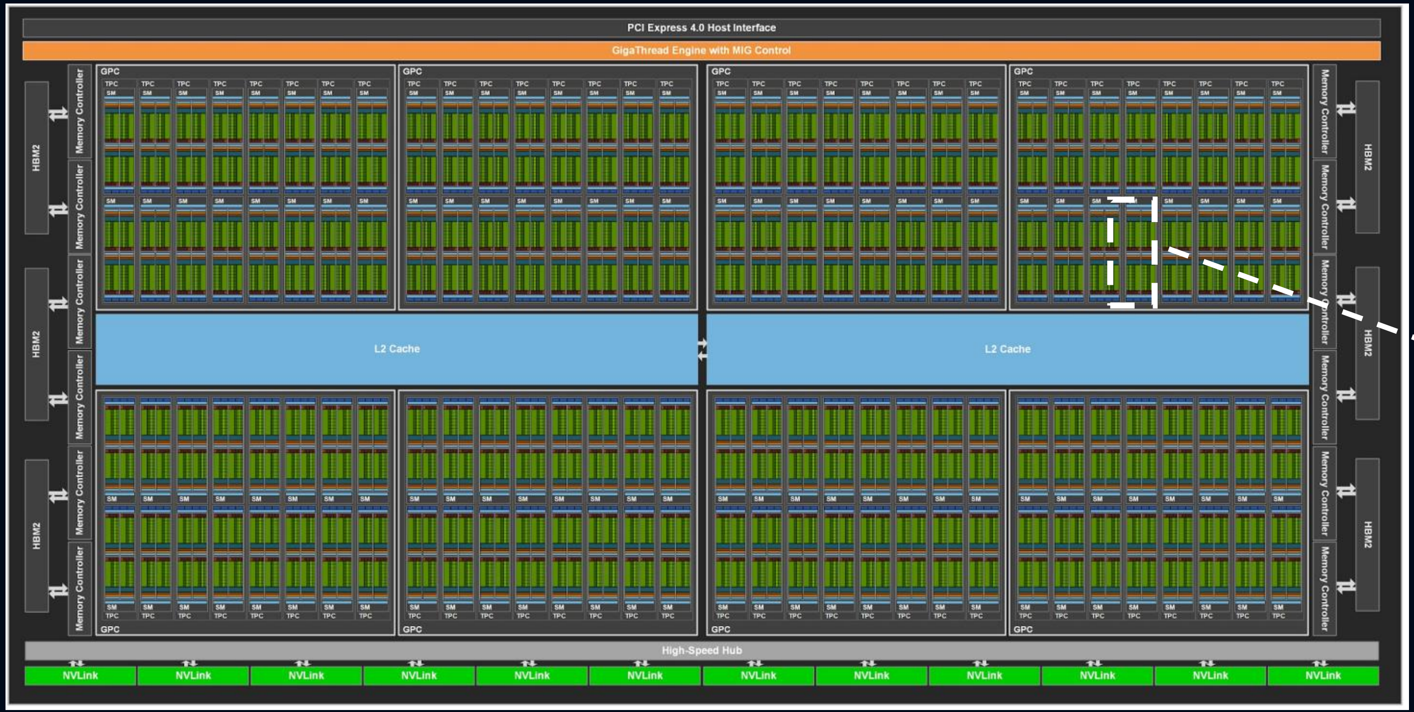
GPU



Transporting many

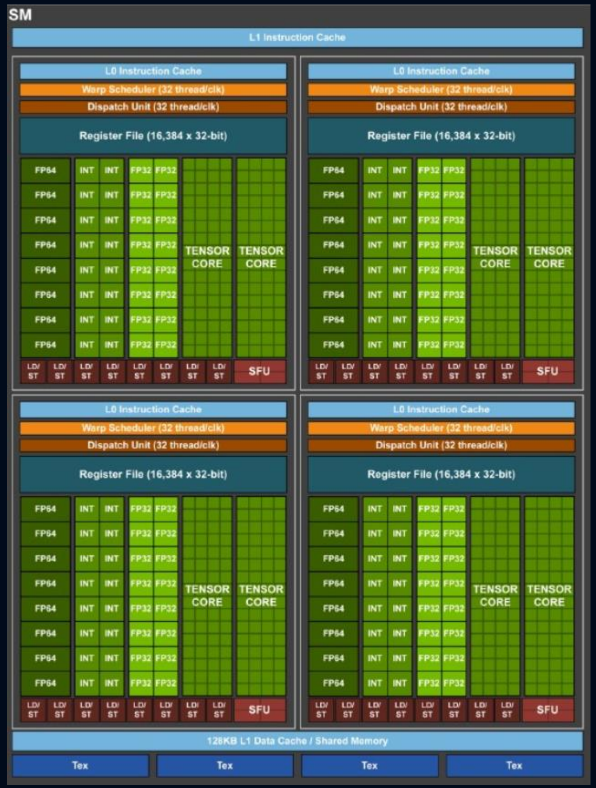
NVIDIA AMPERE GPU ARCHITECTURE

Streaming Multiprocessors (SMs)



NVIDIA Ampere Architecture
(128 SMs)

SM ≅ CPU core

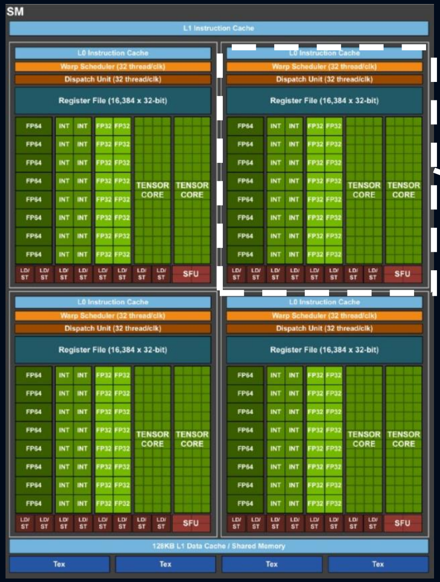


Ronny Krashinsky, et al., NVIDIA Ampere Architecture In-Depth
https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

NVIDIA AMPERE GPU ARCHITECTURE

Streaming Processors (SPs)

SM



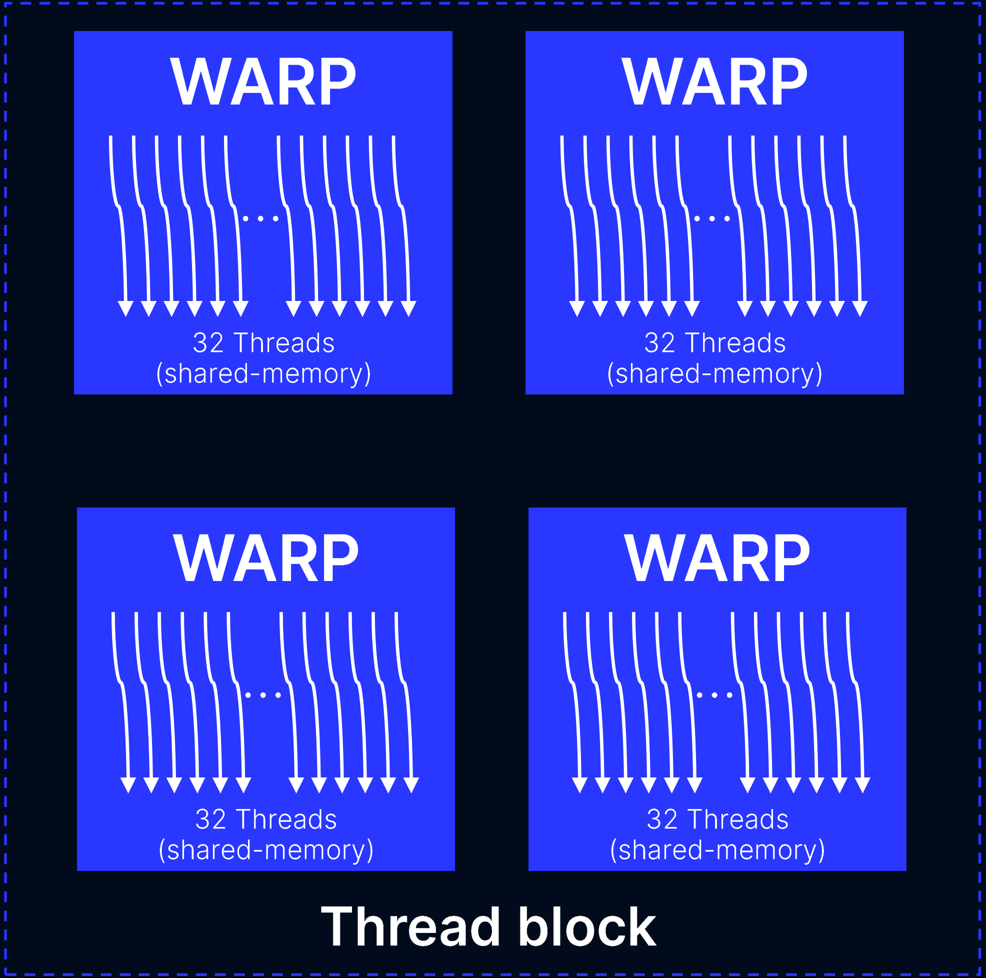
SP (CUDA core)

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)

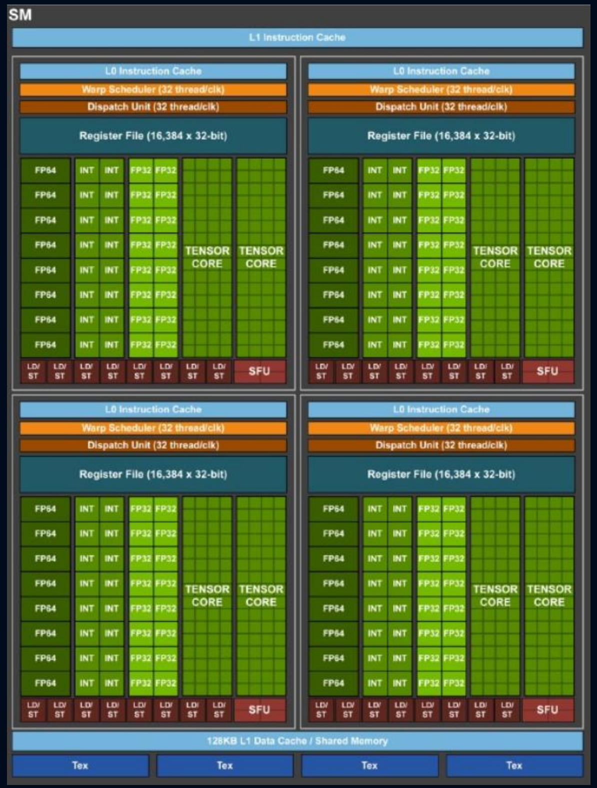
In-processor parallelism - SIMT= SIMD + SMT



