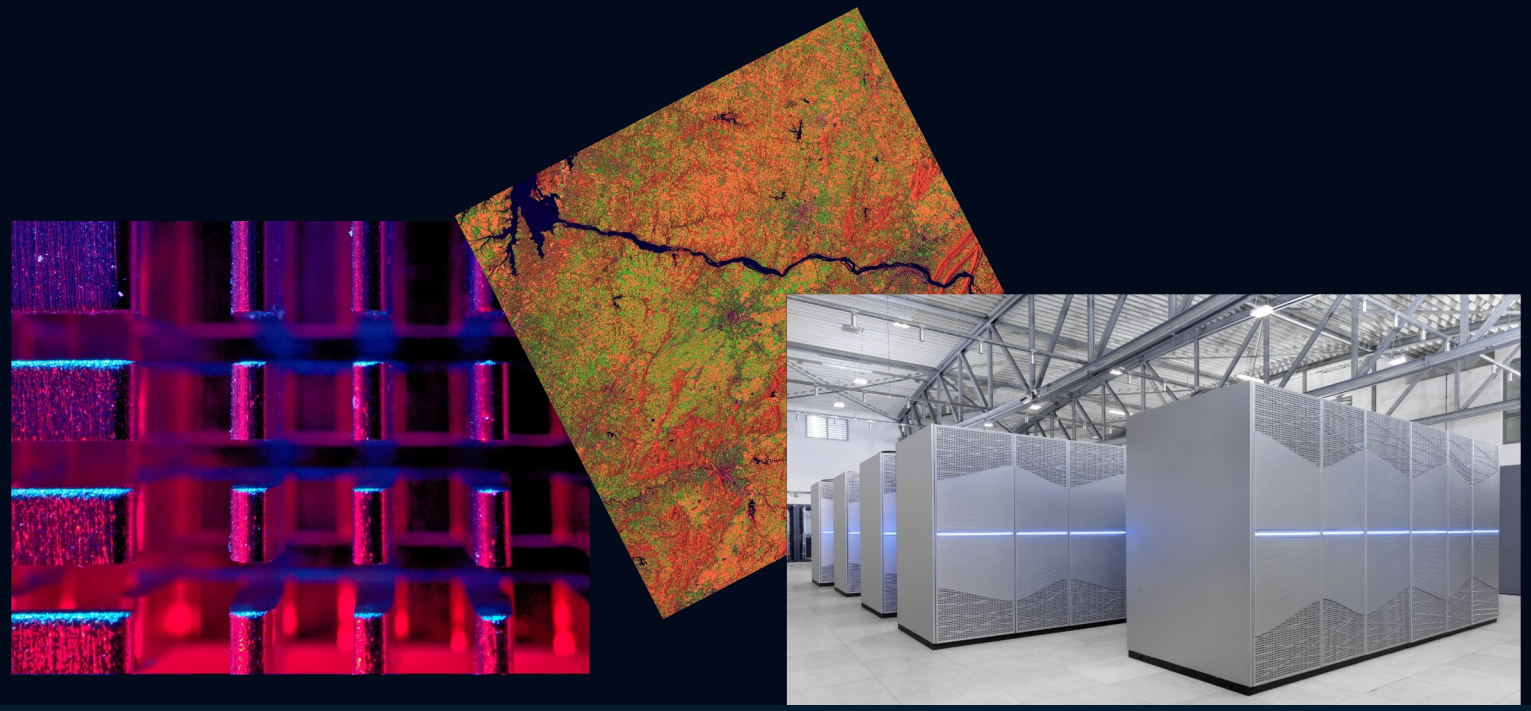


LECTURE 1: MACHINE LEARNING FOR EARTH OBSERVATION POWERED BY SUPERCOMPUTING

GRSS ESI/HDCRS End-to-End Machine Learning with Supercomputing and in the Cloud - Tutorial IGARSS 2023 – 16 July, 2023

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

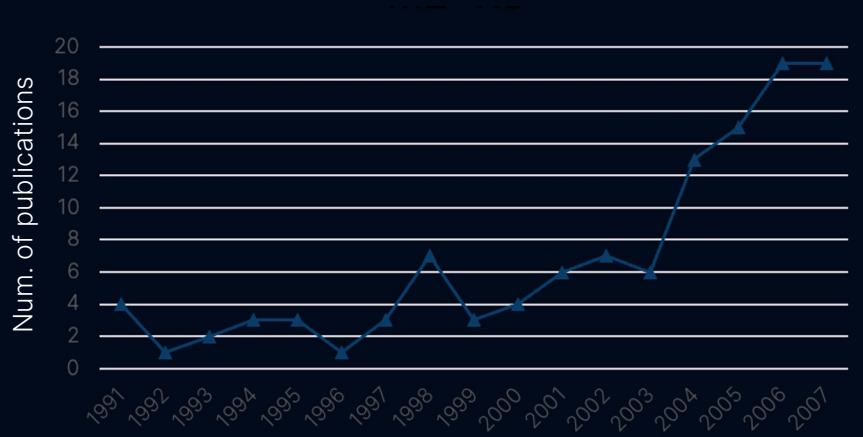
HOW TO EXTRACT KNOWLEDGE IN A TIMELY MANNER?

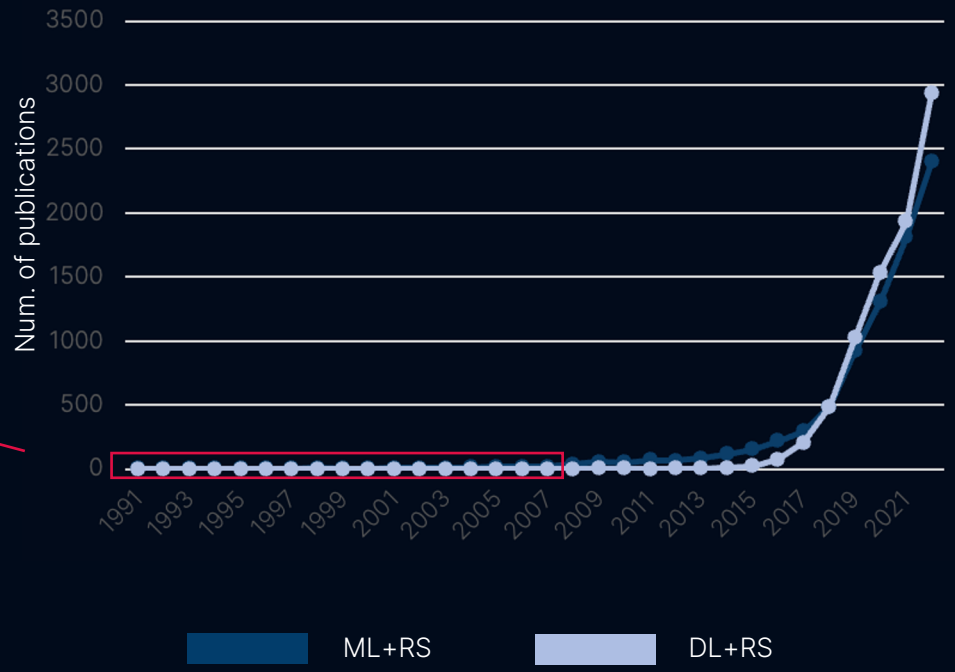
From data acquired by diverse observational systems

HOW TO USE ARTIFICIAL INTELLIGENCE FOR EARTH OBSERVATION?

MACHINE LEARNING AND DEEP LEARNING IN REMOTE SENSING

- Classical ML such as Support Vector Machine (SVM) and Random Forest (RF) since the ‘90s

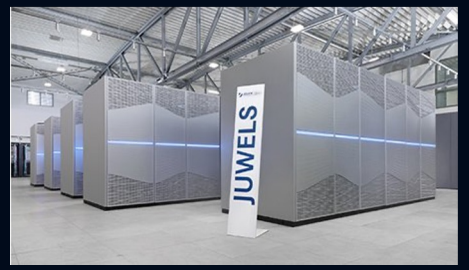




- DL unleashed advances in the last decade
- Figures may differ depending on the source, but the overall trend remains consistent

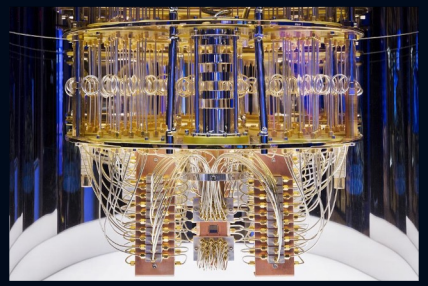
https://www.scopus.com

HOW TO EXPLOIT EMERGING COMPUTING PARADIGMS?



Supercomputing

Specialized Hardware Computing



Quantum Computing

Edge Computing

Cloud computing

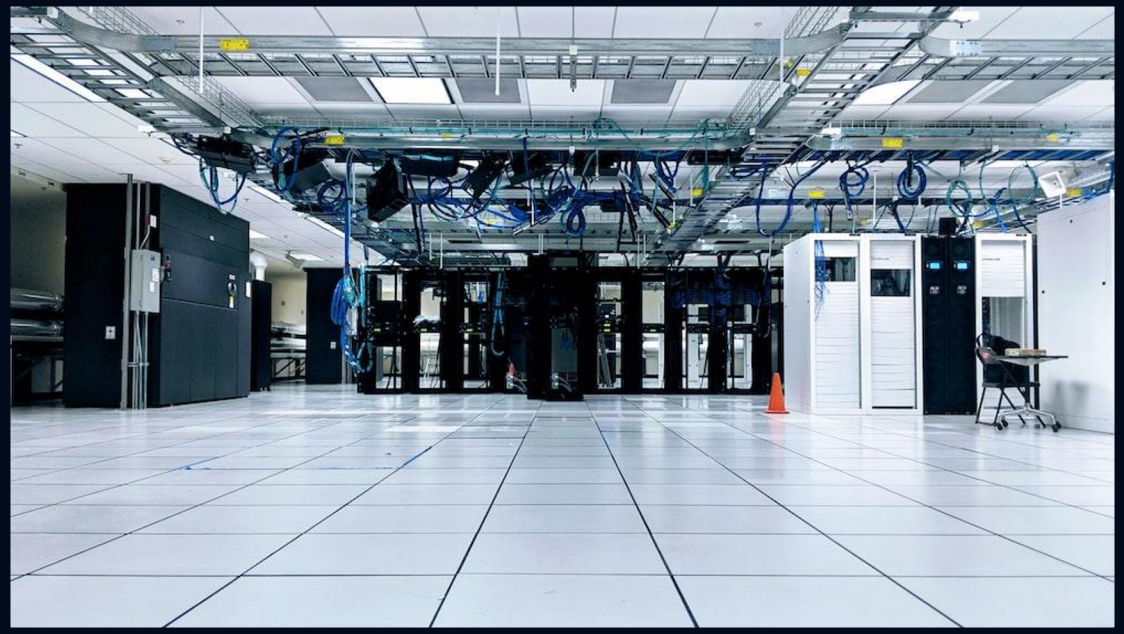
...

Blockchain

HIGH PERFORMANCE COMPUTING

SUPERCOMPUTERS

A high-performance computer for complex calculations that general-purpose computers cannot handle



- High number of processors, vast amounts of memory, and high-speed interconnects
- Optimal for scientific research, weather forecasting, engineering, and more
- Focus on maximum performance and maintainability reliability

DIFFERENCE BETWEEN LAPTOP AND SUPERCOMPUTER

Laptop	Supercomputer
Few cores	Lots of cores
Limited memory	Lots of memory
Slower than supercomputer	Faster than laptop
Personalized Operating System (OS)	Mostly linux based
Owned by you	Owed by public sector and located somewhere
Used by owner	Many users

CURRENT POPULARITY OF SUPERCOMPUTERS

TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings

Industrial Product*

Norman P. Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Cliff Young, Xiang Zhou, Zongwei Zhou, and David Patterson
Google, Mountain View, CA

FORBES > INNOVATION > SUSTAINABILITY

Tesla's Biggest News At AI Day Was The Dojo Supercomputer, Not The Optimus Robot

James Morris Contributor
I write about the rapidly growing world of electric vehicles

Follow

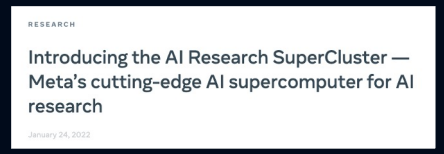
0 Oct 6, 2022, 07:23am EDT

Tech > Science

BABY STEPS Google artificial intelligence supercomputer creates its own 'AI child' that can outperform its human-made rivals

The NASNet system was created by a neural network called AutoML earlier this year

Mark Hodge
Published: 15:22, 5 Dec 2017 | Updated: 11:27, 6 Dec 2017



RESEARCH

Introducing the AI Research SuperCluster — Meta's cutting-edge AI supercomputer for AI research

January 24, 2022

- Big tech companies are announcing their AI supercomputers
- Supercomputing now goes far beyond traditional scientific computing, which was driven by large governments
- The field is currently propelled by major industries building highly specialized supercomputers

N. P. Jouppi, G. Kurian, et al., "TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings", 2023, https://doi.org/10.48550/arXiv.2304.01433

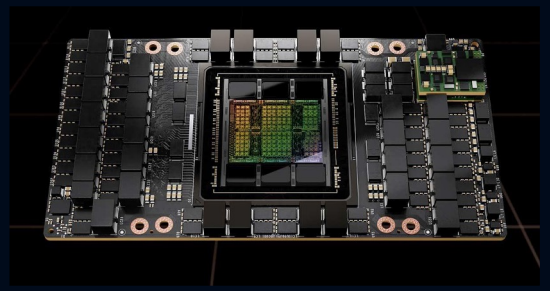
https://www.forbes.com/sites/jamesmorris/2022/10/06/teslas-biggest-news-at-ai-day-was-the-dojo-supercomputer-not-the-optimus-robot/?sh=22ba4ab780bd

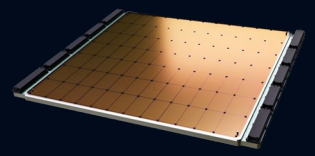
https://ai.facebook.com/blog/ai-rsc/

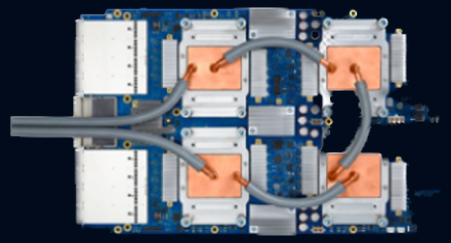
https://www.thesun.co.uk/tech/5072741/google-nasnet-ai-child-reinforcement-learning/

Torsten Hoefler, "Efficient AI: From supercomputers to smartphones", Scalable Parallel Computing Lab @ ETH Zurich, https://youtu.be/xxwT45ljG4o

TRAINING OF DEEP LEARNING MODELS REQUIRES ACCELERATORS







- GPUs (currently NVIDIA dominant), TPUs (Google)
- GPUs: generic deep learning hardware (parallelizing matrix/tensor operations via vectorization)
- Spezialized hardware, eg. in-memory computing chips, Graphcore IPU: Colossus MK2, Cerebras Wafer Scale Engine 2 (850k cores)

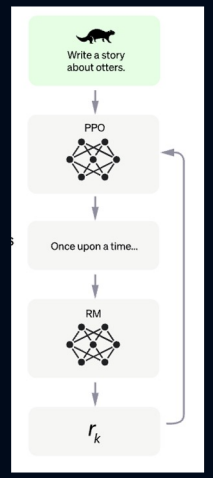
https://www.nvidia.com/en-us/data-center/technologies/hopper-architecture/