The WARPS are executed simultaneously by a SM

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

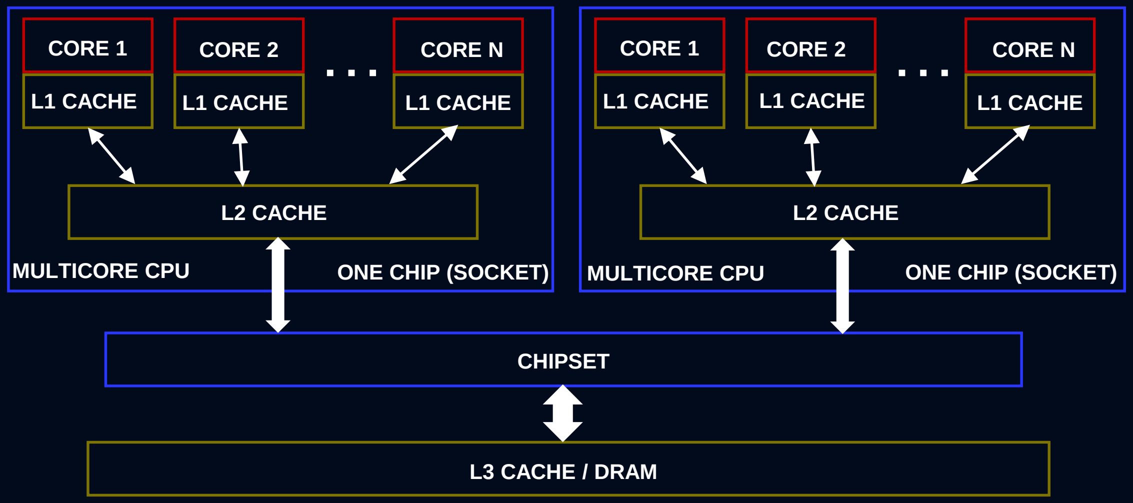
Streaming Multiprocessor (SM)



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

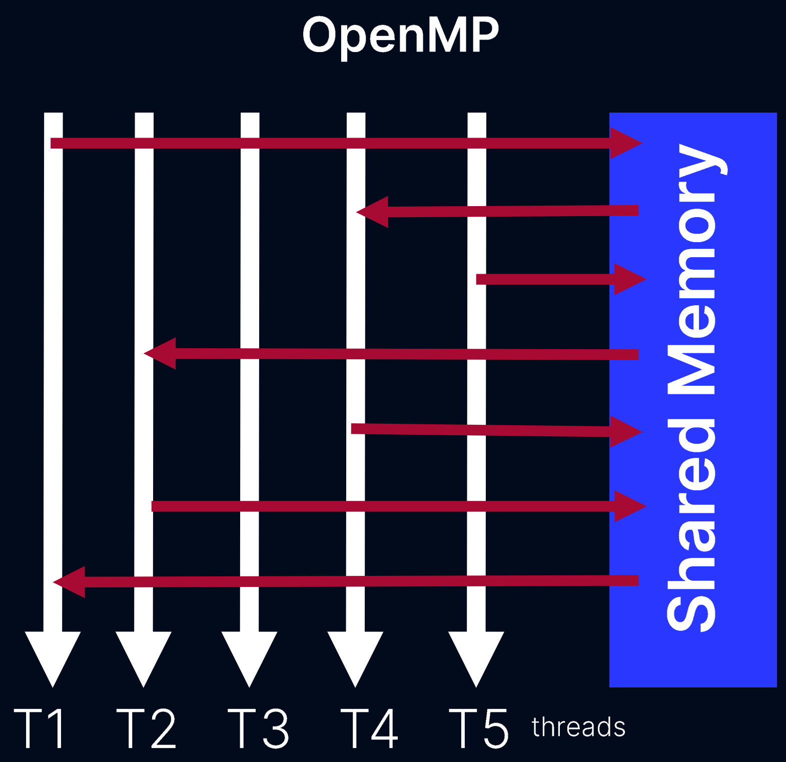
SHARED-MEMORY ARCHITECTURE

Single computer



PARALLEL PROGRAMMING MODEL: OPENMP

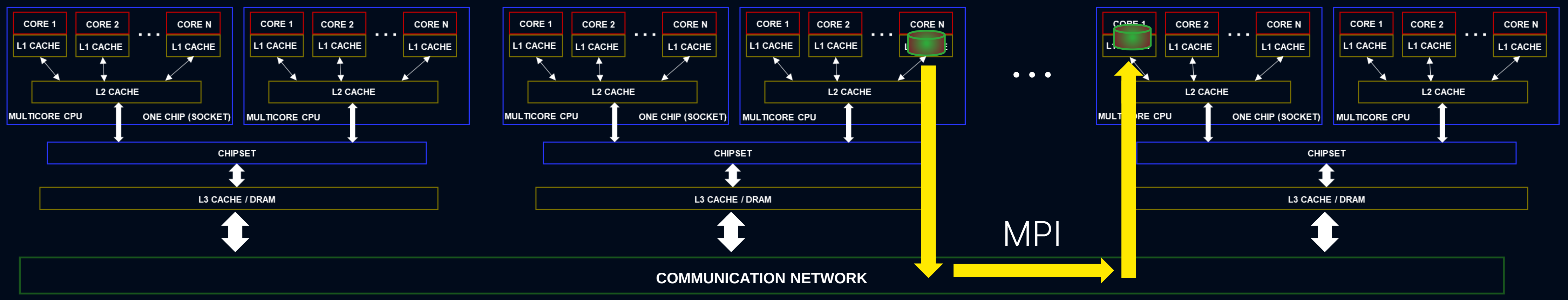
Single Computer



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

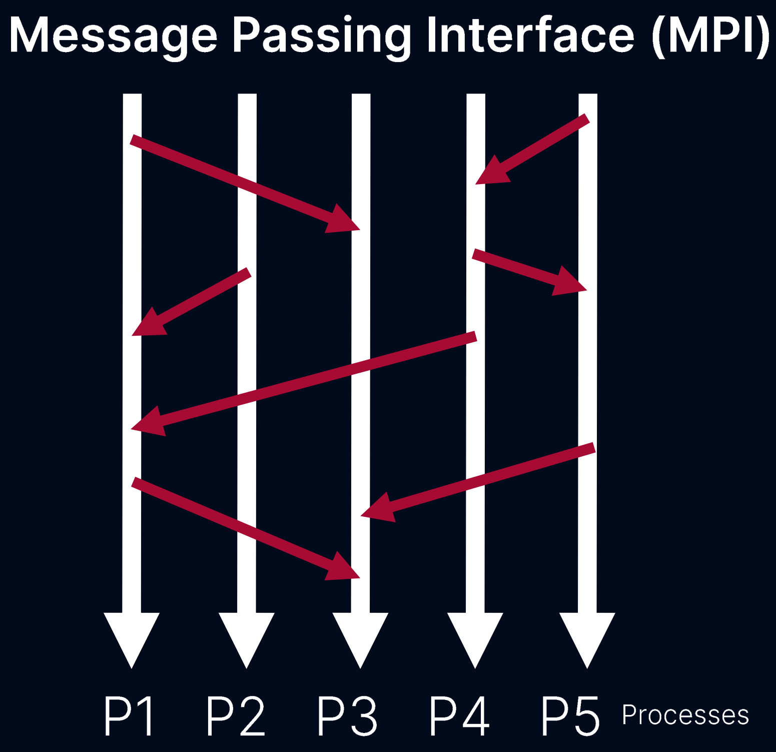
DISTRIBUTED-MEMORY ARCHITECTURE

Multiple Computers



PARALLEL PROGRAMMING MODEL: MPI

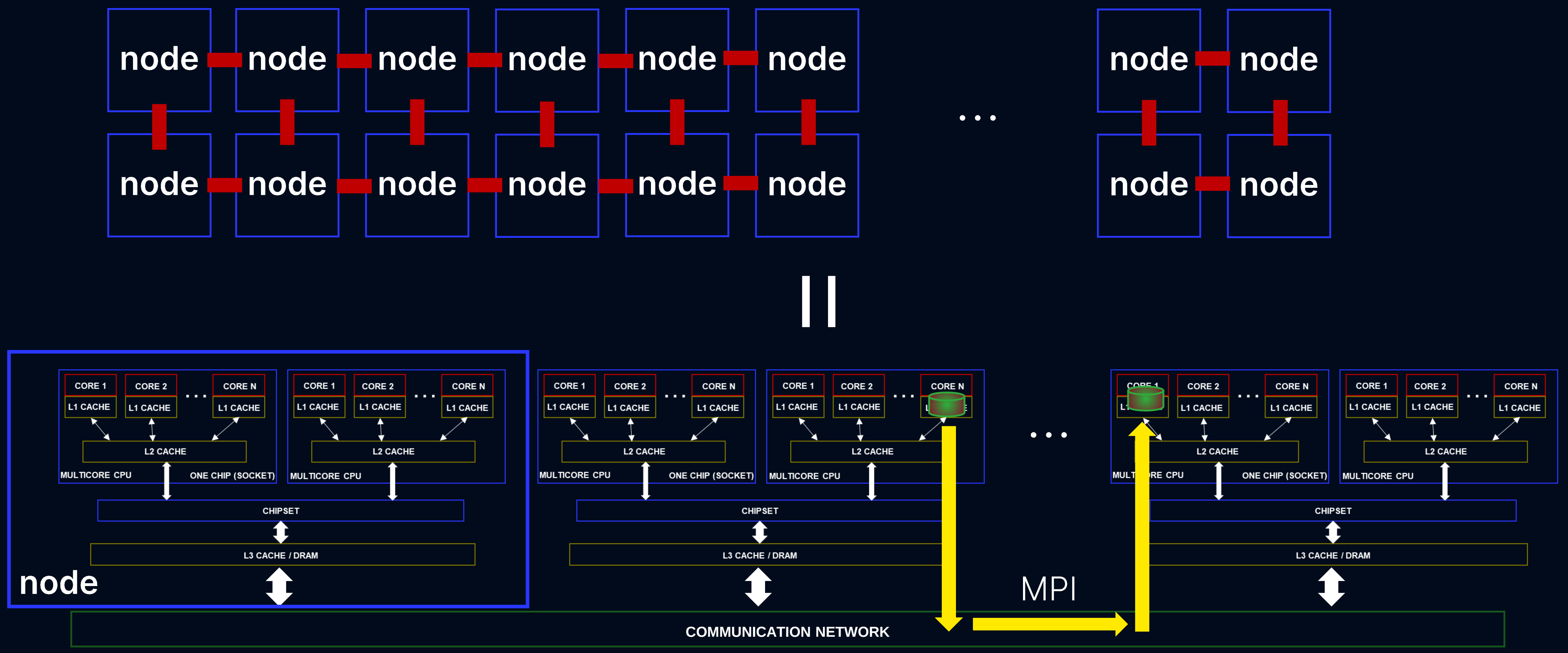
Multiple Computers



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



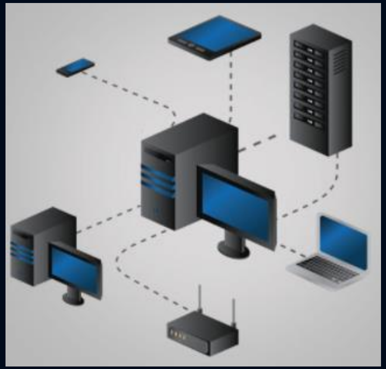
REVIEW ON HARDWARE LEVELS OF PARALLELISM

Best performance is achieved with a combination of them!

SIMD	SIMT	SMT	MPI	MPI+MSA
In-core parallelism	In-processor parallelism Many threads on many cores	Single Computer Simultaneous Multithreading Cross-core, Cross-socket OpenMP, pthreads	Multiple “Computers” Tightly-coupled Supercomputing	Multiple HPC Systems Tightly-coupled Heterogeneous Hardware

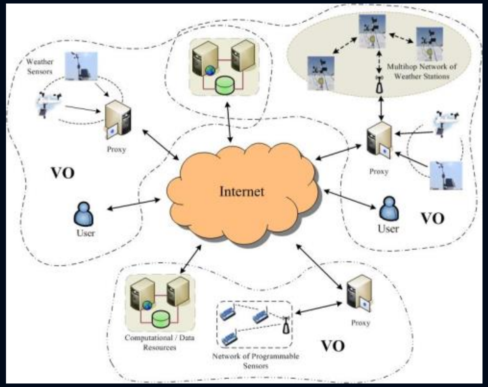
WHAT IS NOT A SUPERCOMPUTER?

Computer cluster



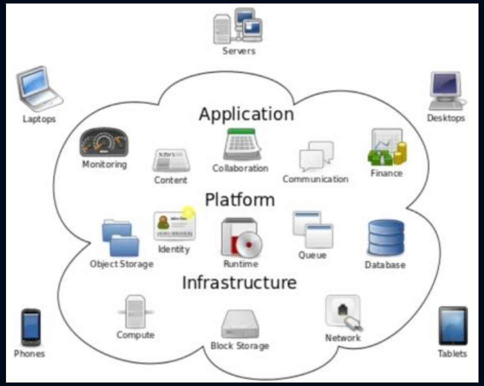
Many computers bound together locally

Grid computing



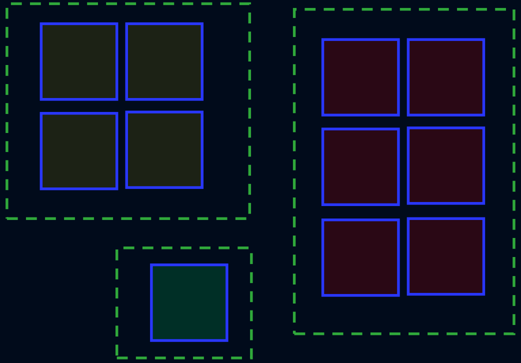
Many clusters working as some kind of supercomputer

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

June 2022

https://www.top500.org/

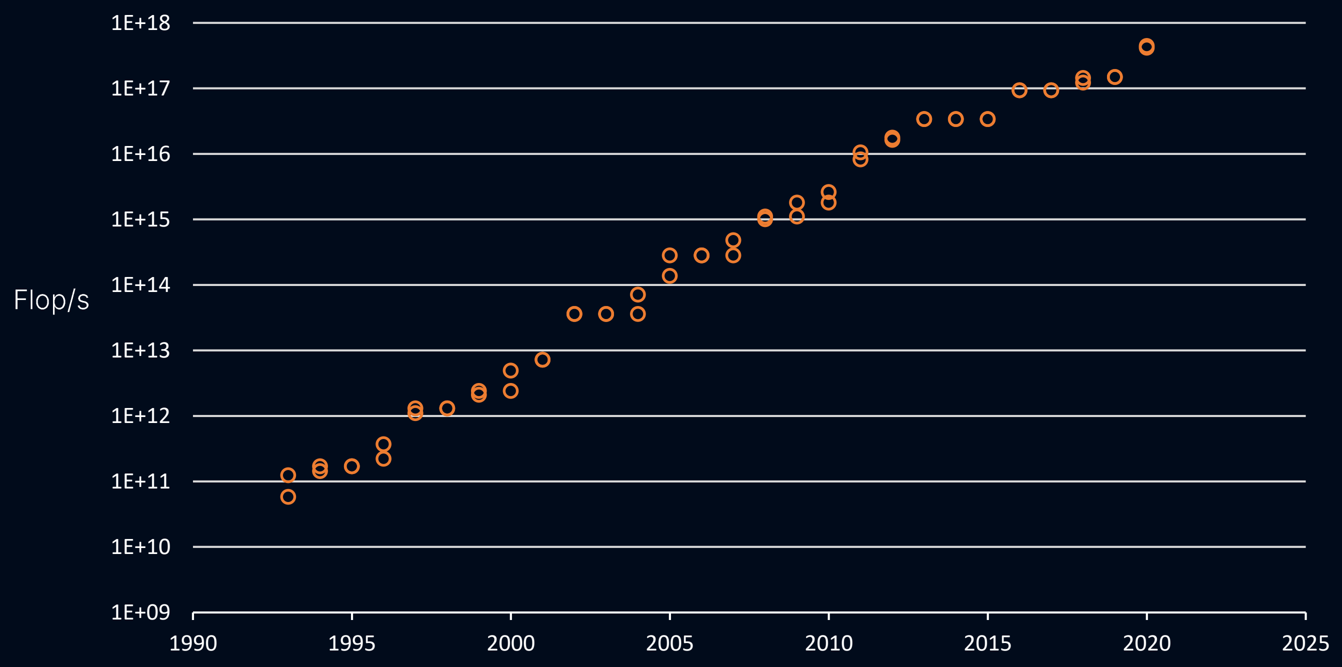
TOP500

1. Frontier — 1.1 PFlop/s
2. Fugaku — 442 Pflop/s
3. LUMI — 151.9 Pflop/s

GREEN500

1. Frontier TDS — 62.68 GFlops/watts
2. Frontier — 52.22 GFlops/watts
3. LUMI — 51.62 GFlops/watts

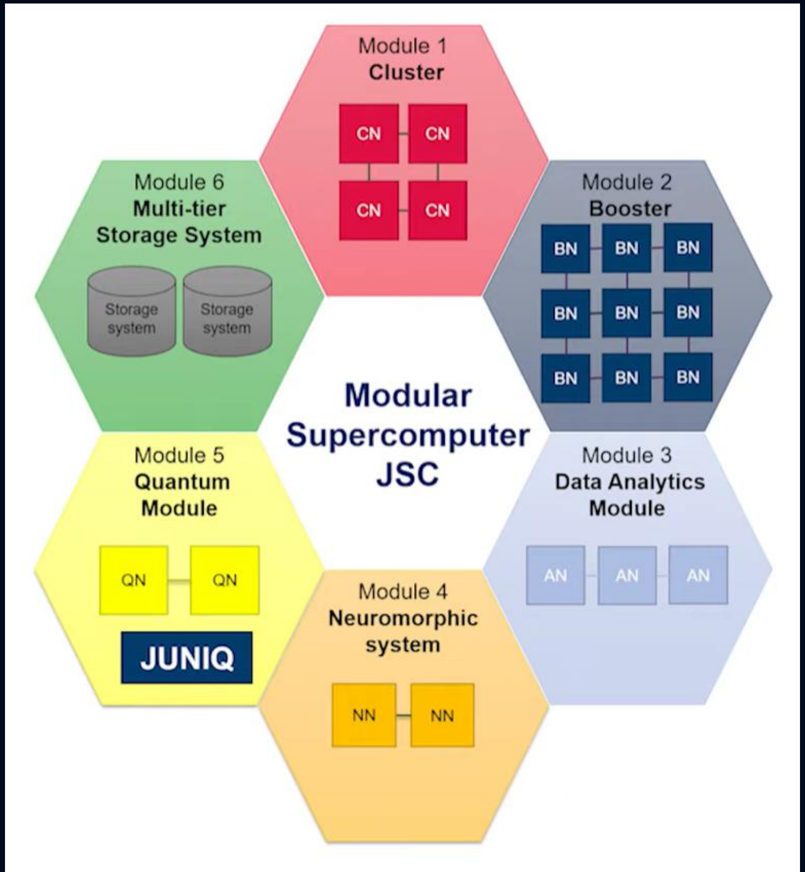
A RACE TOWARD EXASCALE COMPUTING



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

MODULAR SUPERCOMPUTING ARCHITECTURE (MSA)

Heterogeneous HPC clusters (modules) within a single system



PERSPECTIVE AT JSC: THE ROAD TO EXASCALE

Modular Supercomputing Architecture

MSA Pre-Exascale

2x MSA pilots

FP7 – H2020

EuroHPC

DEEP Projects ...

SEA Projects

MELUXINA

LEONARDO

EUPEX

⟨HPC|QS⟩

EXASCALE

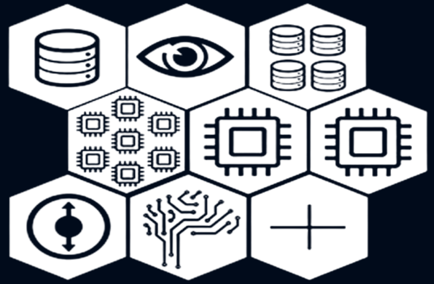
2011

2020

2022

2023

2024



Thomas Lippert "Exascale Ante Portas: Machines for Mankind",
Luxembourg Supercomputing Day 2022, https://youtu.be/nw4rVr4yChg

PRESS RELEASE | 15 June 2022 | European High-Performance Computing Joint Undertaking

EuroHPC Joint Undertaking announces five sites to host new world-class supercomputers

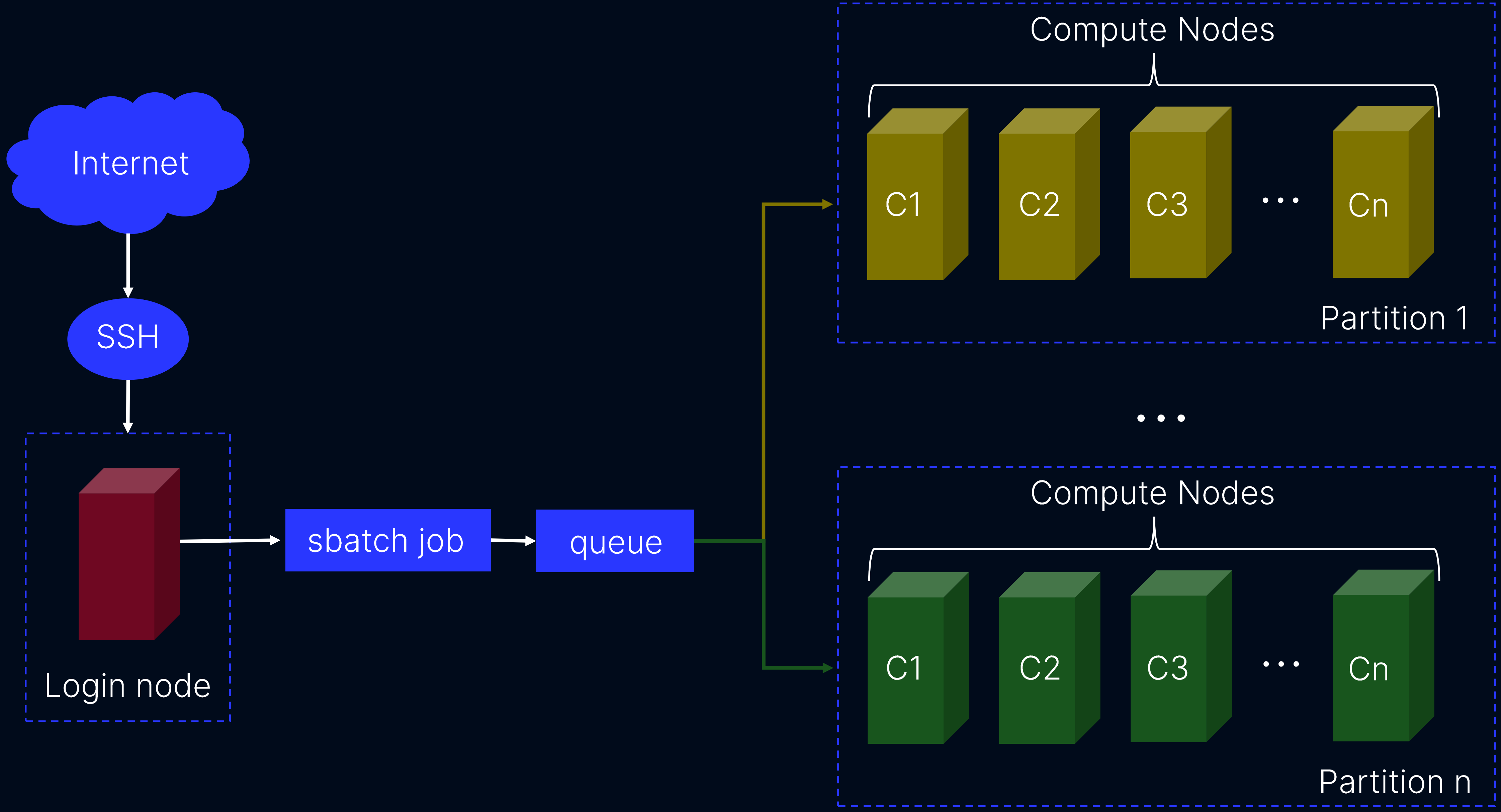
The European High Performance Computing Joint Undertaking (EuroHPC JU) has selected five new sites across the European Union (EU) to host and operate the next generation of EuroHPC supercomputers: Germany, Greece, Hungary, Ireland and Poland.