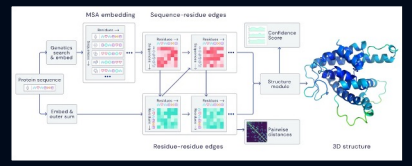
https://cloud.google.com/tpu

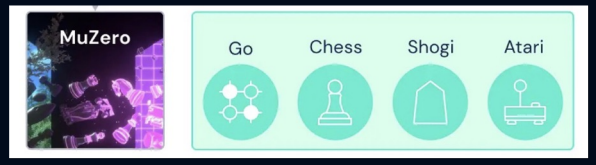
https://www.graphcore.ai/products/ipu

https://www.cerebras.net/product-chip/

MOST BREAKTHROUGHS REQUIRE HEAVY COMPUTE POWER, USING MANY ACCELERATORS SIMULTANEOUSLY







OpenAI

DeepMind

- GPT-3: natural language generation, language understanding
- CLIP, DALL-E 2, Stable Diffusion: image understanding and image generation
- AlphaFold 2: protein structure prediction
- AlphaZero, MuZero: learning control in highly dimensional state-action spaces

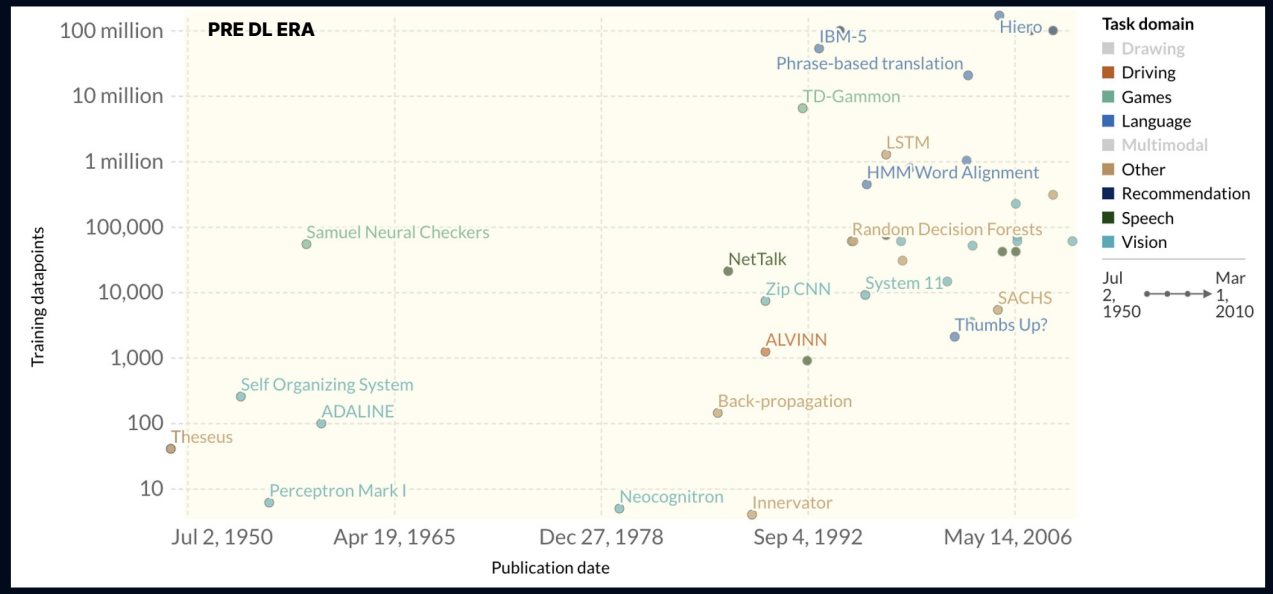
https://openai.com/blog/chatgpt

AlphaFold: a solution to a 50-year-old grand challenge in biology, https://www.deepmind.com/blog/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology

MuZero: Mastering Go, chess, shogi and Atari without rules, https://www.deepmind.com/blog/muzero-mastering-go-chess-shogi-and-atari-without-rules

Jenia Jitsev, Towards Scalable Deep Learning, Scalable Learning & Multi-Purpose AI Lab, Helmholtz AI, LAION @ JSC

PRE DEEP LEARNING ERA (1950-2010)

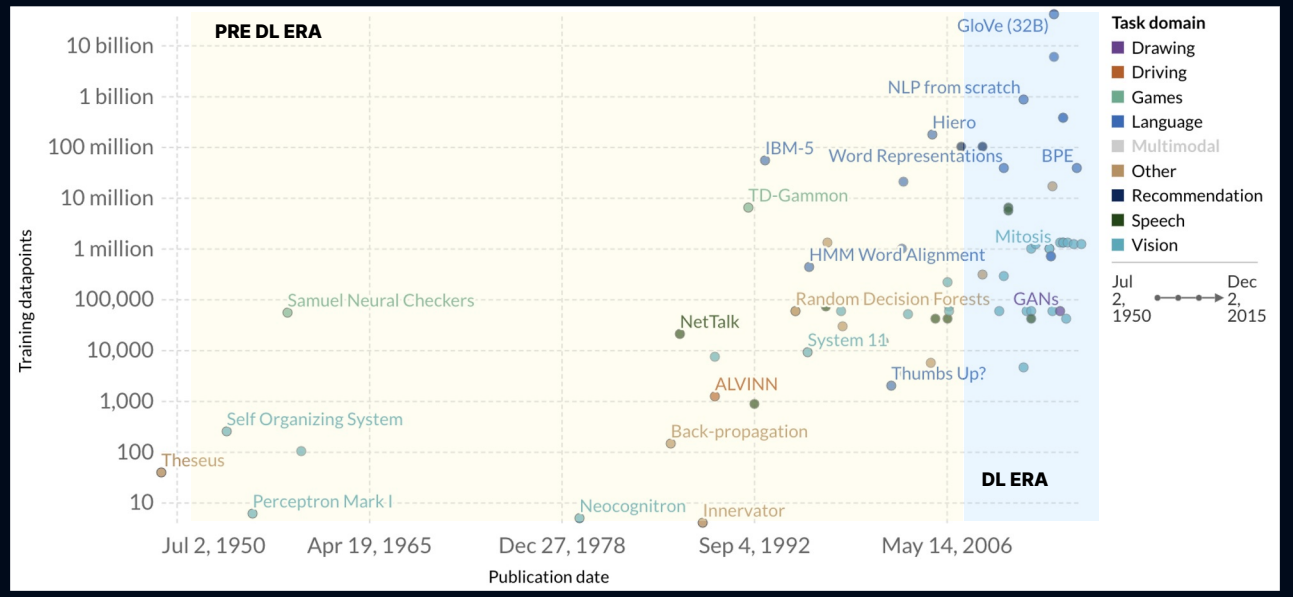


C. Giattino, et al., Artificial Intelligence, https://ourworldindata.org/artificial-intelligence

J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn and P. Villalobos, "Compute Trends Across Three Eras of Machine Learning," 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2022, https://doi.org/10.1109/IJCNN55064.2022.9891914

- Training compute doubled every 17 to 29 months
- This increase is roughly in line with Moore's Law

DEEP LEARNING ERA AROUND 2010

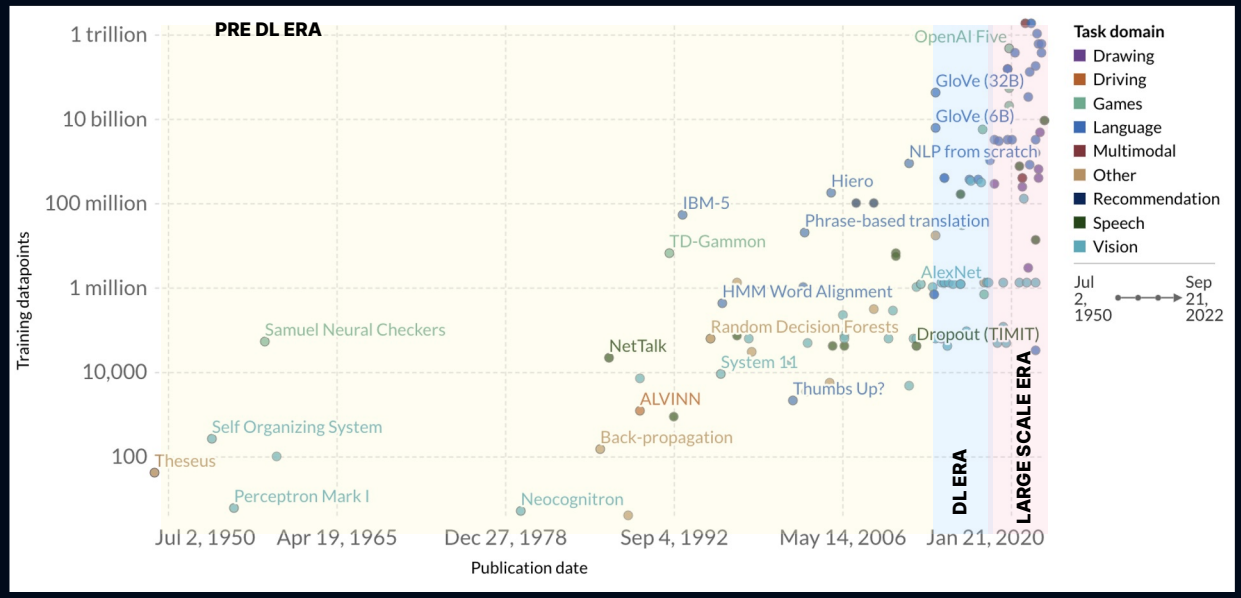


C. Giattino, et al., Artificial Intelligence, https://ourworldindata.org/artificial-intelligence

J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn and P. Villalobos, "Compute Trends Across Three Eras of Machine Learning," 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2022, https://doi.org/10.1109/IJCNN55064.2022.9891914

- The trend accelerated along the with the popularity of deep learning
- From around 2010, compute doubles every 4 to 9 months

LARGE-SCALE ERA AROUND 2015



C. Giattino, et al., Artificial Intelligence, https://ourworldindata.org/artificial-intelligence

J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn and P. Villalobos, "Compute Trends Across Three Eras of Machine Learning," 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2022, https://doi.org/10.1109/IJCNN55064.2022.9891914

D. Silver et al., "Mastering the game of Go without human knowledge," Nature, vol. 550, pp. 354–359, 2017, https://doi.org/10.1038/nature24270

- In late 2015, a new trend of large-scale models emerged
- Computational capacity significantly higher than that of other models published in the same year (e.g., release of AlphaGo)
- This growth trend is slower than the overall DL trend, with a doubling time of roughly every 8 to 17 months

JUWELS

Exascale Pathfinder

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

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

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)

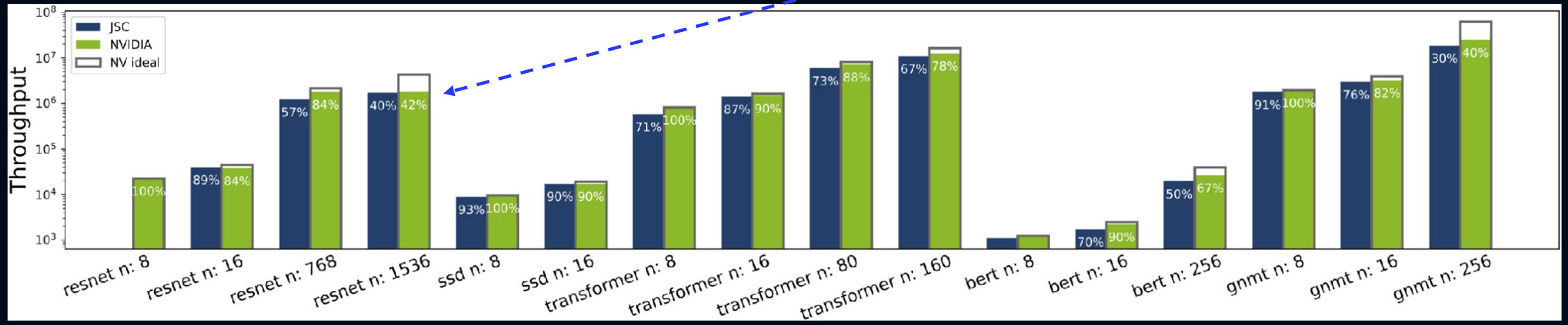
JUWELS BOOSTER – A "DUAL USE" SYSTEM

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

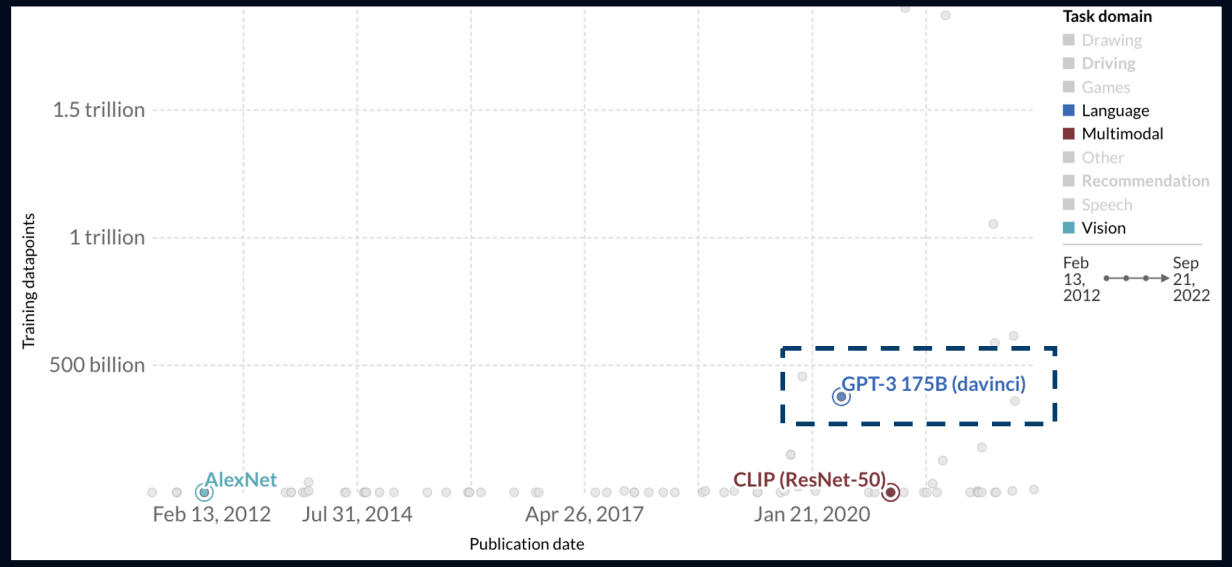
Example

- 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.

GPT-3 MODEL (175 BILLION WEIGHT PARAMETERS)



- AlexNet, winner ILSRVC 2012 – 60M
- ResNet-50, winner ILSRVC 2015 – 25M
- CLIP ViT L/14, multi-modal learning, 2021 - 600M

C. Giattino, et al., Artificial Intelligence, https://ourworldindata.org/artificial-intelligence ://developer.nvidia.com/blog/scaling-language-model-training-to-a-trillion-parameters-using-megatron/

Jenia Jitsev, Towards Scalable Deep Learning, Scalable Learning & Multi-Purpose AI Lab, Helmholtz AI, LAION @ JSC

TIME REQUIRED FOR A FULL TRAINING OF GPT-3 MODEL

- ≈ 355 years with one Nvidia Volta 100 GPU
- ≈ 90 years with one Nvidia Ampere 100 GPU
- ≈ 16 days if scaled well with 2,000 A100 GPUs on JUWELS Booster

Jülich Wizard for European Leadership Science (JUWELS), https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/juwels

Jenia Jitsev, Towards Scalable Deep Learning, Scalable Learning & Multi-Purpose AI Lab, Helmholtz AI, LAION @ JSC

TIME REQUIRED FOR A FULL TRAINING OF GPT-4 MODEL

- ≈ 5500 years with one Nvidia Ampere 100 GPU
- ≈ 1200 years with one Nvidia Hopper 100 GPU
- ≈ 900 days if scaled well with 2,000 A100 GPUs on JUWELS Booster

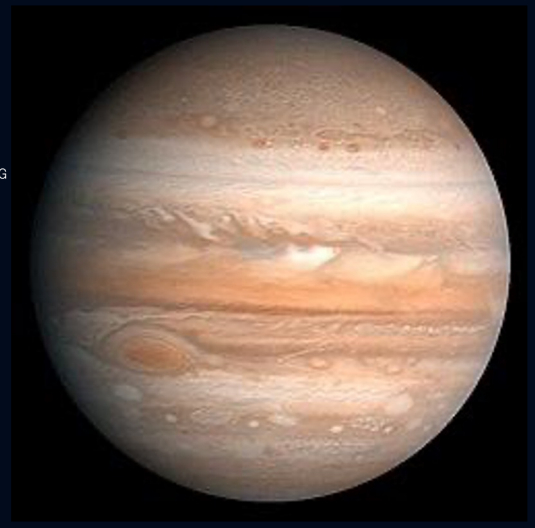
Jülich Wizard for European Leadership Science (JUWELS), https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/juwels

JU PIONEER FOR INNOVATIVE AND TRANSFORMATIVE EXASCALE RESEARCH (JUPITER)

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

Mars
(JUWELS)

Jupiter



Mars: d= 6,800 Km
Jupiter: d= 143,00 Km

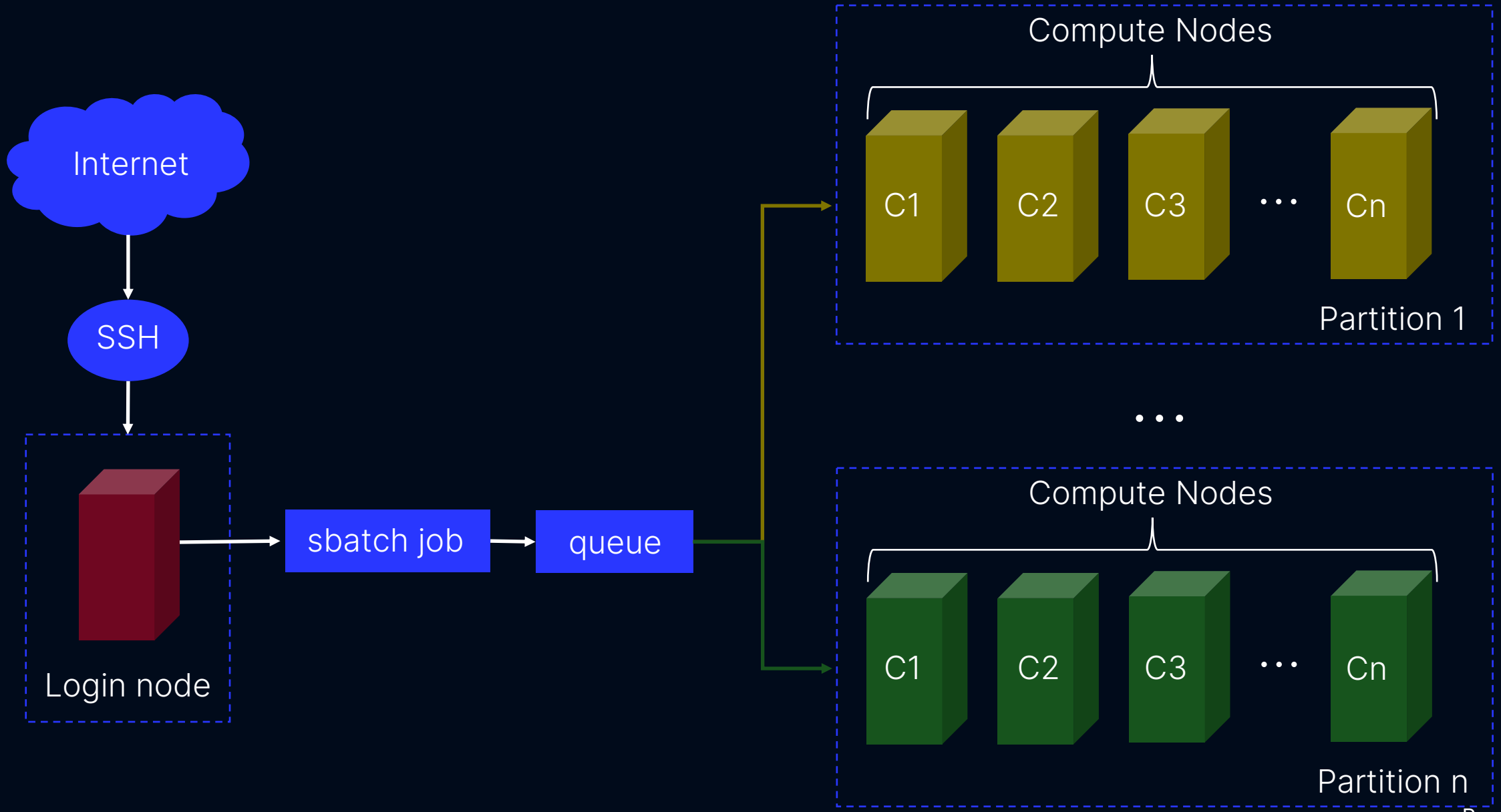
JUWELS: 85 PF/s peak
JUPITER: 1,300 PF/s peak

Thomas Lippert, "Jupiter Ascending, Driving the Forefront of Europe's HPC Effort", EuroHPC Summit 2023, https://www.eurohpcsummit.eu/

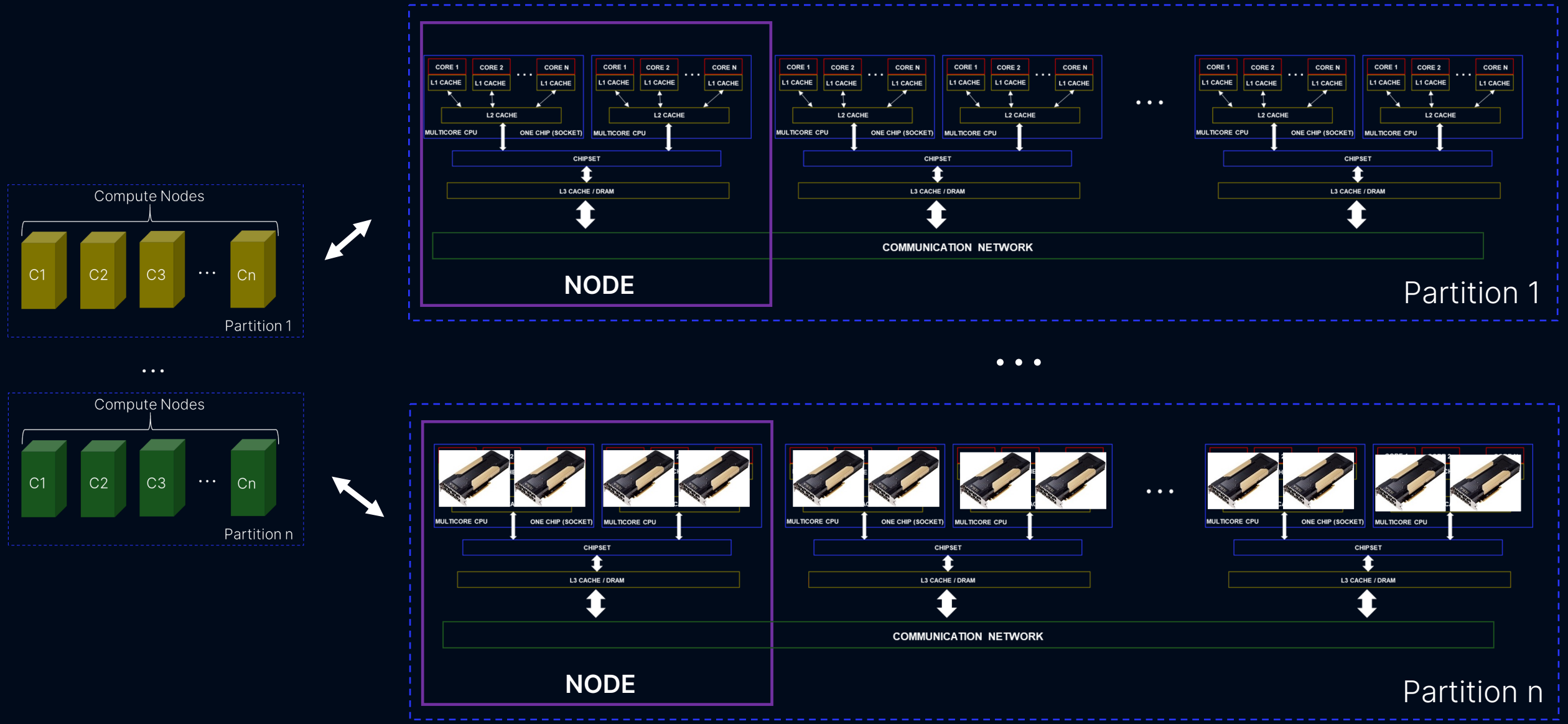
JUPITER | The Arrival of Exascale in Europe, https://www.fz-juelich.de/en/ias/jsc/jupiter

- The Forschungszentrum Jülich in Germany will house Europe's first exascale computer, named JUPITER
- This machine is set to surpass the threshold of one quintillion (a “1” followed by 18 zeros) calculations per second
- JUWELS 20 = JUPITER (when using the diameters of planets as a measurement scale)

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)

- 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

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

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

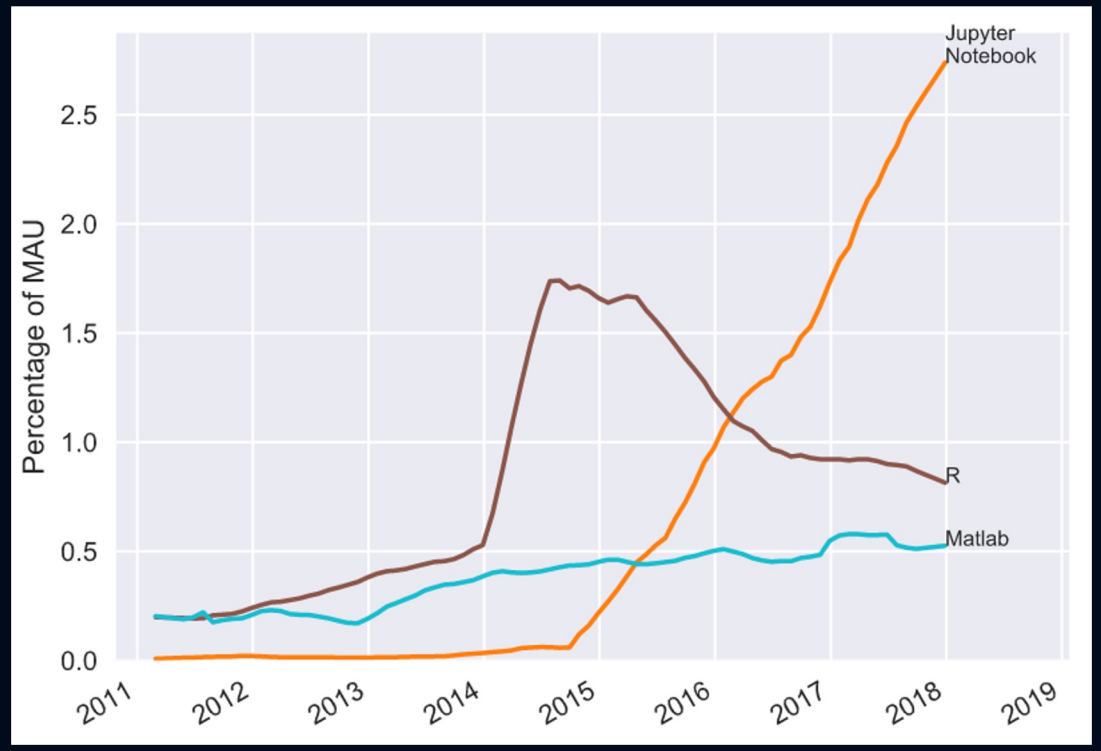
UNICORE

UNICORE

Uniform Interface to Computing Resources

RISE OF JUPYTER'S POPULARITY

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

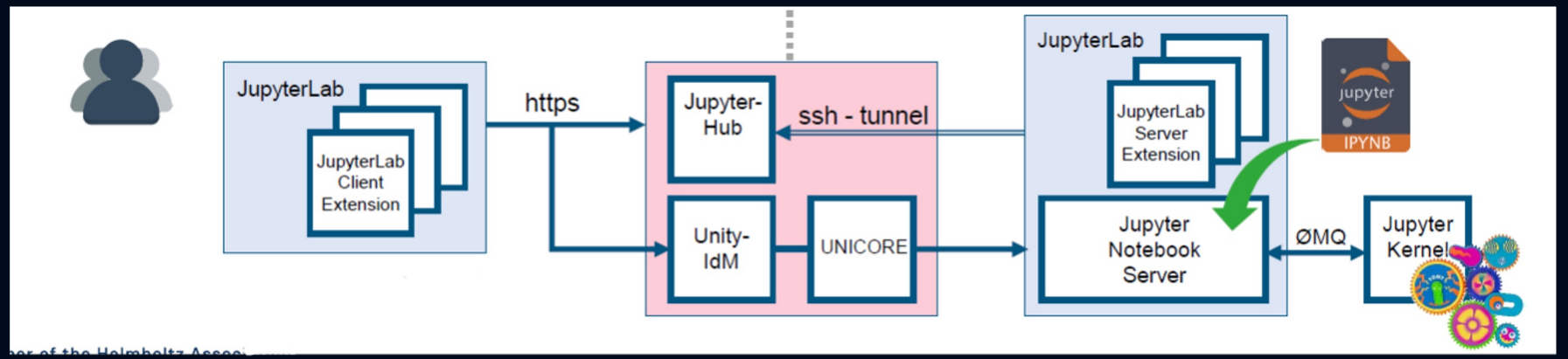


https://www.benfrederickson.com/ranking-programming-languages-by-github-users/

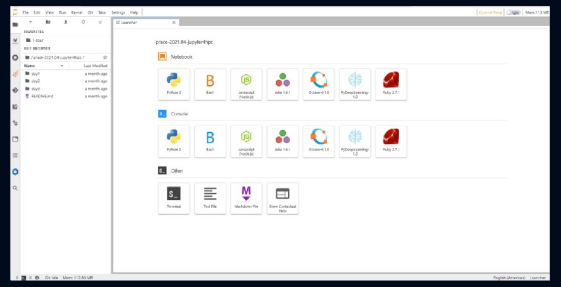
Jupyter Notebooks have seen significant and steady growth over the last years