baranozdemir - iStock Getty Images Plus

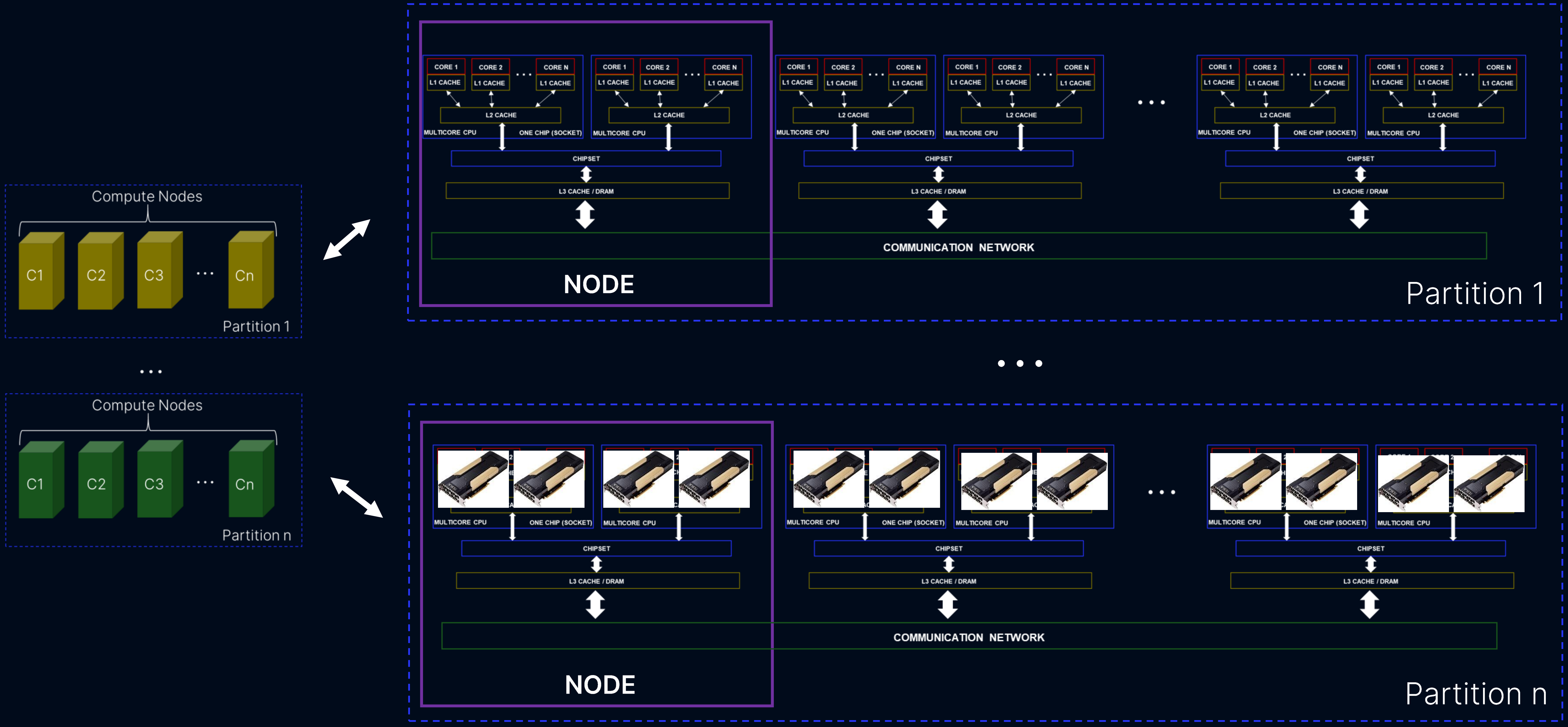
JUPITER, the first European exascale supercomputer, will be hosted by the Jülich Supercomputing Centre in Germany. Exascale supercomputers are systems capable of performing more than a billion billion calculations per second and represent a significant milestone for Europe. By supporting the development of high-precision models of complex systems, they will have a major impact on European scientific excellence.

https://eurohpc-ju.europa.eu/eurohpc-joint-undertaking-announces-five-sites-host-new-world-class-supercomputers-2022-06-15_en

ANATOMY OF A SUPERCOMPUTER



PARTITIONS AND NODES



KEY INFORMATION ABOUT A SUPERCOMPUTER

- TYPE OF PARTITIONS
- NUMBER OF NODES
- PERFORMANCE
- ACCELERATORS
- RAM
- NETWORK: BANDWIDTH, LATENCY
- CPUs, NUMBER OF CORES
- ...

JÜLICH RESEARCH ON EXASCALE CLUSTER ARCHITECTURES - DATA CENTRIC (JURECA-DC)

https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/jureca

- 98,304 CPU cores, 768 GPUs
- 3.54 (CPU) + 14.98 (GPU) Petaflops per second peak performance
- Mellanox InfiniBand HDR (HDR100/HDR) DragonFly+ network
  - Ca. 15 Tb/s connection to Booster via gateway nodes
- 350 GB/s network connection to JUST for storage access

JURECA-DC – CONFIGURATION

576 COMPUTE NODES

2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz

Partition 1

192 ACCELLERATED COMPUTE NODES

2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz

4× NVIDIA A100 GPU, 4× 40 GB HBM2e

Partition 2

19 LOGIN NODES

2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz

Partition 3

ACCESS SUPERCOMPUTERS AT JSC

Secure Shell (SSH)

Cryptographic protocol for operating services securely over an unsecured network

Jupyter-JSC

Supercomputing in your Browser

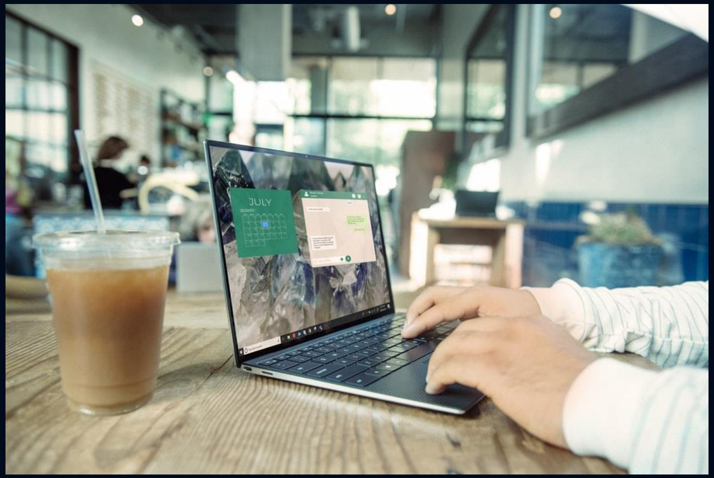
UNICORE

UNICORE

Uniform Interface to Computing Resources

SECURE SHELL (SSH)

Network protocol that gives users and system administrators a secure way to access a computer over an unsecured network



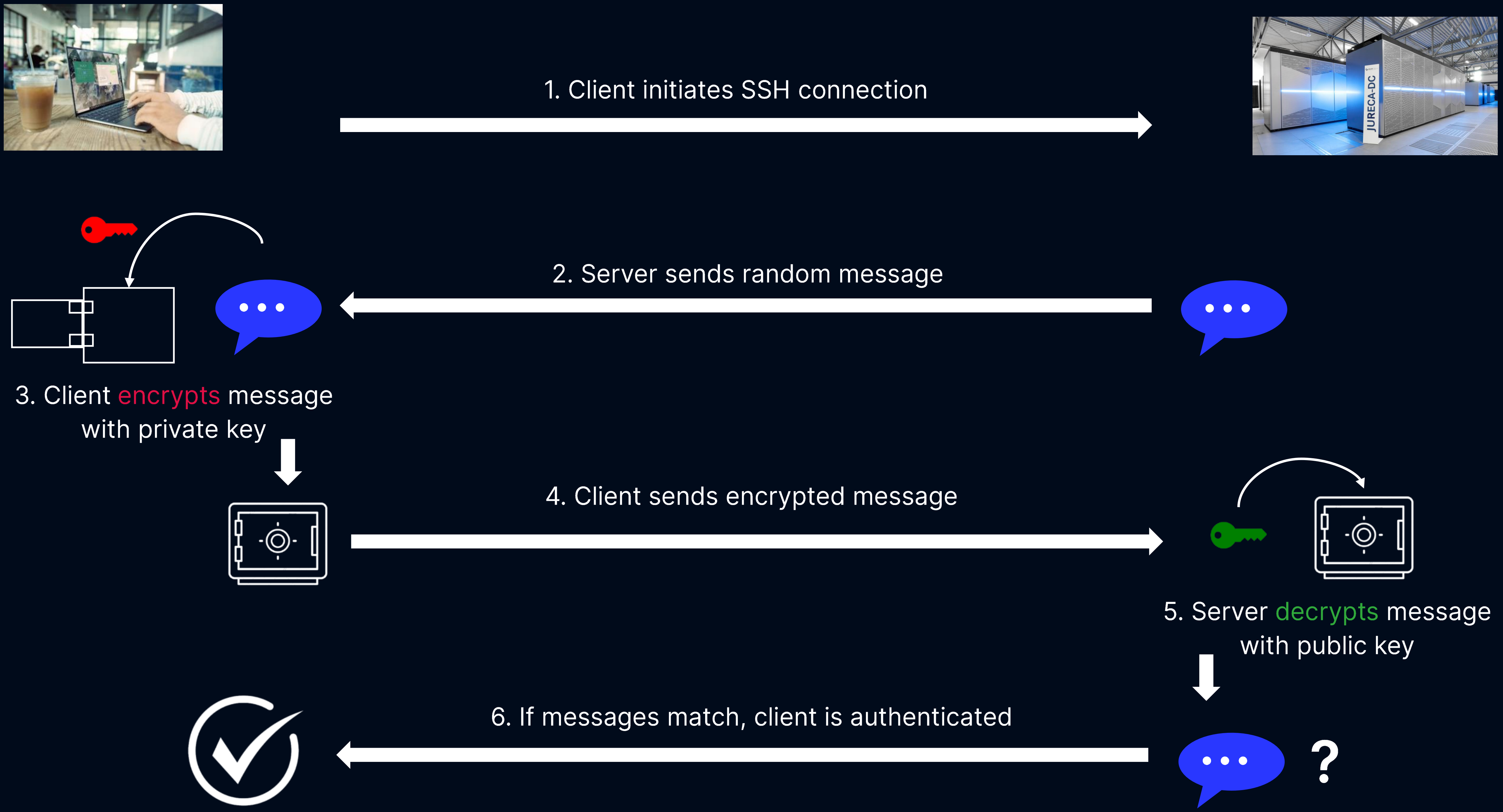
Private key

~/.ssh/id_rsa

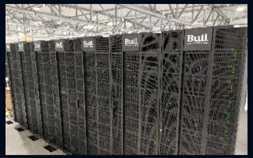
Public key

~/.ssh/id_rsa.pub

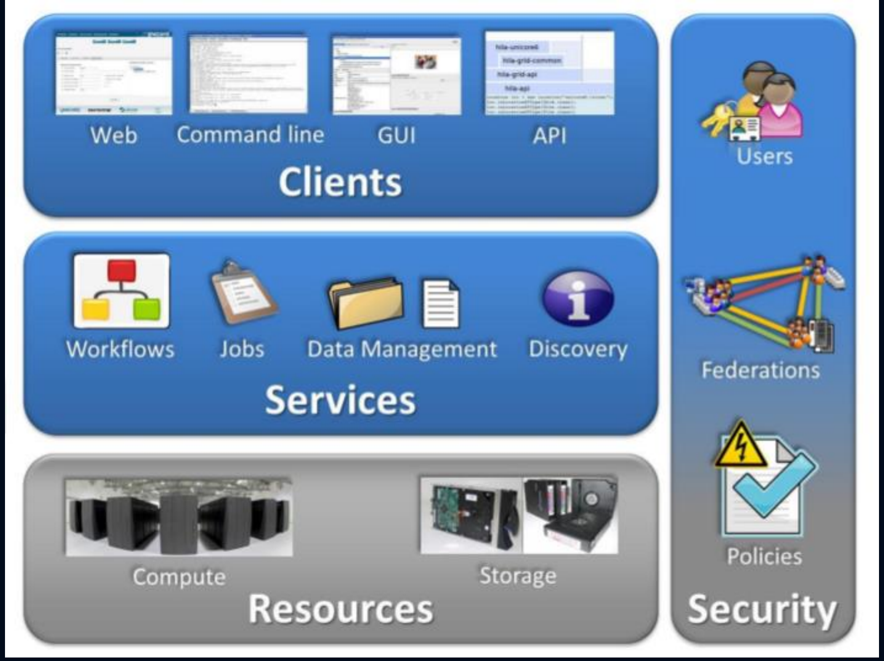
Public key


JURECA-DC
1. Client initiates SSH connection
2. Server sends random message
3. Client encrypts message with private key
4. Client sends encrypted message
5. Server decrypts message with public key
6. If messages match, client is authenticated
?

Different supercomputers can have different resource management system



...



Open Source: https://github.com/UNICORE-EU

UNICORE

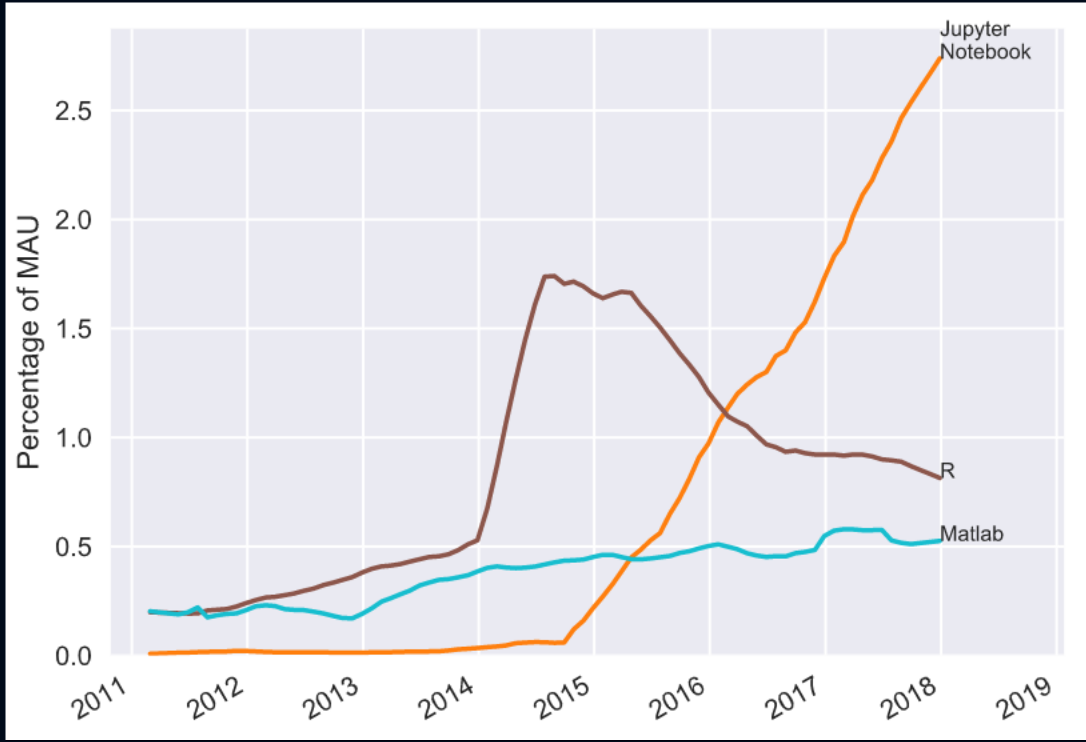
Unify and simplify supercomputer access

Hide system specific commands

Create, submit and monitor jobs access

RISE OF JUPYTER'S POPULARITY

Monthly aggregated number of user interactions with GitHub repos (= Monthly Active Users (MAU))

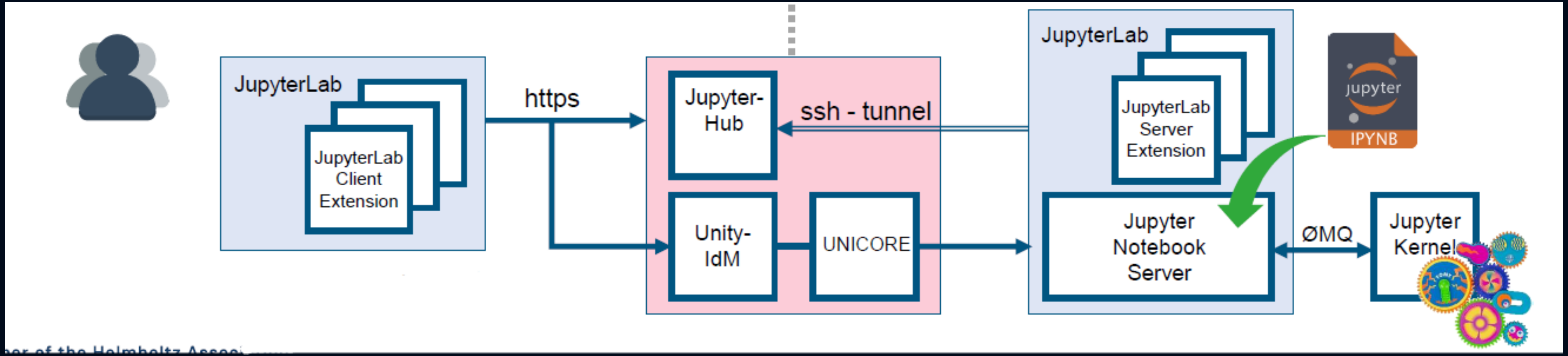


https://www.benfrederickson.com/ranking-programming-languages-by-github-users/

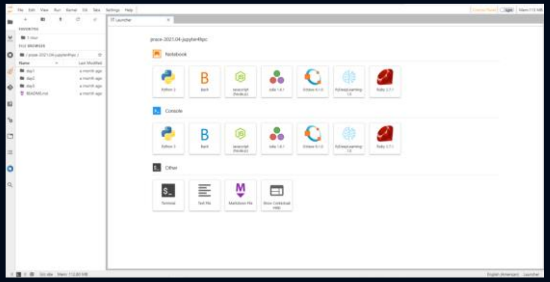
Jupyter Notebooks have seen significant and steady growth over the last years

Python is pushing this trend

JUPYTER-JSC WEB SERVICE



Browser



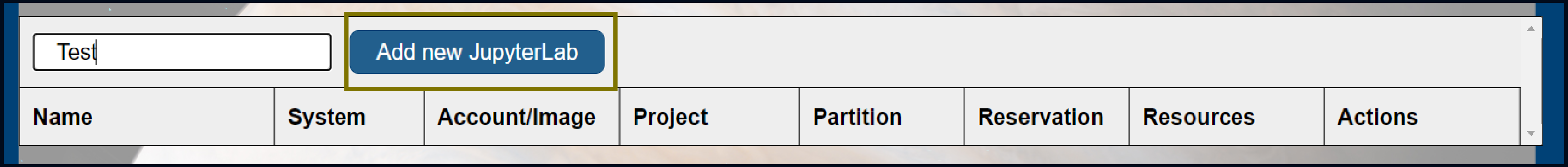
HPC system

ACCESS JUPYTER-JSC

Step 1

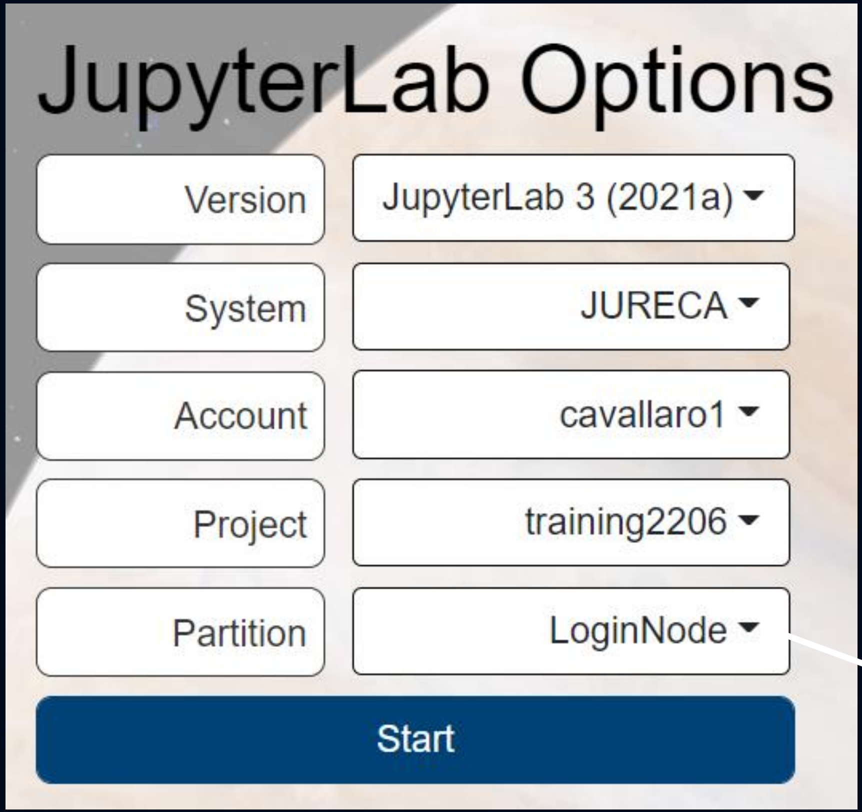
- Go to https://jupyter-jsc.fz-juelich.de/, login