Python is pushing this trend

JUPYTER-JSC WEB SERVICE



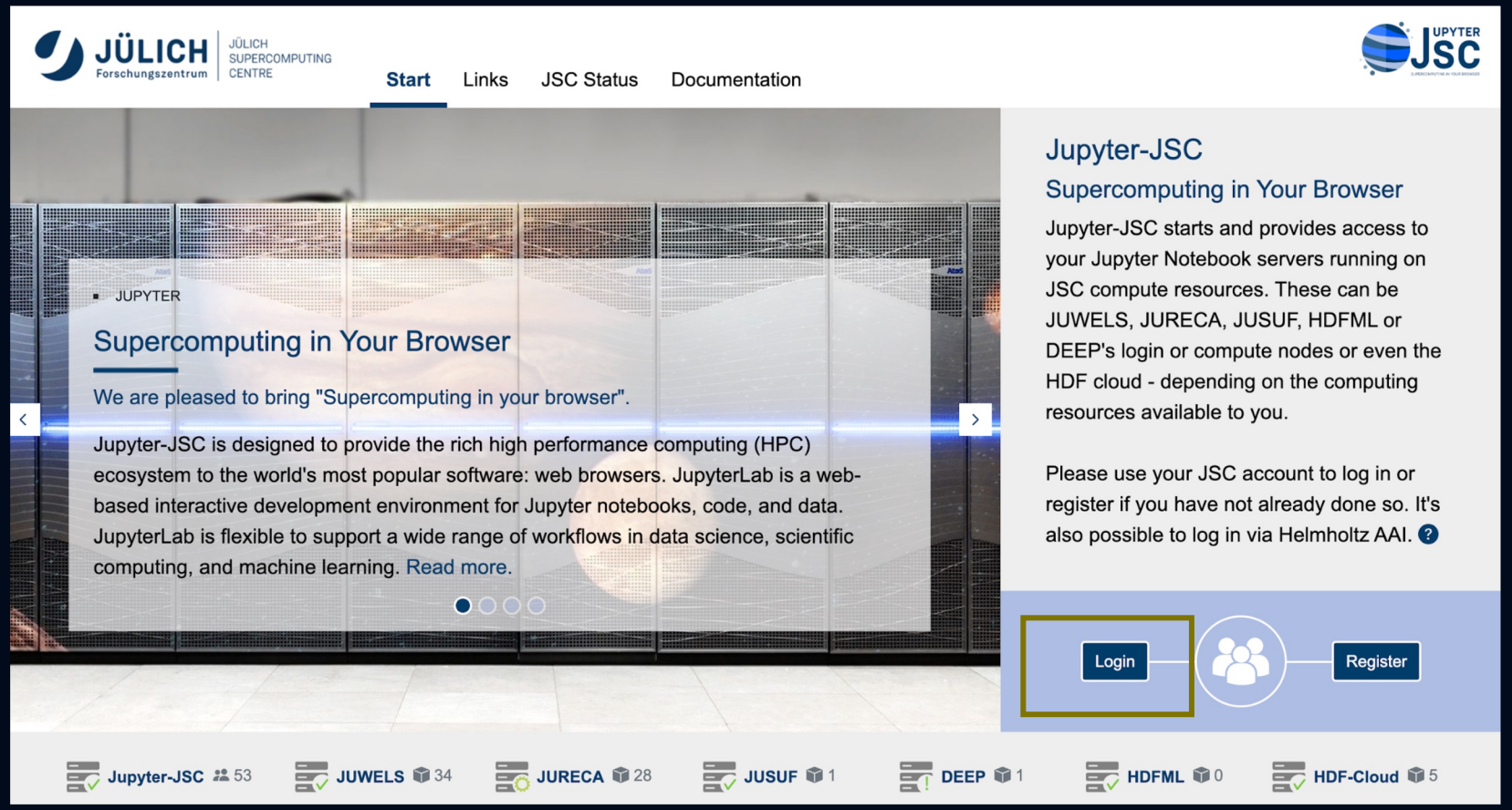
Browser



HPC system

ACCESS JUPYTER-JSC

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



JURECA DC – HPC SYSTEM

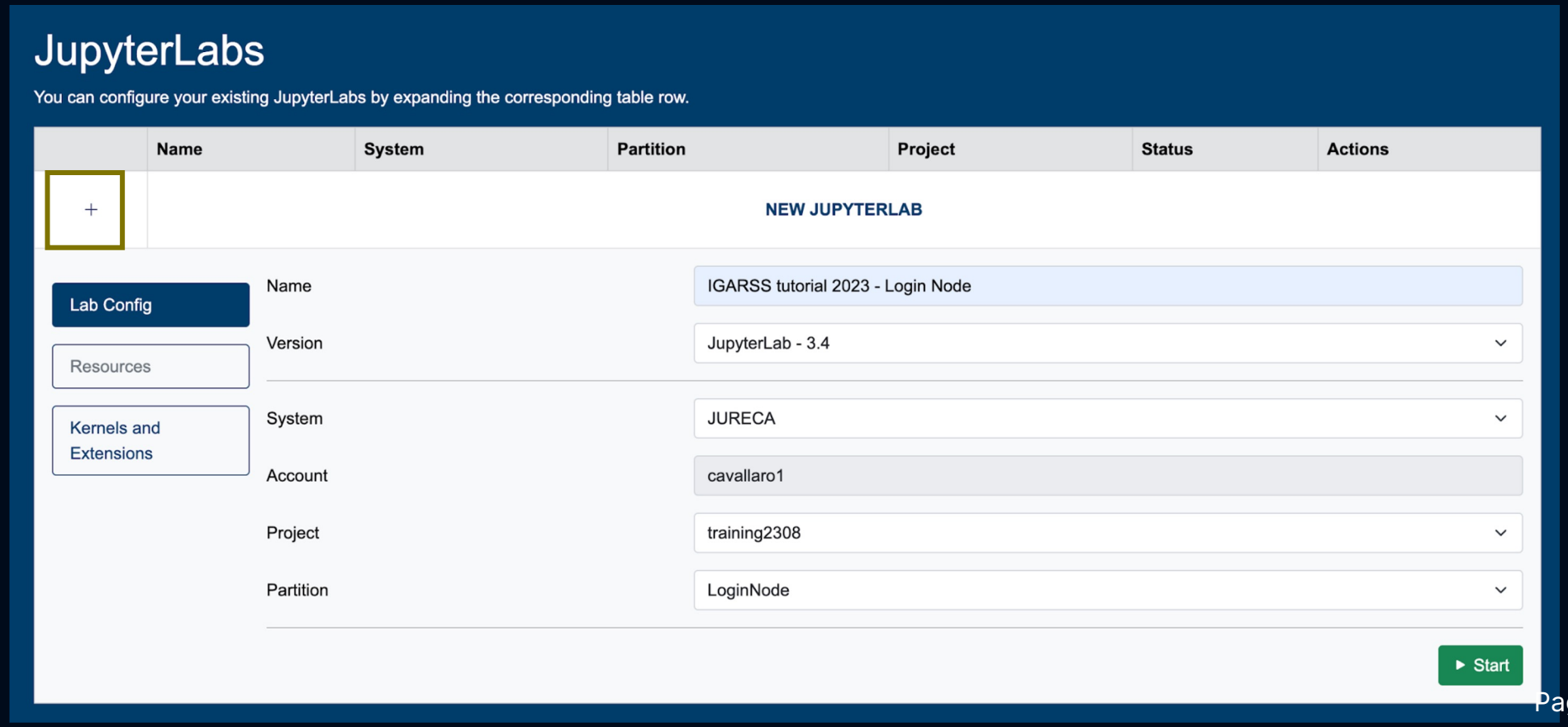
Hardware Configuration of the JURECA DC Module (Phase 2: as of May 2021)

- **480 standard compute nodes**
  - 2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz
  - 512 (16× 32) GB DDR4, 3200 MHz
  - InfiniBand HDR100 (NVIDIA Mellanox Connect-X6)
  - diskless
- **96 large-memory compute nodes**
  - 2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz
  - 1024 (16× 64) GB DDR4, 3200 MHz
  - InfiniBand HDR100 (NVIDIA Mellanox Connect-X6)
  - diskless
- **192 accelerated compute nodes**
  - 2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz
  - 512 (16× 32) GB DDR4, 3200 MHz
  - 4× NVIDIA A100 GPU, 4× 40 GB HBM2e
  - 2× InfiniBand HDR (NVIDIA Mellanox Connect-X6)
  - diskless
- **12 login nodes**
  - 2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz
  - 1024 (16× 64) GB DDR4, 3200 MHz
  - 2× NVIDIA Quadro RTX8000
  - InfiniBand HDR100 (NVIDIA Mellanox Connect-X6)
  - 100 Gigabit Ethernet external connection

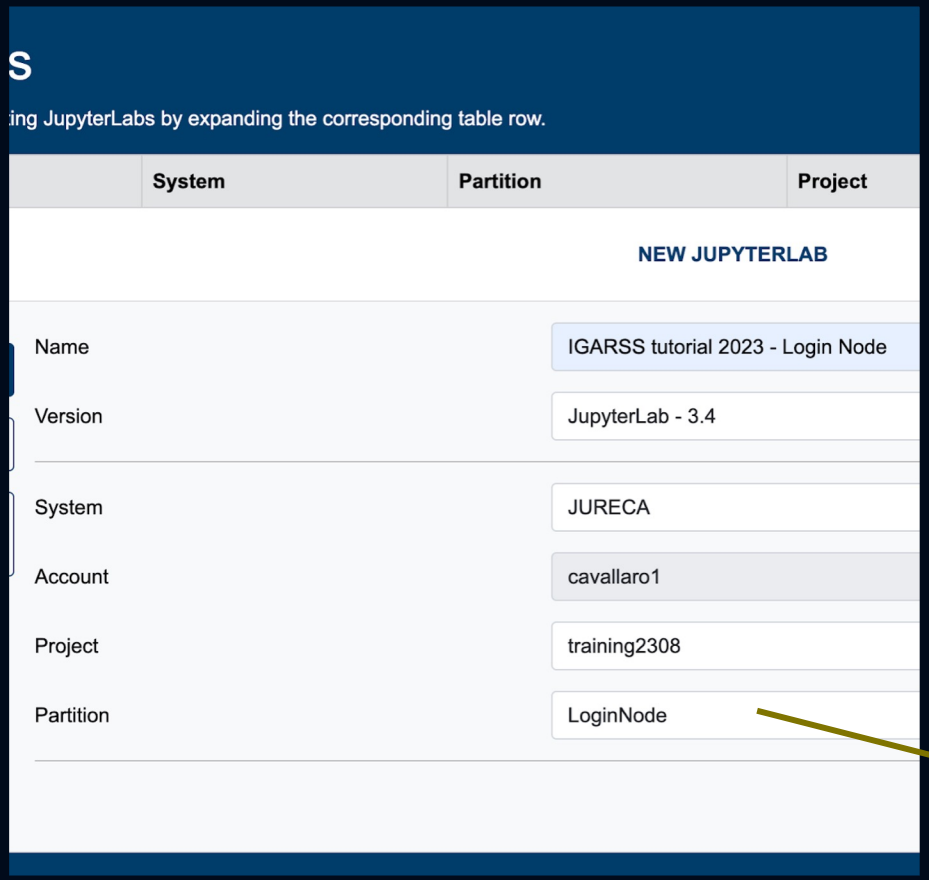
https://apps.fz-juelich.de/jsc/hps/jureca/configuration.html

CREATE A NEW JUPYTERLAB

- Partition: login node



CREATE A NEW JUPYTERLAB



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

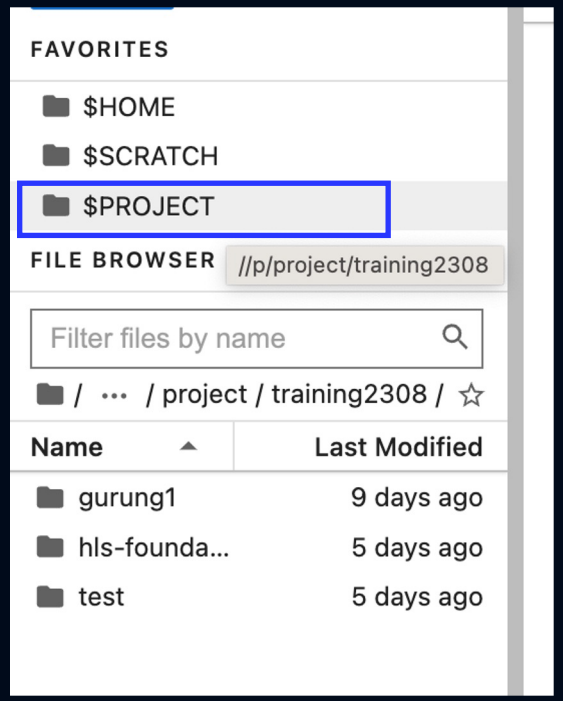
12 LOGIN NODES

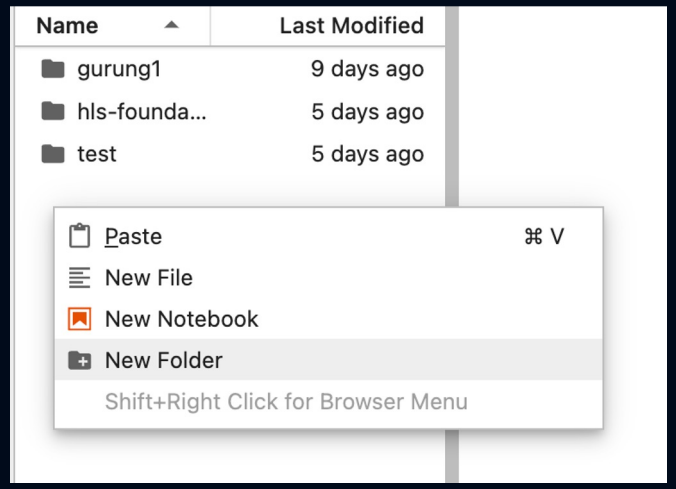
Partition 3

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

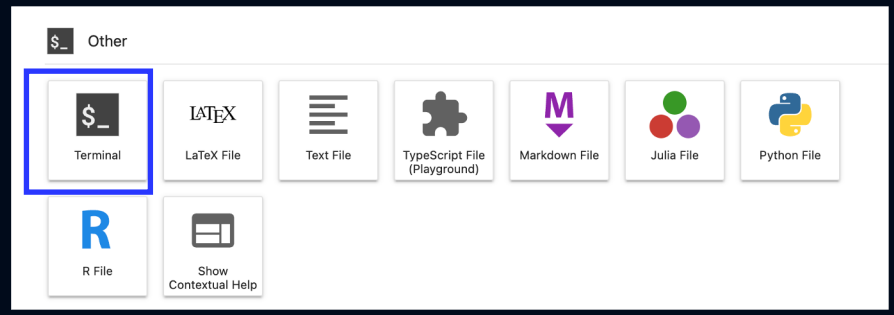
CREATE YOUR OWN FOLDER IN THE PROJECT

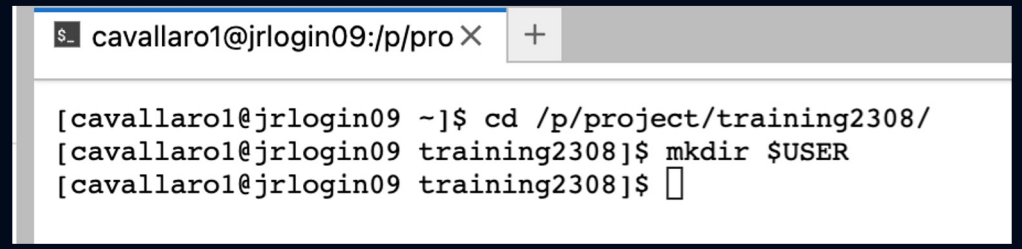
METHOD 1



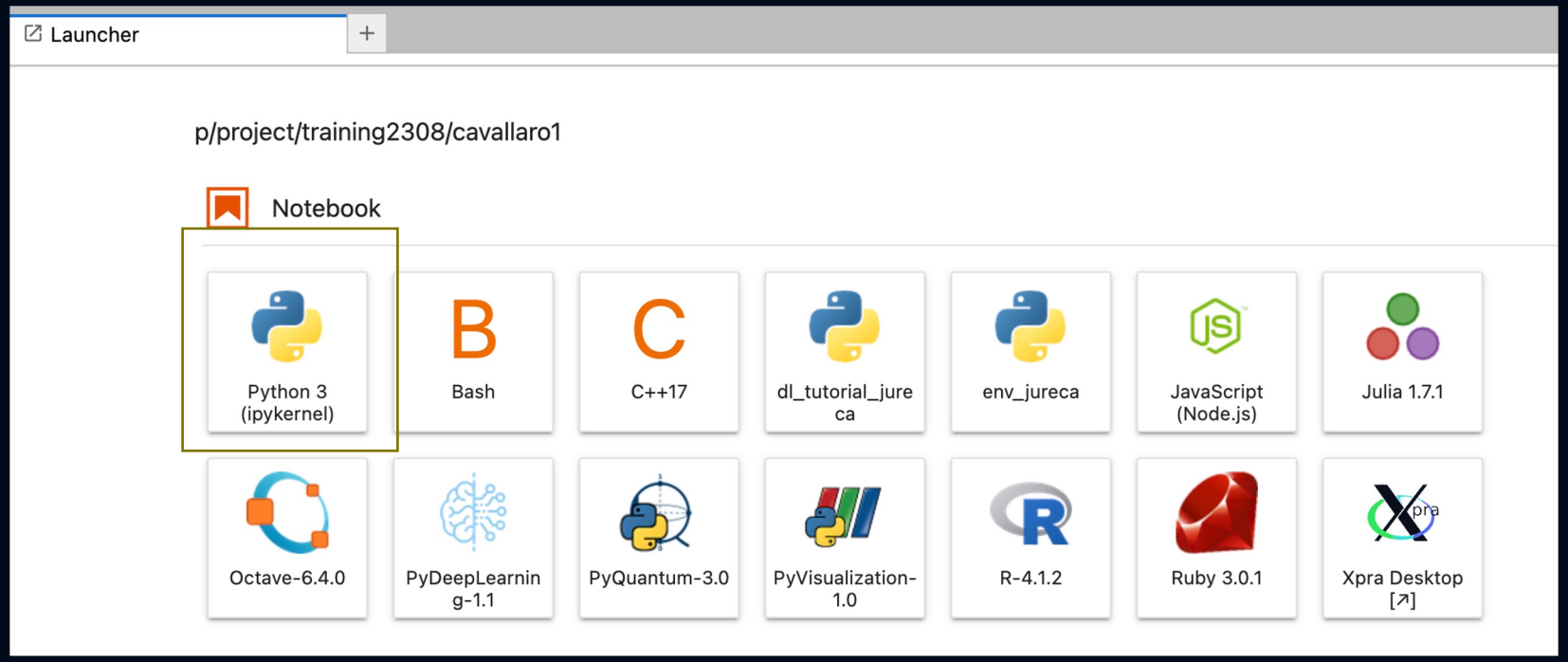


METHOD 2

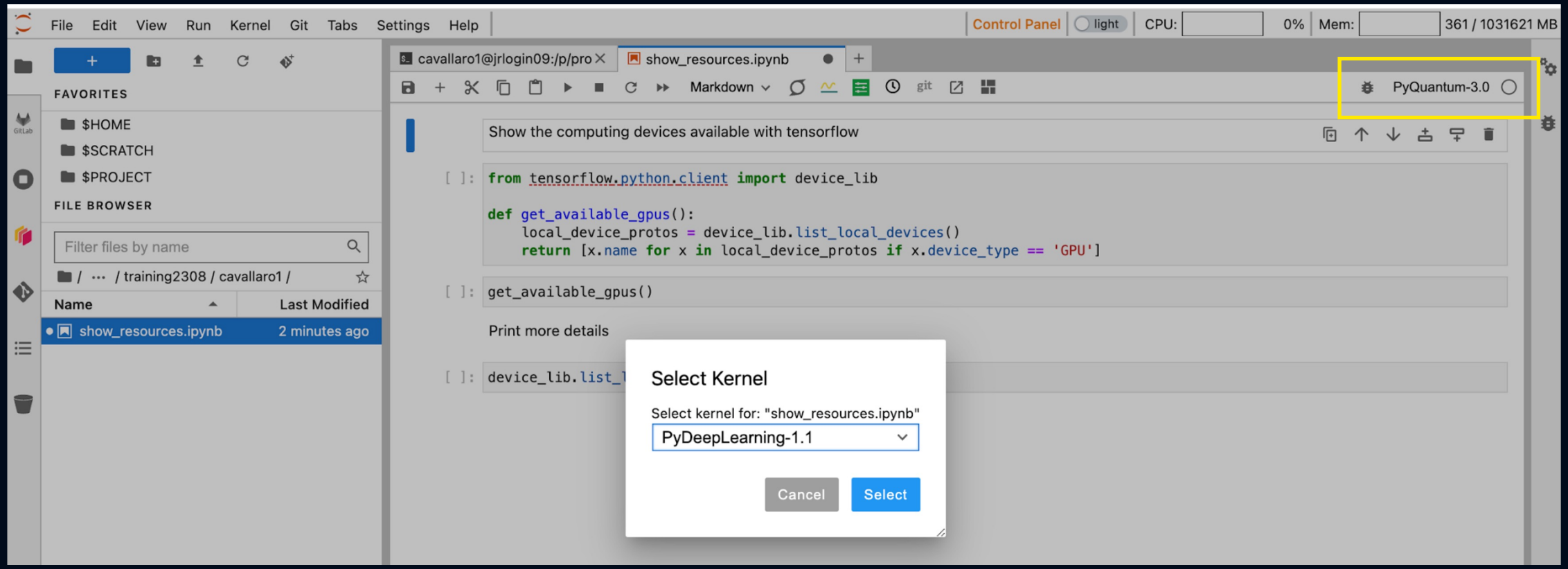




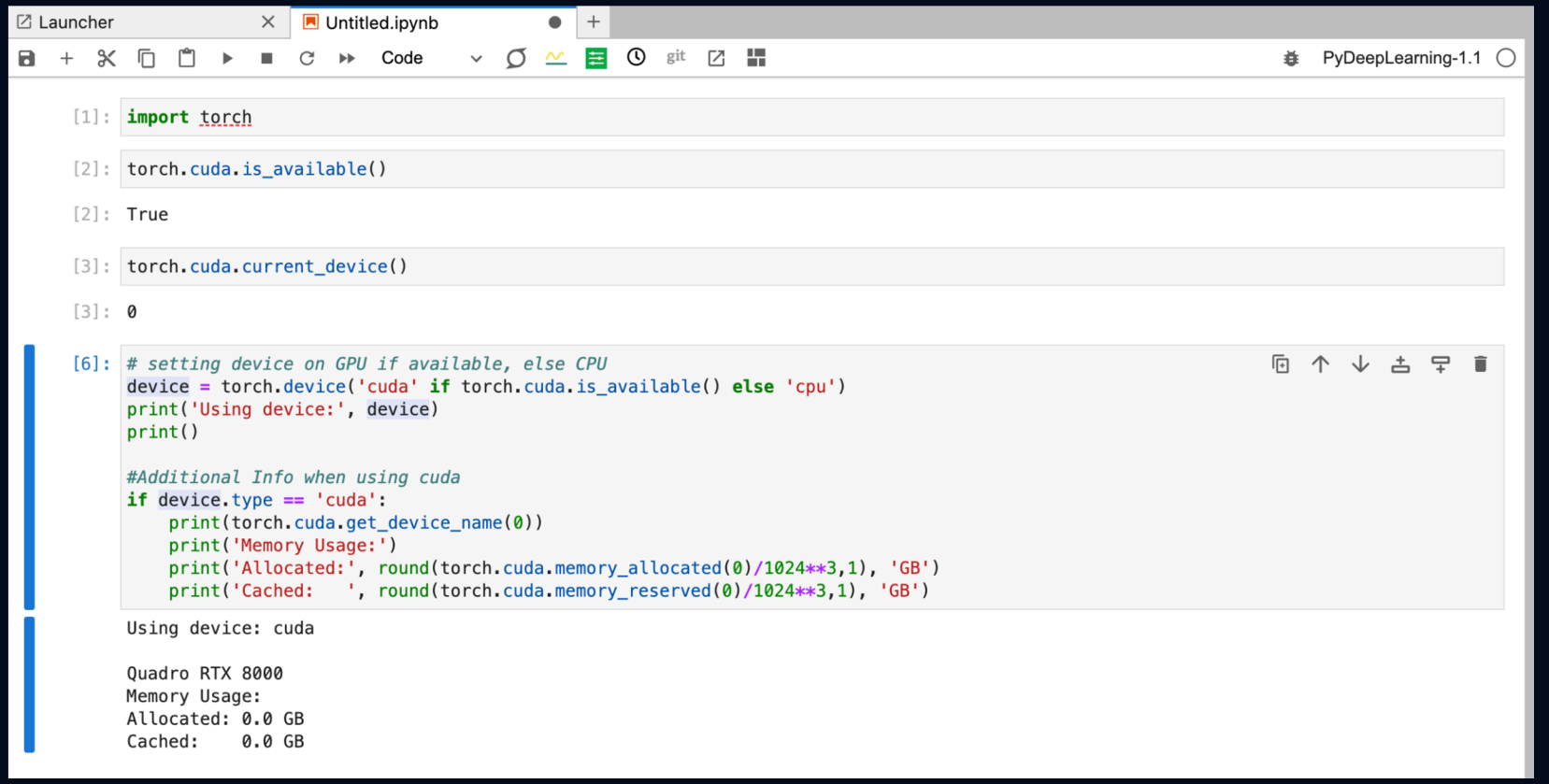
LAUNCH A NEW NOTEBOOK



SELECT THE KERNEL



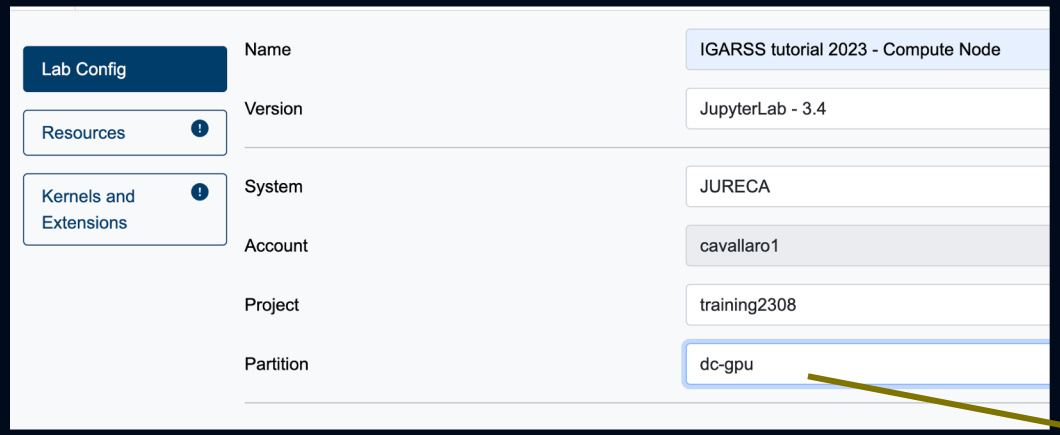
CHECK COMPUTING RESOURCES

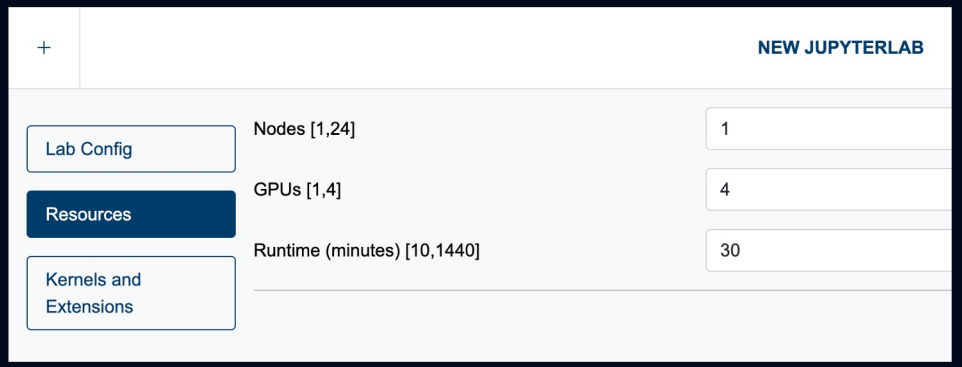


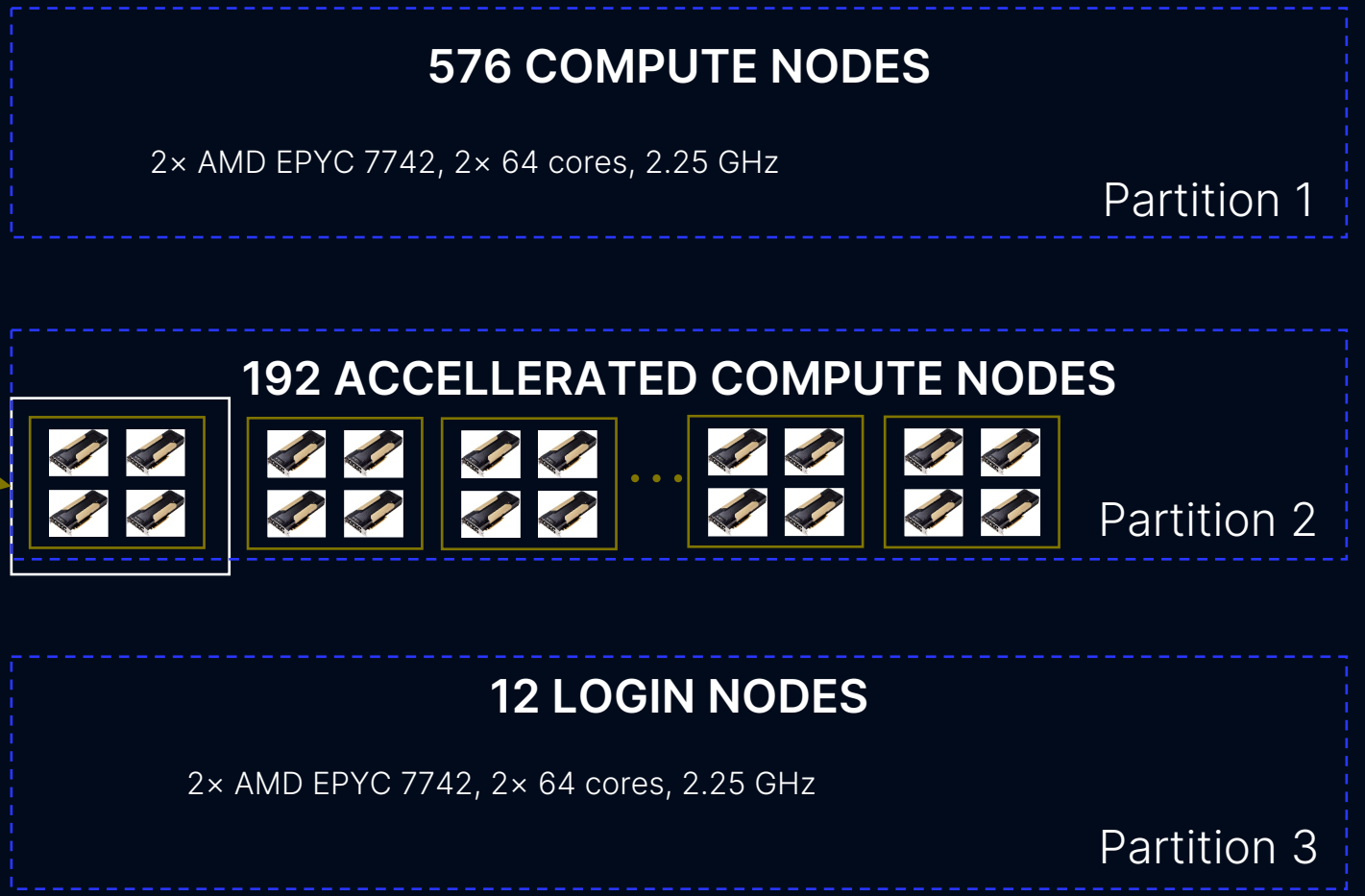
Login nodes are used for job submission, code compilation, and file management, not for computation.