- Add a new JupyterLab



ACCESS LOGIN NODE

Step 2



576 COMPUTE NODES

2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz

Partition 1

192 ACCELLERATED COMPUTE NODES

2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz

4× NVIDIA A100 GPU, 4× 40 GB HBM2e

Partition 2

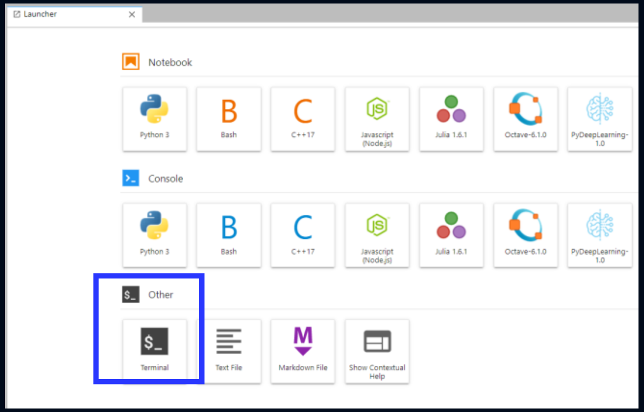
19 LOGIN NODES

Partition 3

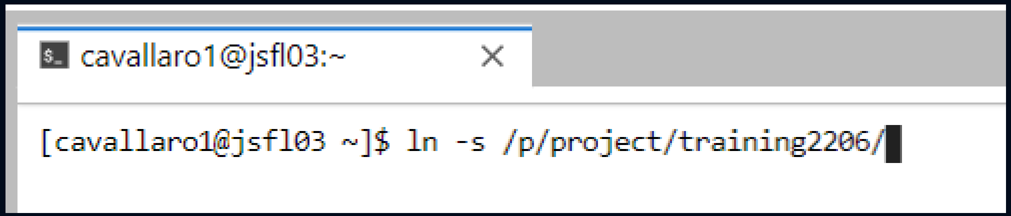
OBTAIN THE PROJECT FOLDER

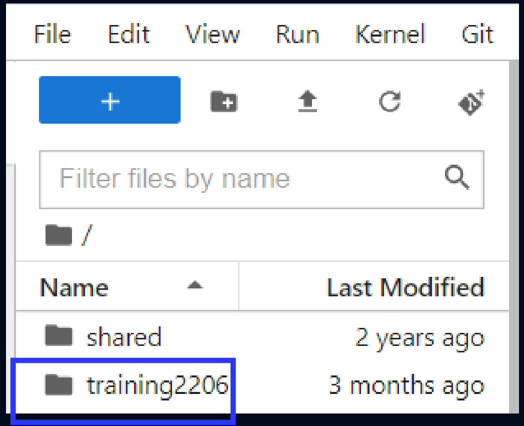
Step 3

- Open terminal



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





CREATE YOUR OWN FOLDER / WORKSPACE

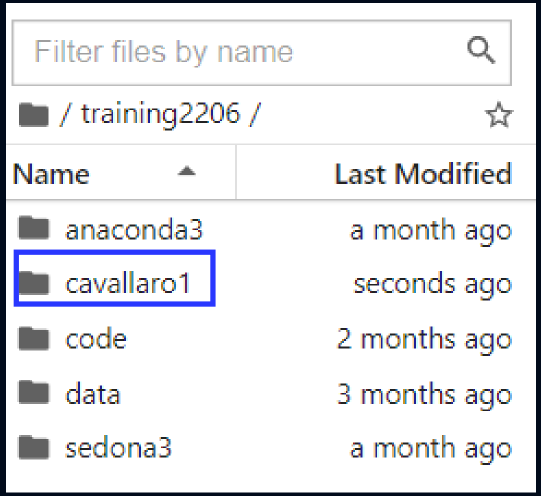
Step 4

- Run the following commands in the terminal to navigate to the tutorial folder "training2206" and create your own folder

```
cavallaro1@jsfl03:/p/project/

[cavallaro1@jsfl03 ~]$ cd /p/project/training2206/
[cavallaro1@jsfl03 training2206]$ mkdir $USER
[cavallaro1@jsfl03 training2206]$
```

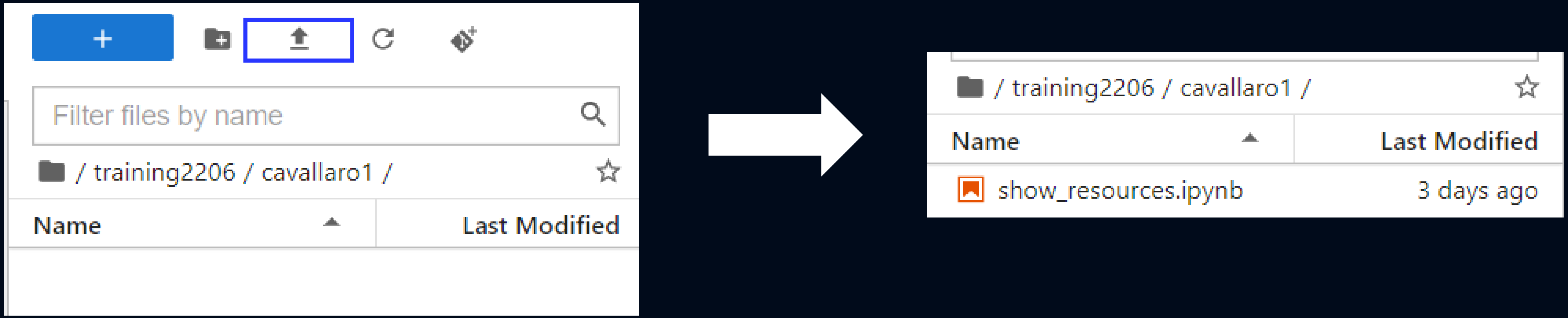
- You can check if the folder was created by looking at the navigator on the left



GET A JUPYTER NOTEBOOK

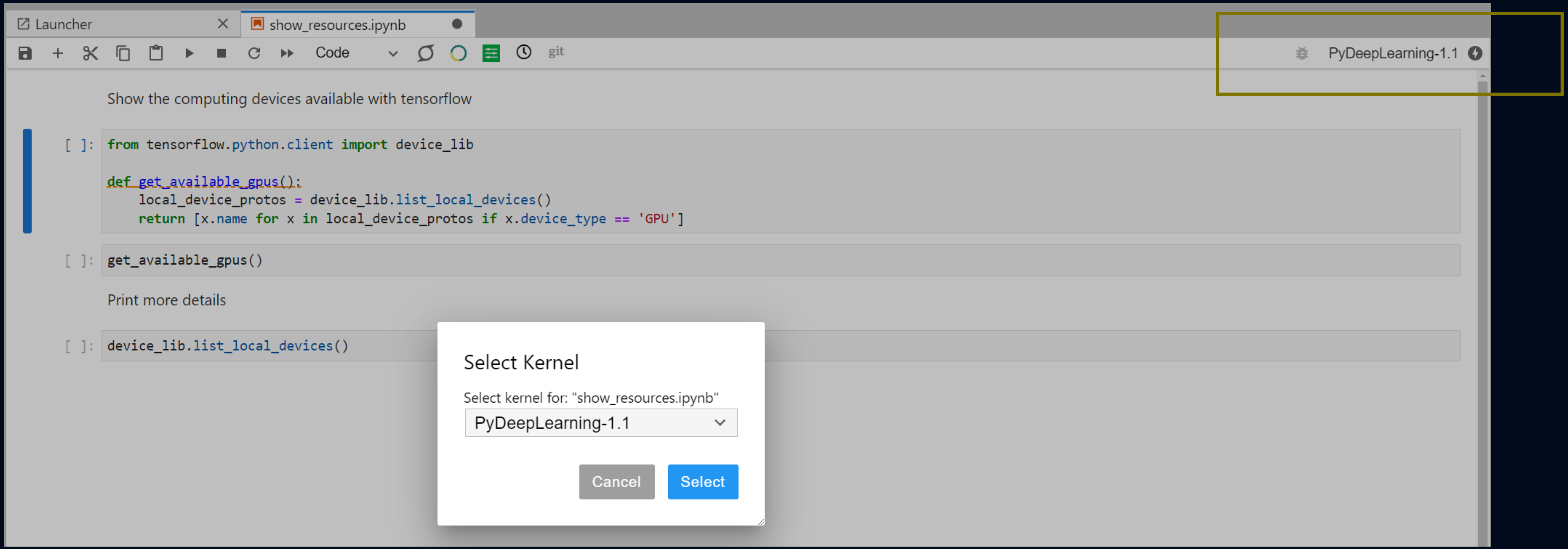
Step 5

- Get the Jupyter notebook “show_resources.ipynb” from the teaching material
  - https://gitlab.jsc.fz-juelich.de/hedgedoc/s/E1Vtpo0TS
- Upload the Jupyter notebook in your own folder



SELECT THE KERNEL

Step 6



RUN AND CHECK THE COMPUTING RESOURCES

Step 7

Show the computing devices available with tensorflow

```
[1]: from tensorflow.python.client import device_lib

def get_available_gpus():
    local_device_protos = device_lib.list_local_devices()
    return [x.name for x in local_device_protos if x.device_type == 'GPU']
```

```
2022-07-09 11:41:25.102199: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0
```

```
[2]: get_available_gpus()
```

```
2022-07-09 11:41:27.416254: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcuda.so.1
```

```
[2]: []
```

```
2022-07-09 11:41:27.838104: E tensorflow/stream_executor/cuda/cuda_driver.cc:328] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
2022-07-09 11:41:27.838170: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (jsfl02.jusuf): /proc/driver/nvidia/version does not exist
```

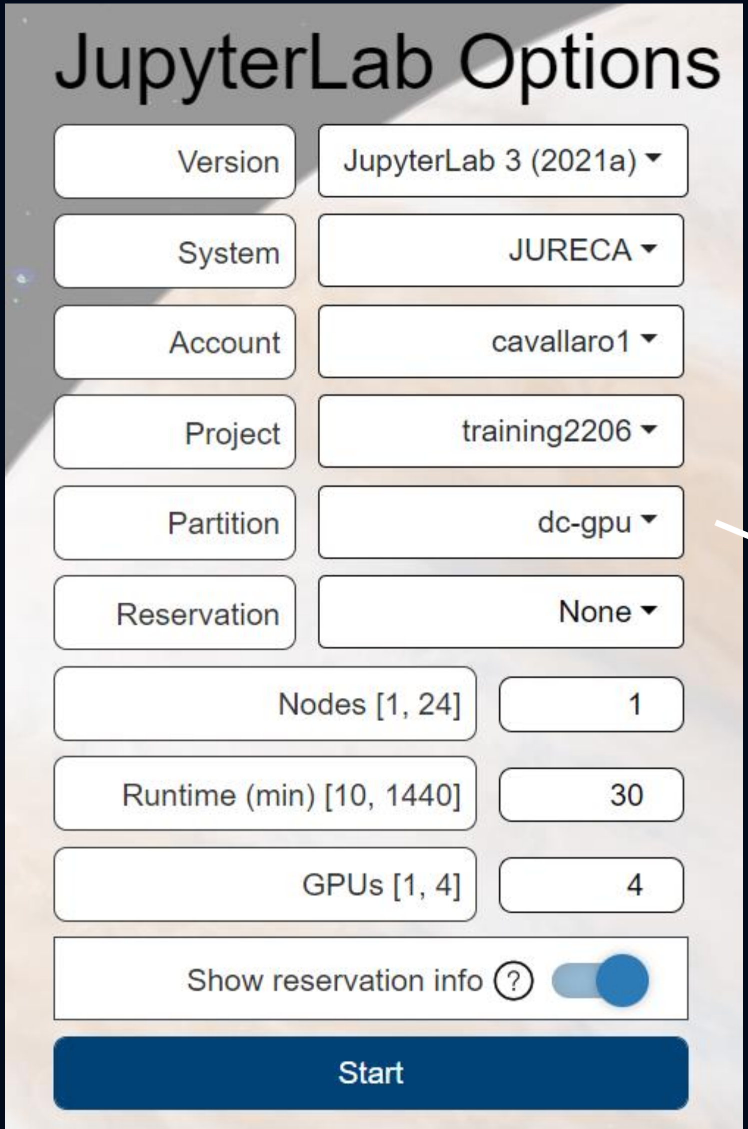
Print more details

```
[3]: device_lib.list_local_devices()
```

```
[3]: [name: "/device:CPU:0"
      device_type: "CPU"
      memory_limit: 268435456
      locality {
      }
      incarnation: 1455075978003438405]
```