They serve as a gateway to HPC resources and aren't designed to handle heavy computational loads.

CREATE A NEW JUPYTERLAB

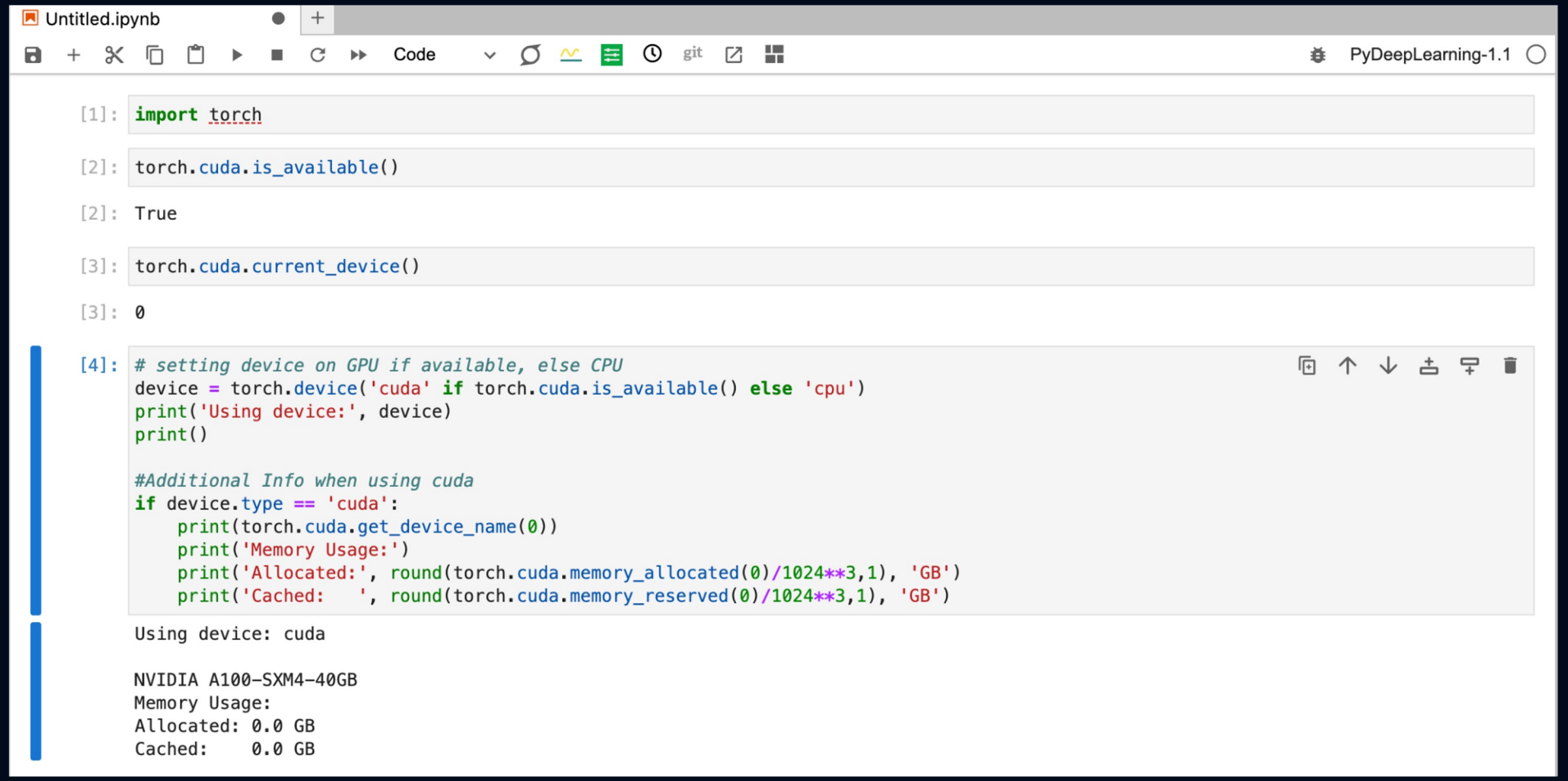




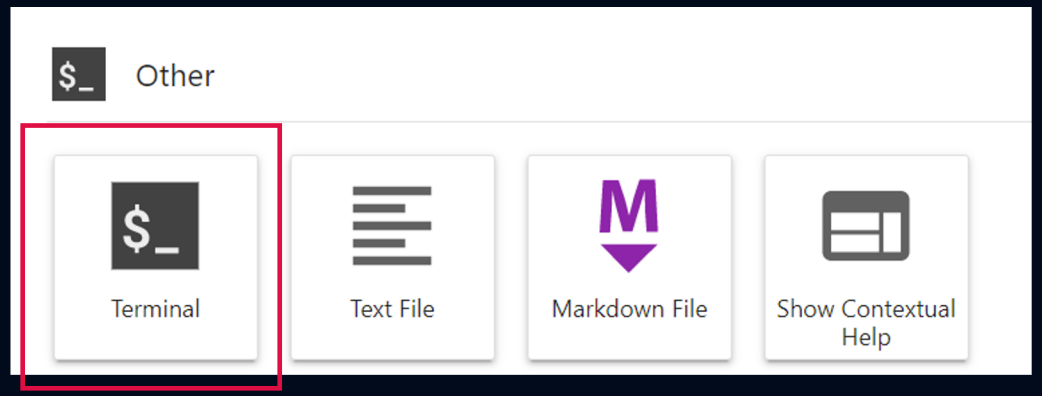


https://apps.fz-juelich.de/jsc/hps/jureca/configuration.html

CHECK COMPUTING RESOURCES



CHECK COMPUTING RESOURCES IN THE TERMINAL



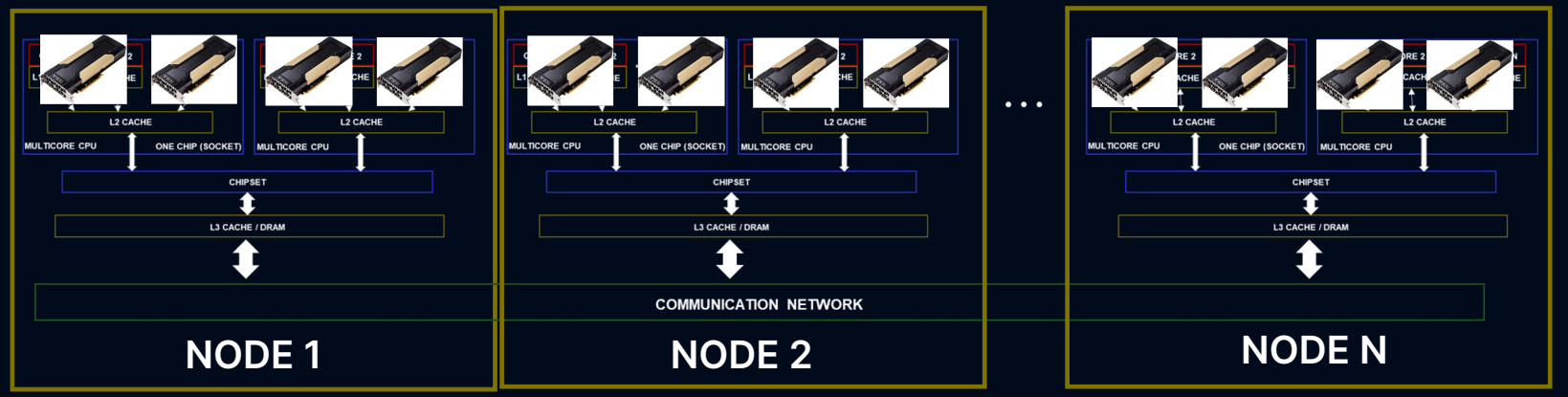

```
cavallaro1@jrc0226:/p/proj
[cavallaro1@jrc0226 cavallaro1]$ nvidia-smi
Fri Jul  7 10:50:54 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.105.17   Driver Version: 525.105.17   CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| 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   45C    P0    61W / 400W |      3MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA A100-SXM...  On   | 00000000:44:00.0 Off |                    0 |
| N/A   43C    P0    58W / 400W |      3MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   2  NVIDIA A100-SXM...  On   | 00000000:84:00.0 Off |                    0 |
| N/A   44C    P0    59W / 400W |      3MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   3  NVIDIA A100-SXM...  On   | 00000000:C4:00.0 Off |                    0 |
| N/A   44C    P0    58W / 400W |      3MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
[cavallaro1@jrc0226 cavallaro1]$
```

LATER IN THIS TUTORIAL

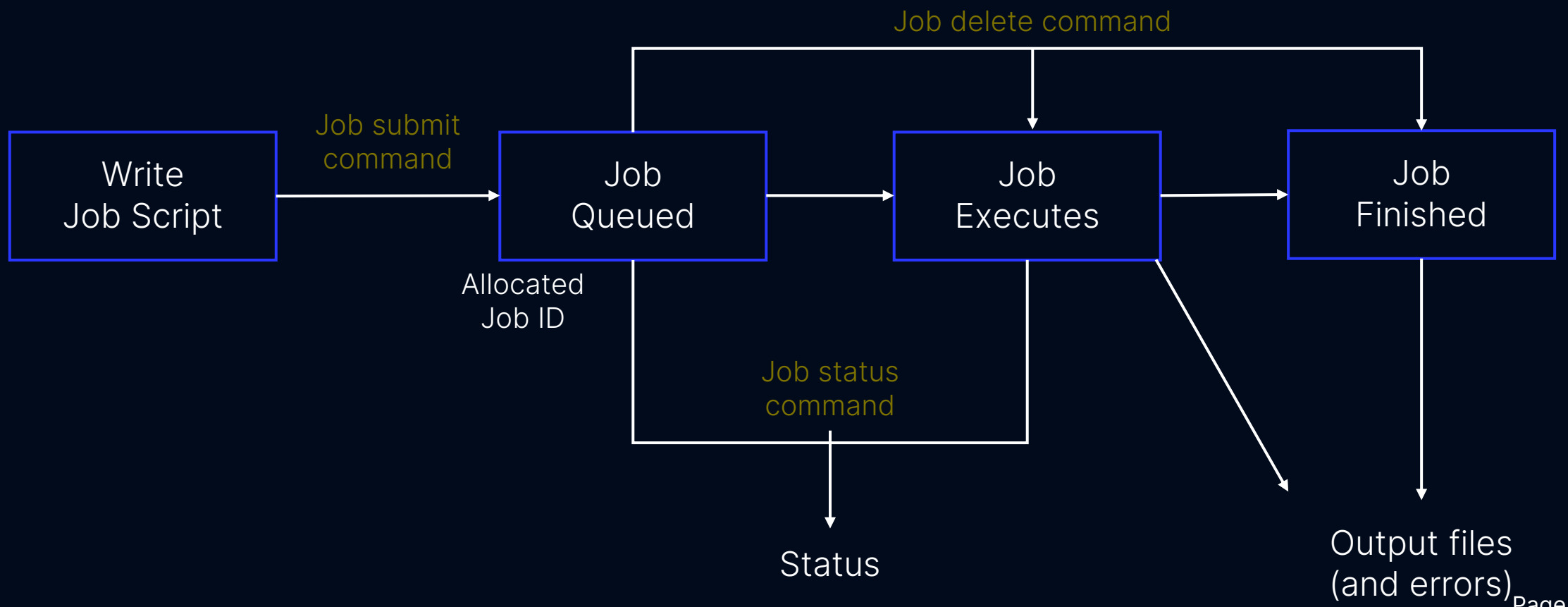
Hands-on - Distributed Deep Learning

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

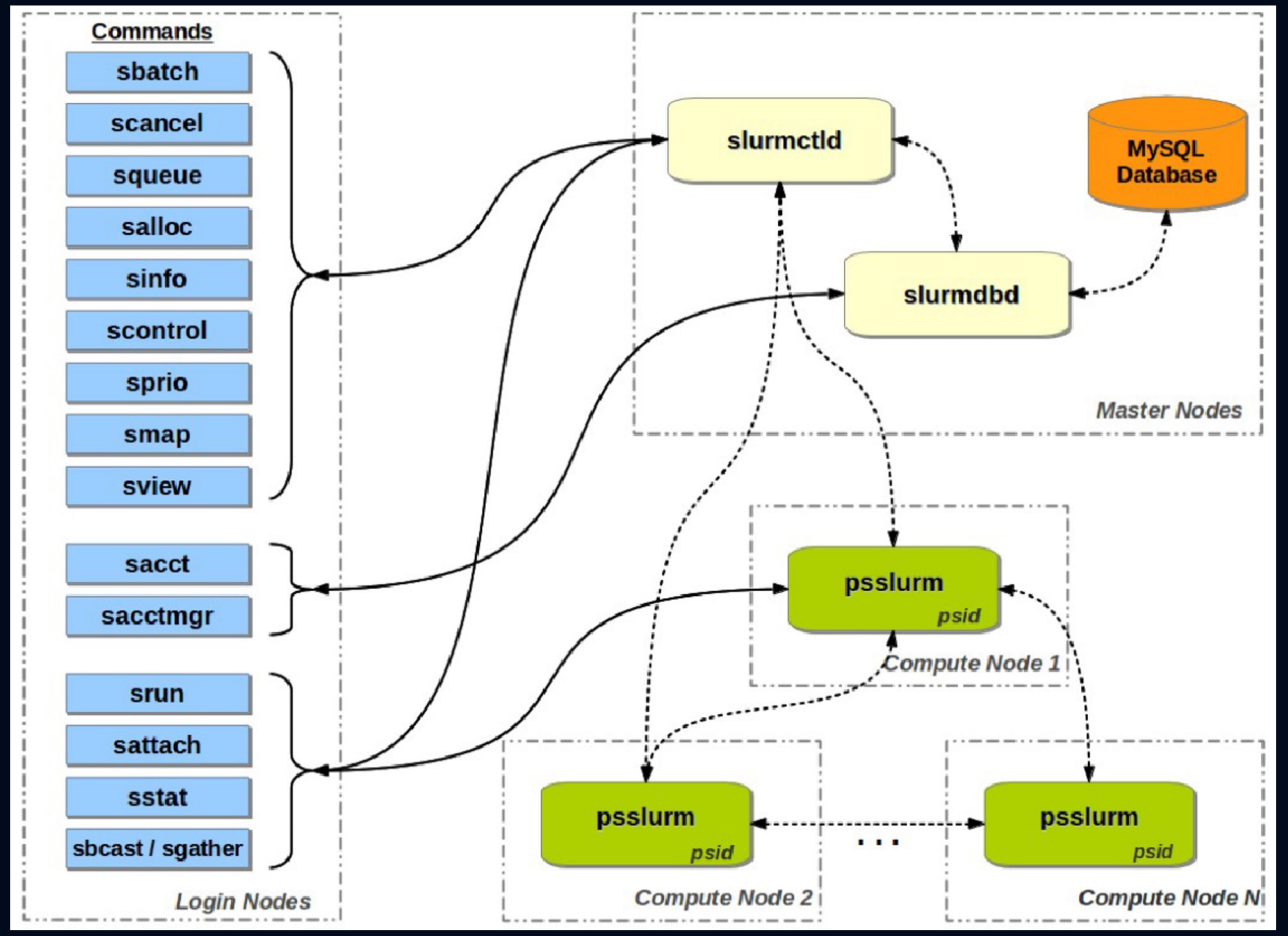


USING THE SUPERCOMPUTER MEANS SUBMITTING A JOB TO A BATCH SYSTEM

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



THE SLURM BATCH SYSTEM



NO NODE-SHARING

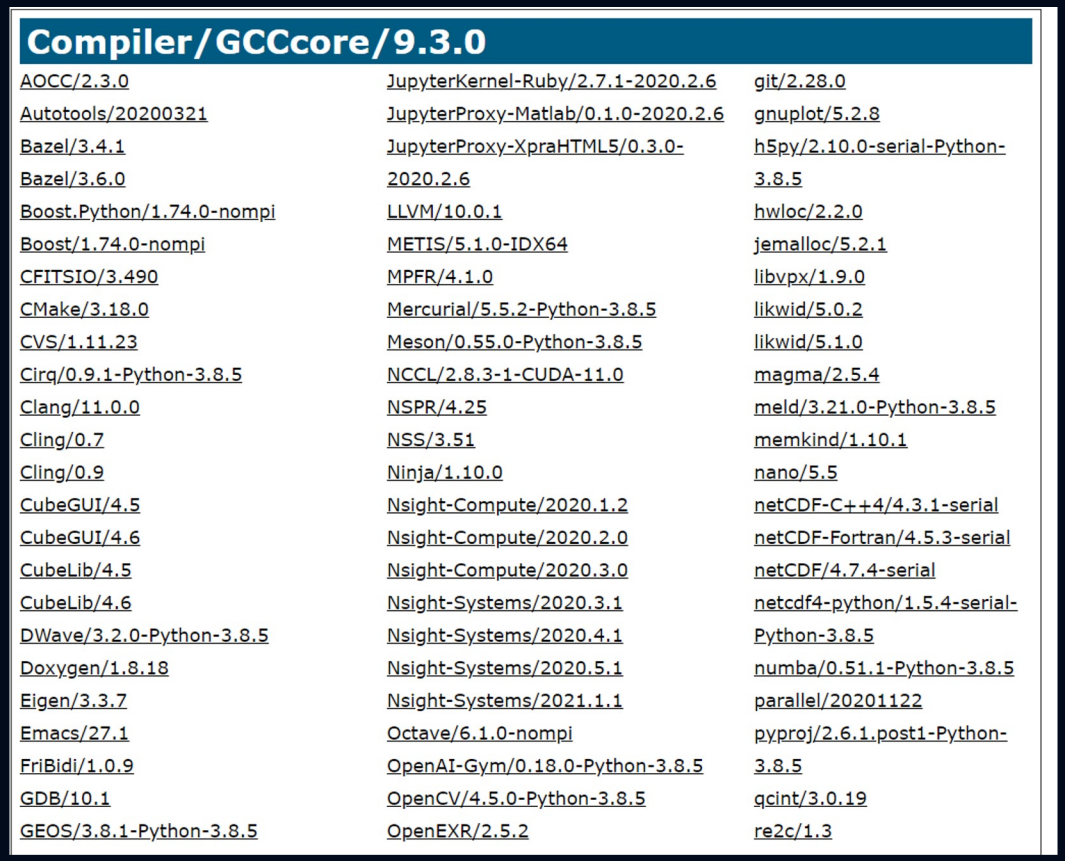
SUPERCOMPUTER USAGE MODEL AT JSC

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.

MODULE SYSTEM: SOFTWARE

All kind of software is already installed in modules.



Compiler/GCCcore/9.3.0

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

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
```

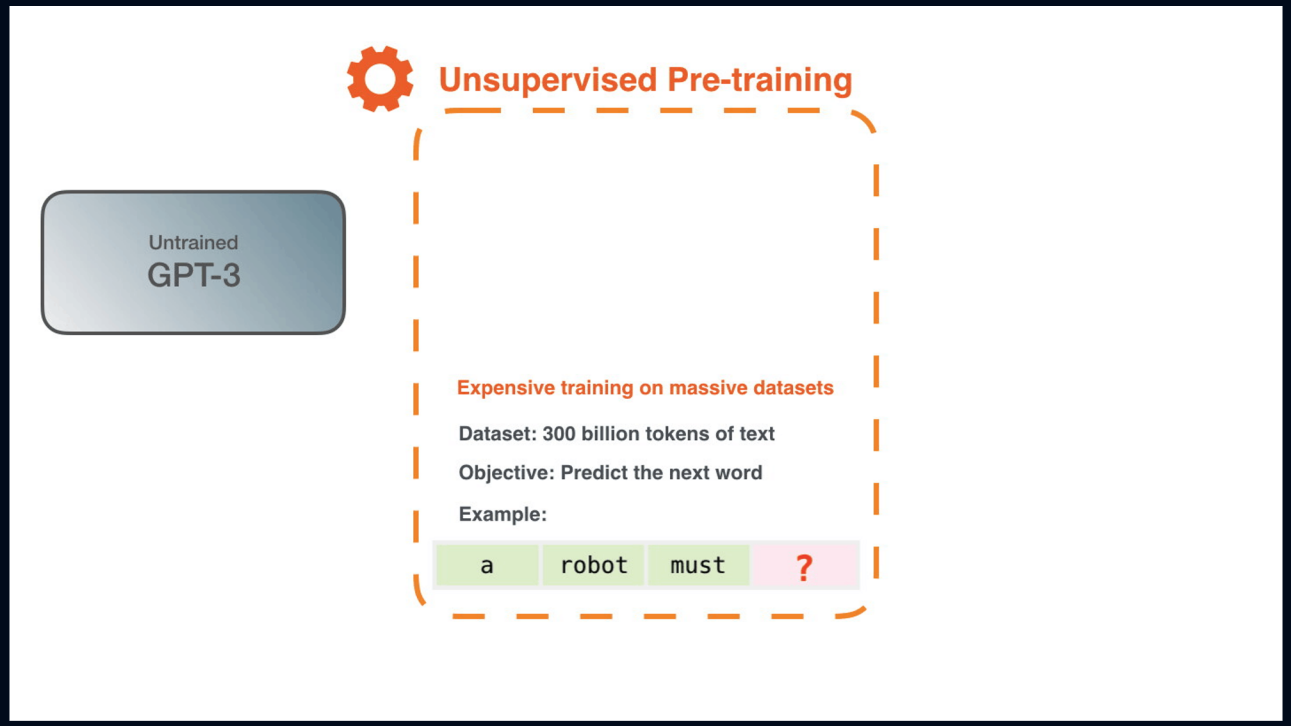
module spider nano