ACCESS COMPUTE NODE

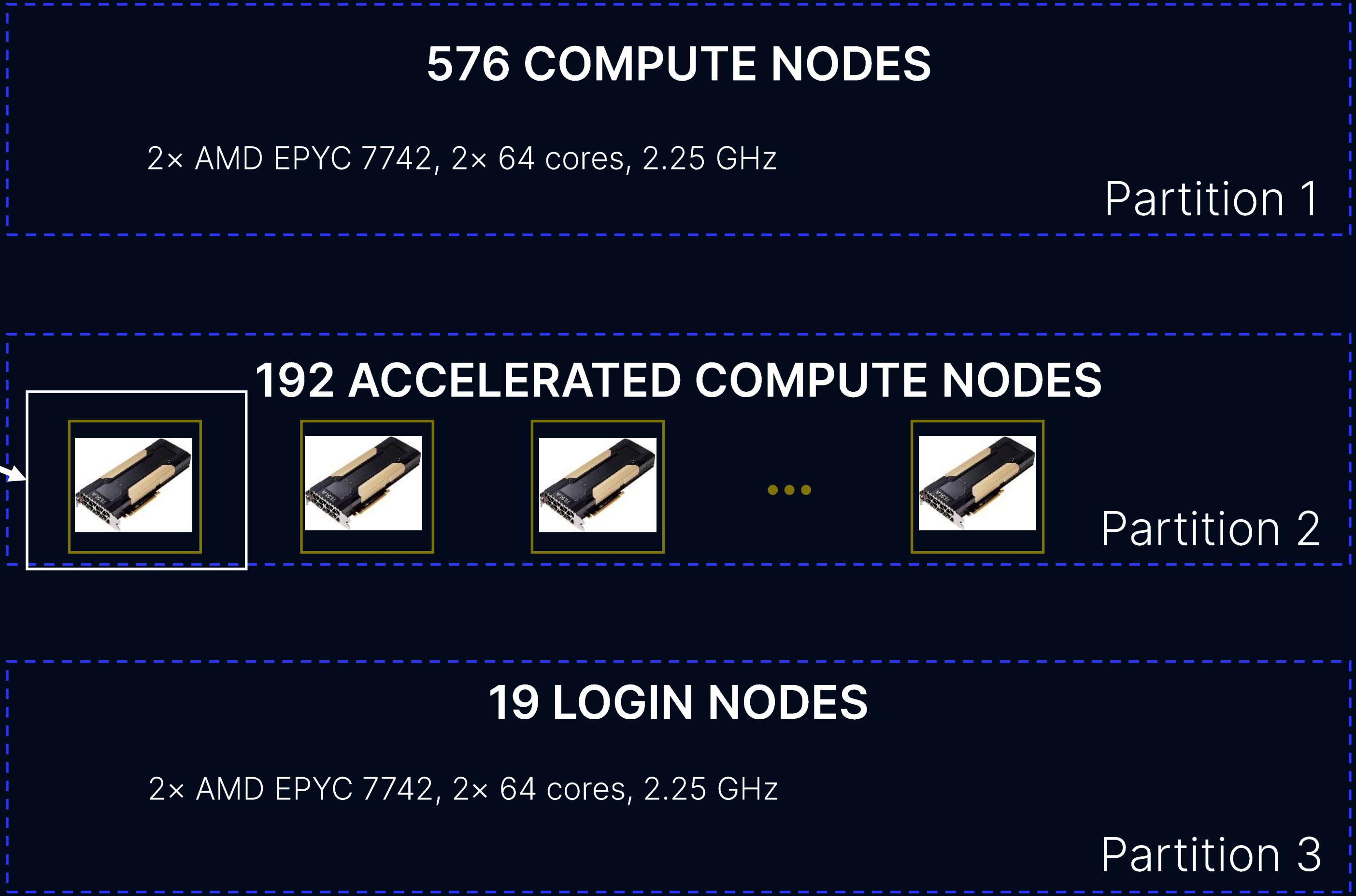
Step 8


JupyterLab Options

Version	JupyterLab 3 (2021a)
System	JURECA
Account	cavallaro1
Project	training2206
Partition	dc-gpu
Reservation	None
Nodes [1, 24]	1
Runtime (min) [10, 1440]	30
GPUs [1, 4]	4

Show reservation info

Start



RUN AND CHECK THE COMPUTING RESOURCES

Step 9

```
[2]:  ['/device:GPU:0', '/device:GPU:1', '/device:GPU:2', '/device:GPU:3']
```

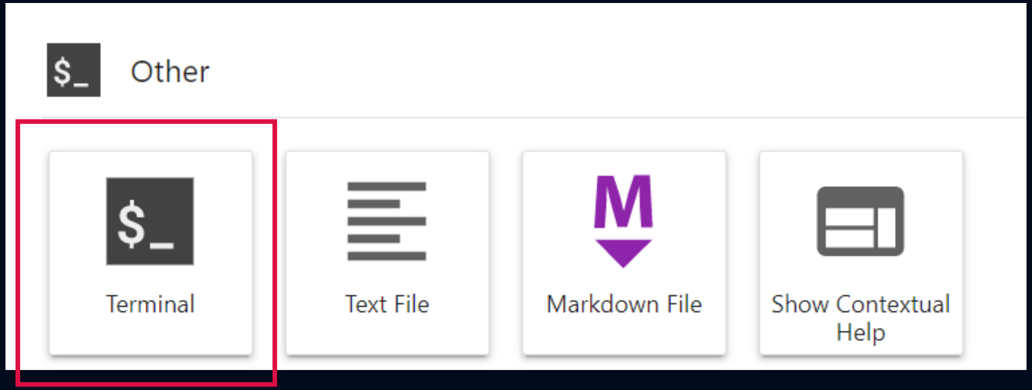
...

```
      }
    }
  }
  incarnation: 12690219694653453005
  physical_device_desc: "device: 0, name: NVIDIA A100-SXM4-40GB, pci bus id: 0000:03:00.0, compute capability: 8.0",
  name: "/device:GPU:1"
  device_type: "GPU"
  memory_limit: 40133197824
  locality {
    bus_id: 2
    numa_node: 1
    links {
      link {
        type: "StreamExecutor"
        strength: 1
      }
```

...

CHECK COMPUTING RESOURCES IN THE TERMINAL

Step 9



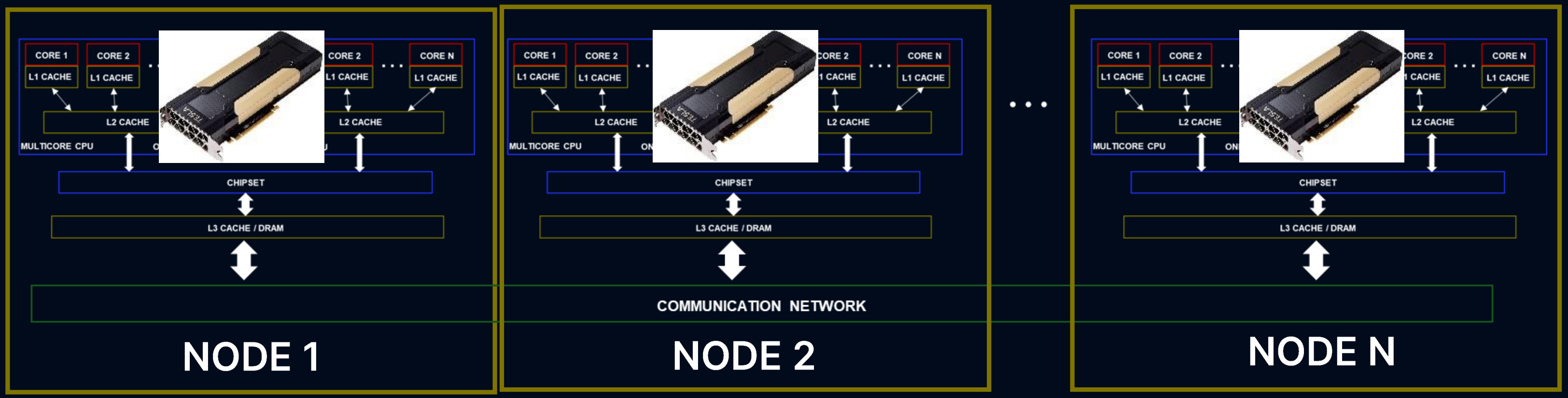

```
(base) [cavallaro1@jrc0197 ~]$ nvidia-smi
Sat Jul 16 14:18:00 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 510.47.03    Driver Version: 510.47.03    CUDA Version: 11.6     |
|-------------------------------+----------------------+----------------------+
| 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  NVIDIA A100-SXM...  On   | 00000000:03:00.0 Off |                    0 |
| N/A   48C    P0    68W / 400W |    544MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA A100-SXM...  On   | 00000000:44:00.0 Off |                    0 |
| N/A   48C    P0    63W / 400W |    544MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   2  NVIDIA A100-SXM...  On   | 00000000:84:00.0 Off |                    0 |
| N/A   46C    P0    62W / 400W |    544MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   3  NVIDIA A100-SXM...  On   | 00000000:C4:00.0 Off |                    0 |
| N/A   48C    P0    64W / 400W |    544MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      6188      C   python                            541MiB |
|    1   N/A  N/A      6188      C   python                            541MiB |
|    2   N/A  N/A      6188      C   python                            541MiB |
|    3   N/A  N/A      6188      C   python                            541MiB |
+-----------------------------------------------------------------------------+
(base) [cavallaro1@jrc0197 ~]$ 
```

THIS AFTERNOON

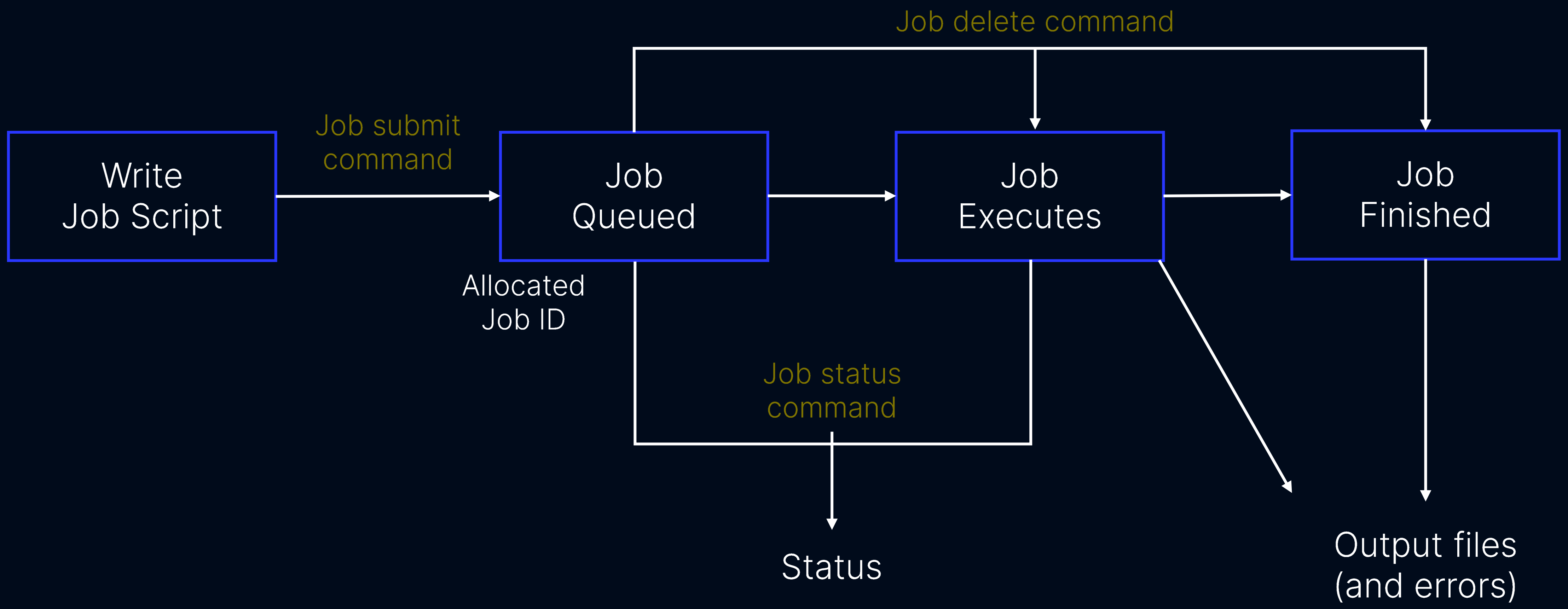
Hands-on - Distributed Deep Learning

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

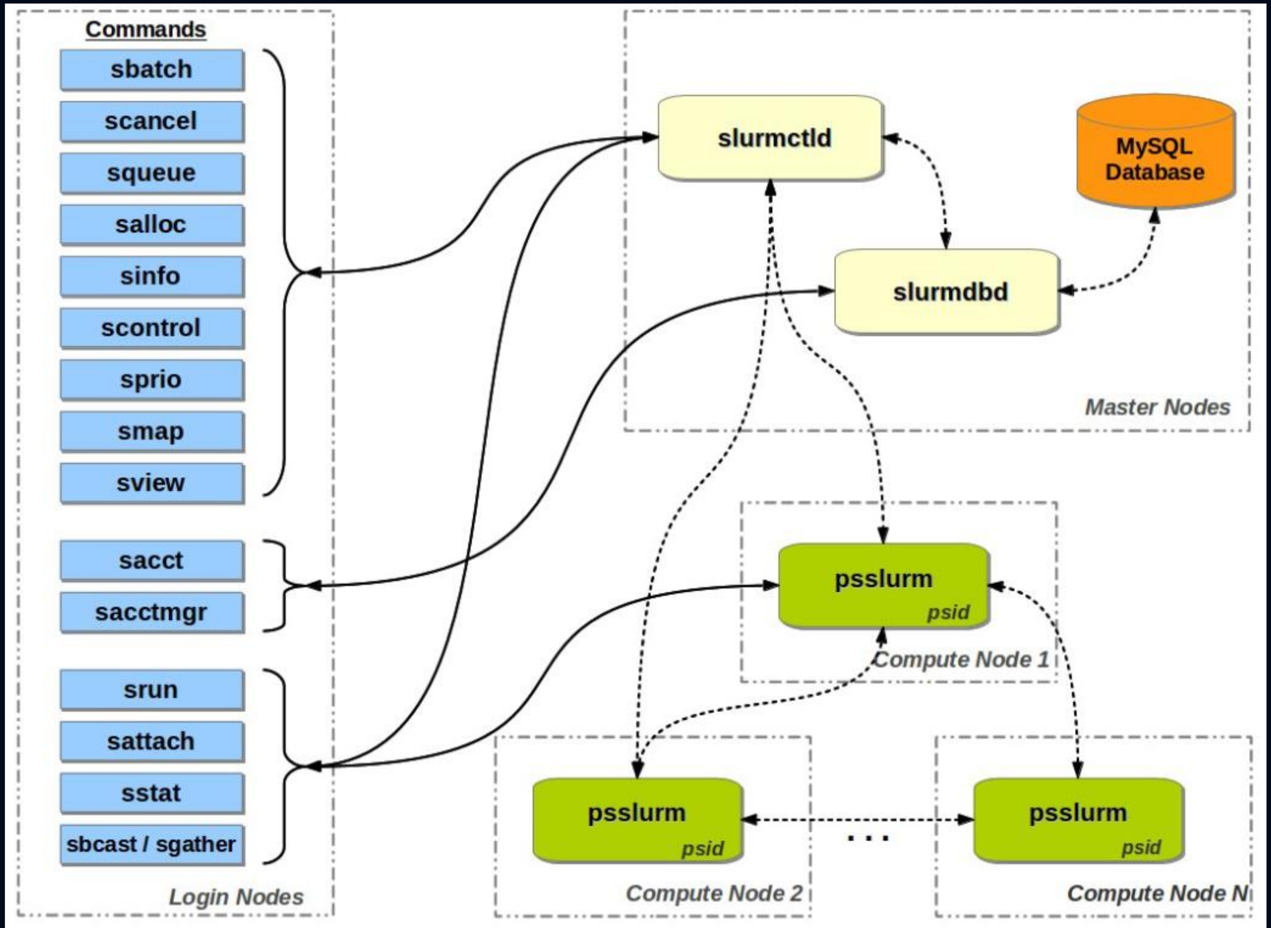


USING THE SUPERCOMPUTER MEANS SUBMITTING A JOB TO A BATCH SYSTEM

Job scheduling according to priorities. The jobs with the highest priorities will be scheduled next.



THE SLURM BATCH SYSTEM



NO NODE-SHARING

SUPERCOMPUTER USAGE MODEL AT JSC

Additional information

- Compute time allocation is based on compute projects. For every compute job, a compute project pays.
- Data projects allocate large amounts of storage, but no compute.
- To enable fair share, only relatively short job runs (24h) are allowed.
  - Please implement check pointing (or make your code fast enough).
- Solution for long-running tasks: Job arrays.

More information

https://cstao-public.pages.jsc.fz-juelich.de/JSCHandsOn/

MODULE SYSTEM: SOFTWARE

All kind of software is already installed in modules.

Compiler/GCCcore/9.3.0

AOCC/2.3.0
Autotools/20200321
Bazel/3.4.1
Bazel/3.6.0
Boost.Python/1.74.0-nompi
Boost/1.74.0-nompi
CFITSIO/3.490
CMake/3.18.0
CVS/1.11.23
Cirq/0.9.1-Python-3.8.5
Clang/11.0.0
Cling/0.7
Cling/0.9
CubeGUI/4.5
CubeGUI/4.6
CubeLib/4.5
CubeLib/4.6
DWave/3.2.0-Python-3.8.5
Doxygen/1.8.18
Eigen/3.3.7
Emacs/27.1
FriBidi/1.0.9
GDB/10.1
GEOS/3.8.1-Python-3.8.5
JupyterKernel-Ruby/2.7.1-2020.2.6
JupyterProxy-Matlab/0.1.0-2020.2.6
JupyterProxy-XpraHTML5/0.3.0-2020.2.6
LLVM/10.0.1
METIS/5.1.0-IDX64
MPFR/4.1.0
Mercurial/5.5.2-Python-3.8.5
Meson/0.55.0-Python-3.8.5
NCCL/2.8.3-1-CUDA-11.0
NSPR/4.25
NSS/3.51
Ninja/1.10.0
Nsight-Compute/2020.1.2
Nsight-Compute/2020.2.0
Nsight-Compute/2020.3.0
Nsight-Systems/2020.3.1
Nsight-Systems/2020.4.1
Nsight-Systems/2020.5.1
Nsight-Systems/2021.1.1
Octave/6.1.0-nompi
OpenAI-Gym/0.18.0-Python-3.8.5
OpenCV/4.5.0-Python-3.8.5
OpenEXR/2.5.2
git/2.28.0
gnuplot/5.2.8
h5py/2.10.0-serial-Python-3.8.5
hwloc/2.2.0
jemalloc/5.2.1
libvpx/1.9.0
likwid/5.0.2
likwid/5.1.0
magma/2.5.4
meld/3.21.0-Python-3.8.5
memkind/1.10.1
nano/5.5
netCDF-C++4/4.3.1-serial
netCDF-Fortran/4.5.3-serial
netCDF/4.7.4-serial
netcdf4-python/1.5.4-serial-Python-3.8.5
numba/0.51.1-Python-3.8.5
parallel/20201122
pyproj/2.6.1.post1-Python-3.8.5
qcint/3.0.19
re2c/1.3

https://apps.fz-juelich.de/jsc/llview/juwels_modules_booster/

MODULE SYSTEM: COMMANDS

module avail

module purge # unload everything

module load

module spider

module avail

```
----------------------------------------------- Core packages ------------------------------------------------
   Advisor/2020_update3                          Python/3.8.5
   Autotools/20200321                            R/4.0.2-nompi
   Autotools/20200321                 (D)        Ruby/2.7.1
   Bazel/3.6.0                                   Rust/1.47.0
   Blender/2.90.1-binary                         SciPy-Stack/2020-Python-3.8.5
   CFITSIO/3.490                                 Shapely/1.7.1-Python-3.8.5
   CMake/3.18.0                                  Singularity-Tools/2020-Python-3.8.5
   CUDA/11.0                          (g)        StdEnv/2020                           (L)
   CVS/1.11.23                                   Subversion/1.14.0
   Cling/0.7                                     Tcl/8.6.10
   CubeGUI/4.5                                   TensorFlow/2.3.1-Python-3.8.5
   Doxygen/1.8.18                                TotalView/2020.1.13
   EasyBuild/4.2.1                               UCX/1.8.1
   EasyBuild/4.2.2                               UCX/1.9.0                             (D)
   EasyBuild/4.3.0                    (D)        VTune/2019_update8
   Eigen/3.3.7                                   Vampir/9.9.0
   Emacs/27.1                                    VirtualGL/2.6.4
   FriBidi/1.0.9                                 Voro++/0.4.6
   GDB/10.1                                      X11/20200222
```

module spider nano

```
[kesselheim1@jwlogin07 ~]$ module spider nano

----------------------------------------------------------------------------------------------------
  nano: nano/5.5
----------------------------------------------------------------------------------------------------
    Description:
      GNU nano is a small and friendly text editor. Besides basic text editing, nano offers features like
      undo/redo, syntax coloring, interactive search-and-replace, auto-indentation, line numbers, word
      completion, file locking, backup files, and internationalization support.
```

module load nano

JUST - JÜLICH STORAGE CLUSTER

- 52 PB total storage
- $HOME: Minimal, only small files
- $SCRATCH: 9 PB, Working storage, deleted after 90 days
- $DATA: 14 PB, Large scale file storage, not available from compute nodes
- $FASTDATA: 9 PB, Large scale file storage that must be continuously accessed
- $PROJECT: 2.3 PB, Project data
- Peak bandwidth up to 400 GB/sec
- Data is shared among all supercomputers

Thank you for your attention