```
[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.
```

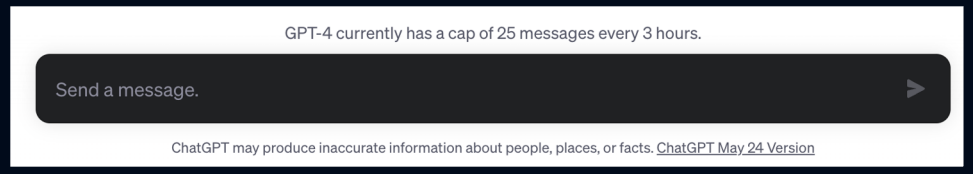
module load nano

FOUNDATION MODELS



Jay Alammar, How GPT3 Works - Visualizations and Animations, http://jalammar.github.io/how-gpt3-works-visualizations-animations/

- Large deep learning models trained on a vast quantity of data at scale (often by self-supervised learning or semi-supervised learning)
- They can be adapted to a wide range of downstream tasks
- Early examples of foundation models were pre-trained large language models, e.g., GPT foundation models



https://chat.openai.com/?model=gpt-4

HOW TO CREATE A FOUNDATION MODEL?

1) Gather data at scale

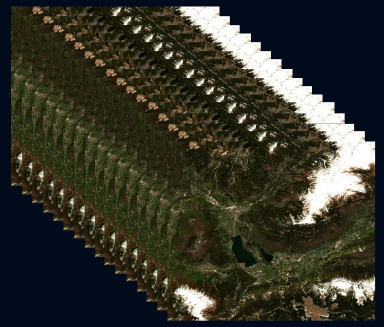
2) Train foundation model one time and evaluate

3) Fine-tune model for multiple downstream tasks

4) Inference (operational)

2. TRAIN FOUNDATION MODEL ONE TIME AND EVALUATE

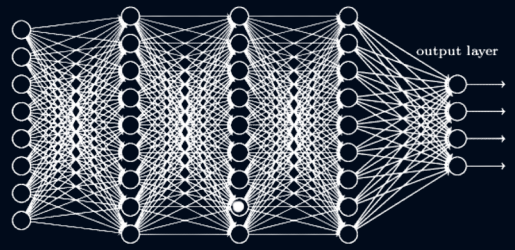
Big Data



Satellite Image Time Series

MAIN TRAINING

Trained model



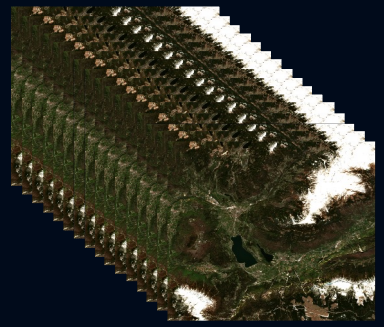
E.g. transformer

High Performance Computing

3. FINE-TUNE MODEL FOR MULTIPLE DOWNSTREAM USES

Data
for specific
events

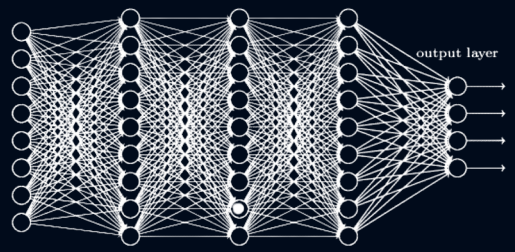
Big Data



Satellite Image Time Series

MAIN
TRAINING

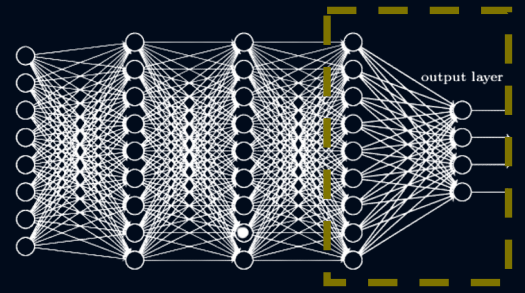
Trained model



E.g. transformer

FINE
TUNING

Fine-tuned model



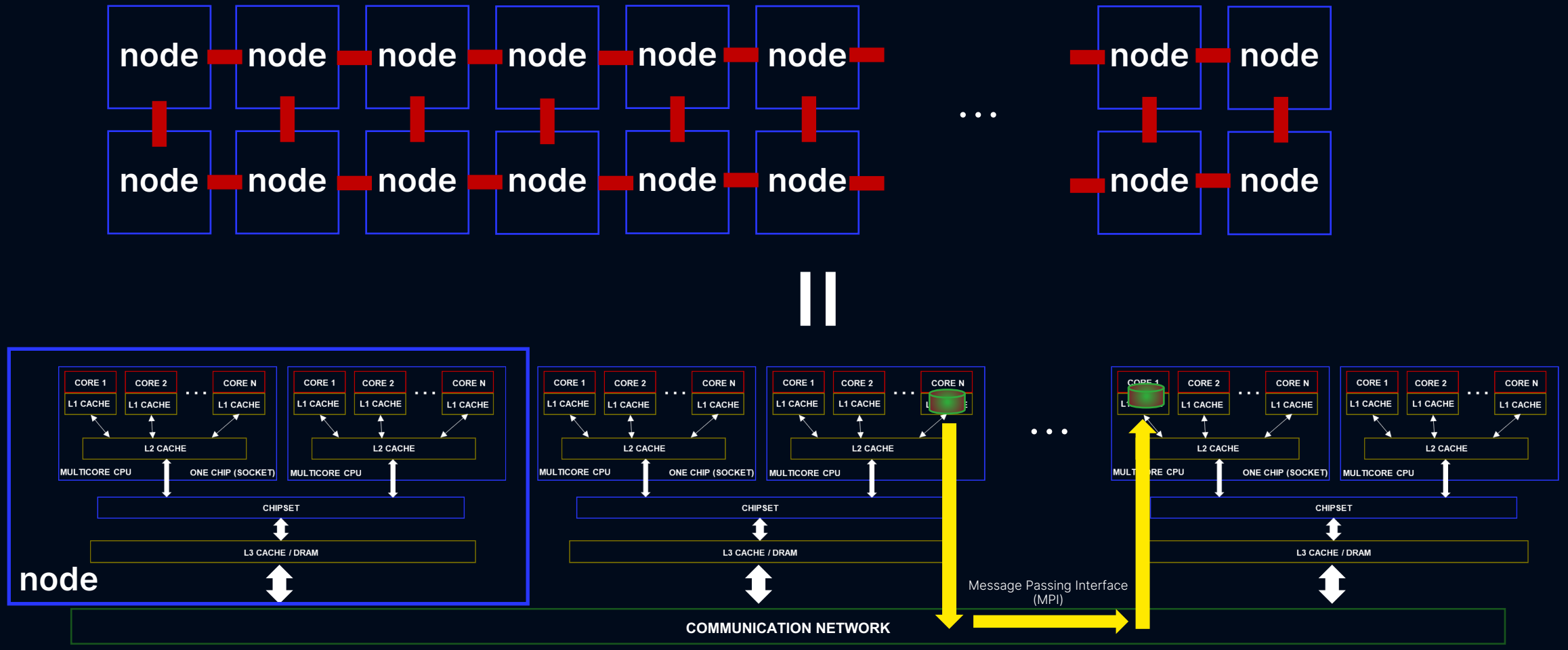
High Performance Computing

Cloud Computing,
Computing workstations

DISTRIBUTED DEEP LEARNING WITH HPC

WHAT IS A SUPERCOMPUTER?

Mixture of shared-memory and distributed-memory architectures



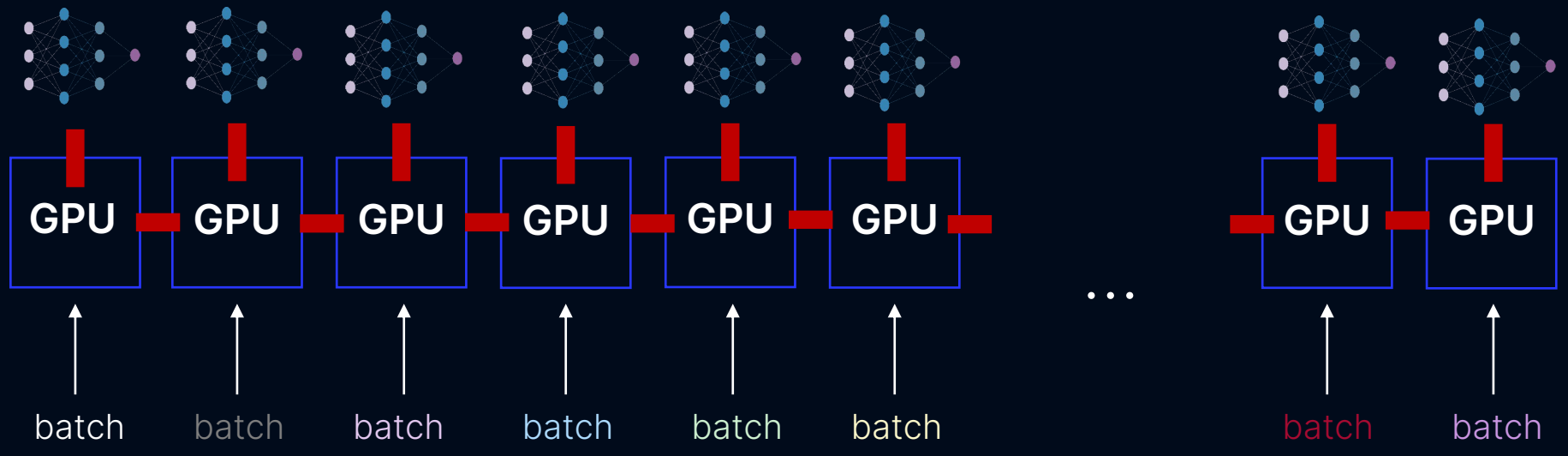
RECALL MINI-BATCH GRADIENT DESCENT

1. Initialize weights randomly $\sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence
3. Pick batch of B data points
4. Compute gradient $\nabla h(W) = \frac{1}{B} \sum_{i=1}^{B} \nabla h_i(W)$

Can use parallelization

DISTRIBUTED TRAINING WITH DATA PARALLELISM: REPLICATE THE MODEL AND TRAIN ON LOCAL BATCHES

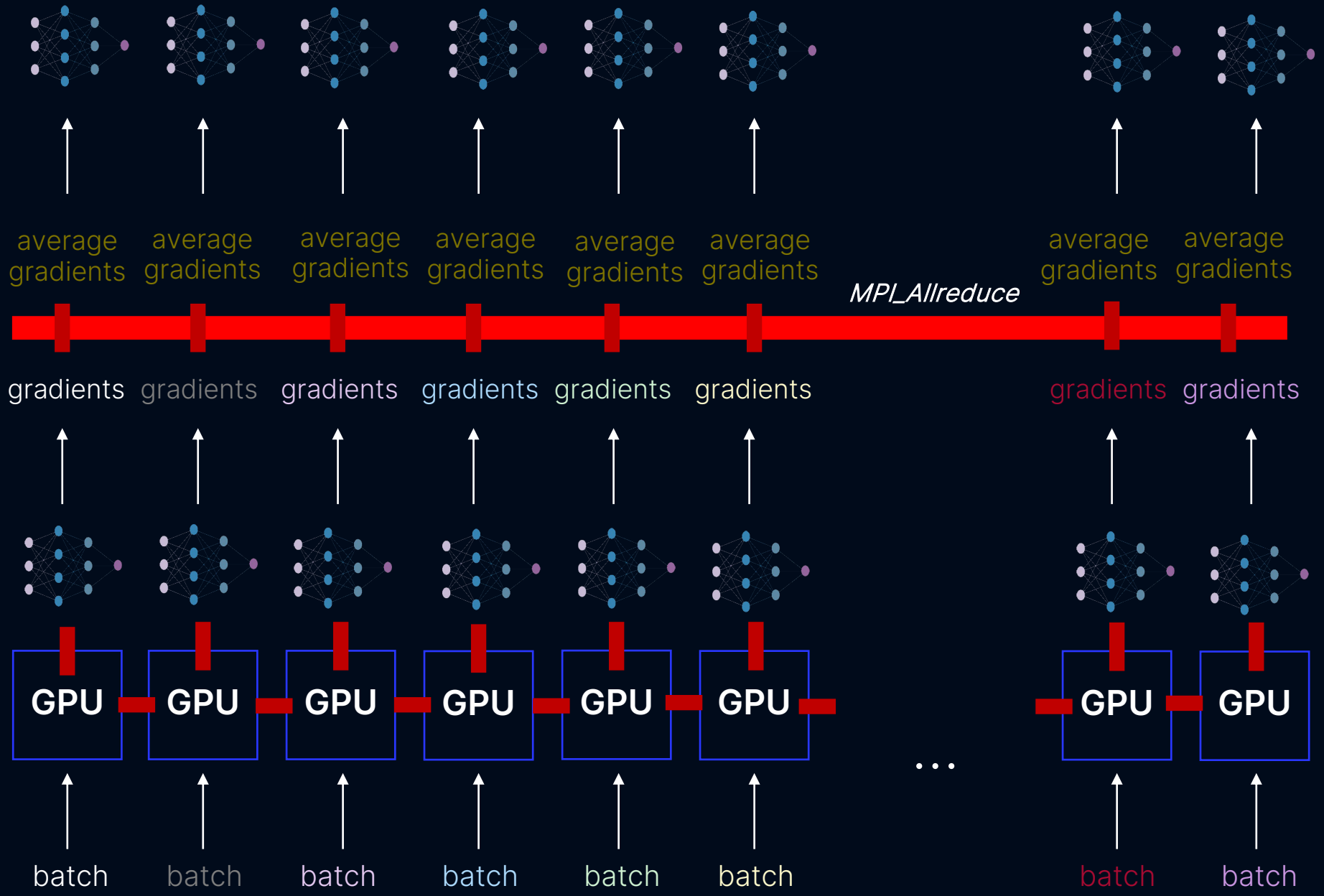
1. READ LOCAL BATCHES



4. UPDATE MODEL

3. AVERAGE THE GRADIENTS

2. COMPUTE LOCAL GRADIENTS



DISTRIBUTED DEEP LEARNING FRAMEWORKS FOR HPC

https://github.com/microsoft/DeepSpeed

https://pytorch.org/tutorials/

...

...

https://github.com/horovod/horovod

https://github.com/helmholtz-analytics/heat

https://github.com/cc-hpc-itwm/tarantella

HOROVOD: DISTRIBUTED TRAINING FRAMEWORK WITH DATA PARALLELISM

Works on top of MPI and NCCL

On top of Tensorflow, Keras, PyTorch, MxNet

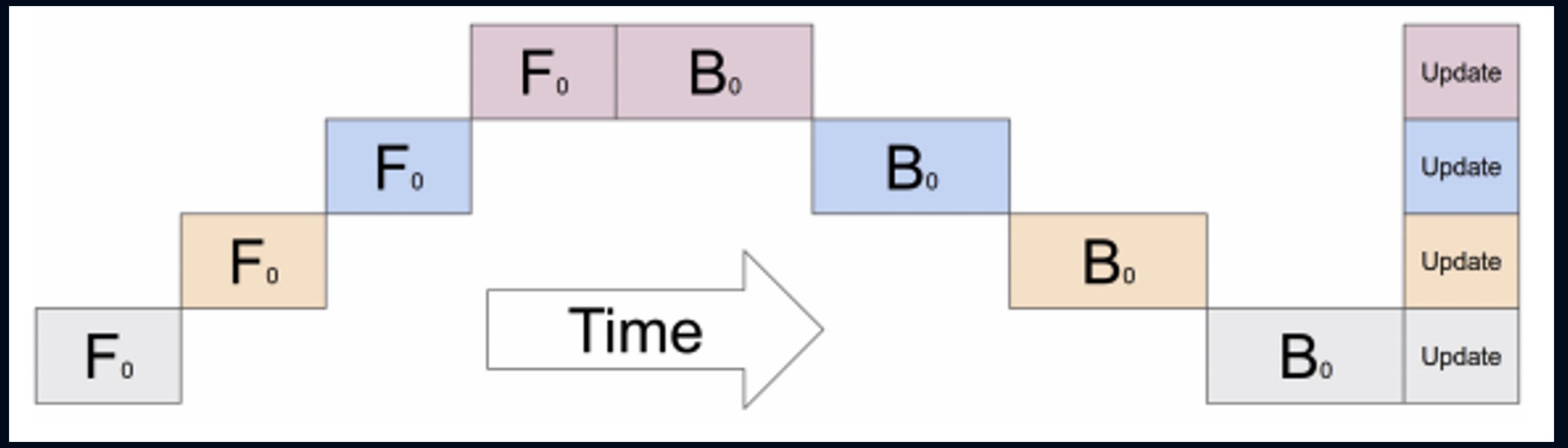
A. Sergeev and M. D. Balso, "Horovod: Fast and Easy Distributed Deep Learning in TensorFlow", arXiv:1802.05799, 2018.

PYTORCH NATIVE DISTRIBUTED ALGORITHMS

PyTorch DDP similar to Horovod

PyTorch DDP https://pytorch.org/docs/stable/pipeline.html

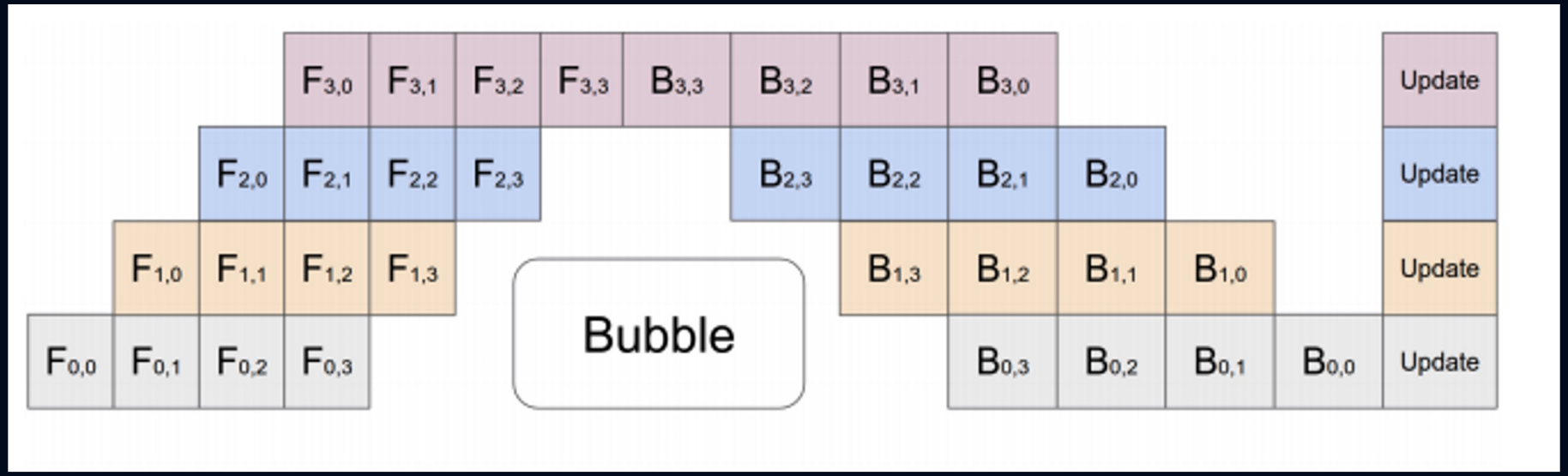
MODEL PARALLELISM



https://pytorch.org/docs/stable/pipeline.html

Model parallelism splits the model across multiple GPUs

PIPELINING



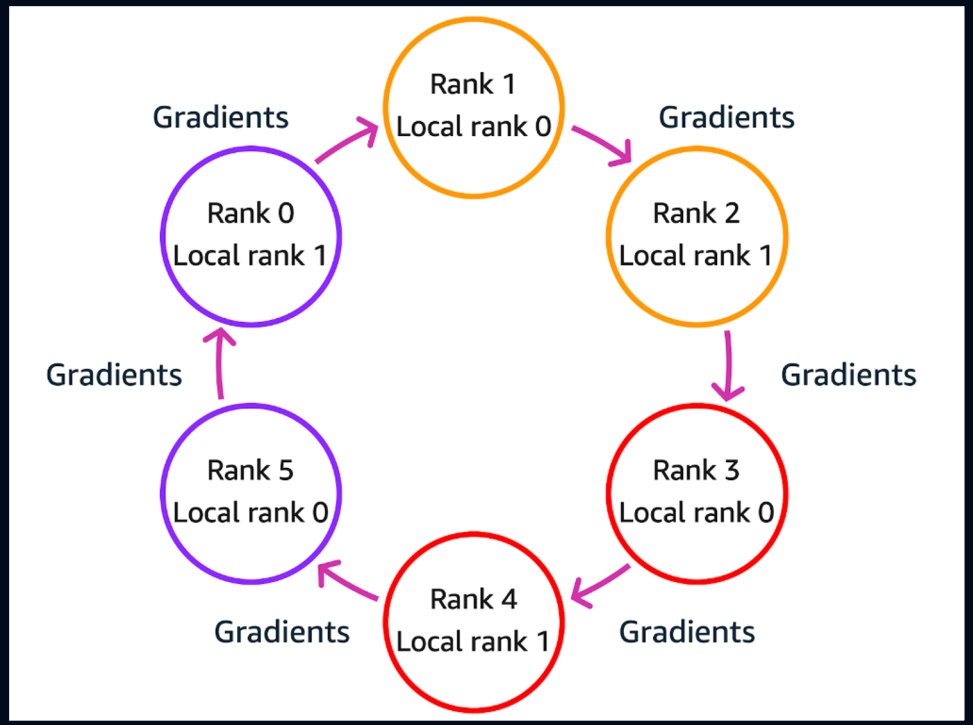
https://pytorch.org/docs/stable/pipeline.html

Pipelining splits the input minibatch into multiple microbatches and pipelines the execution of these microbatches across multiple GPUs

HOROVOD USES RING-ALLREDUCE TO AVERAGE GRADIENTS AND UPDATE ALL COPIES OF THE MODELS

Arranges processes in a logical ring

Each process receives data from it's "left" neighbor and sends data to it's "right" neighbor



Makes communication cost independent of the number of processes

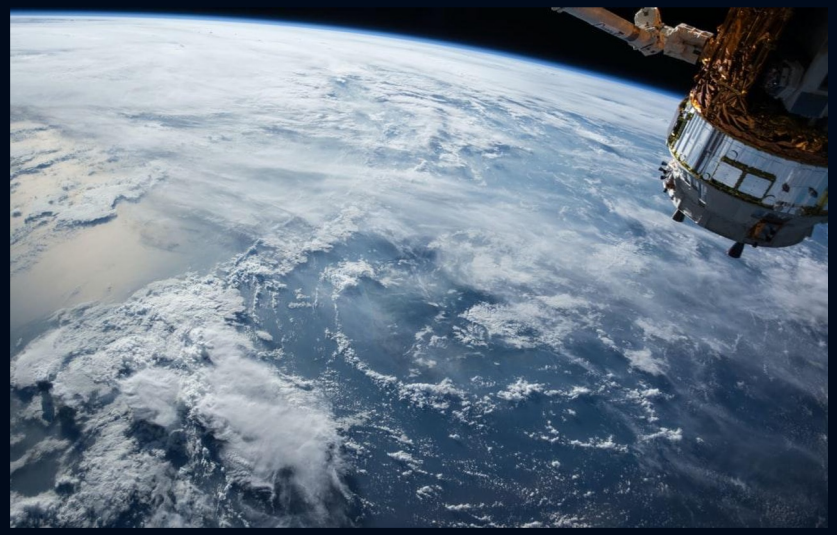
A quick guide to distributed training with TensorFlow and Horovod on Amazon SageMaker https://towardsdatascience.com/a-quick-guide-to-distributed-training-with-tensorflow-and-horovod-on-amazon-sagemaker-dae18371ef6e

HDCRS

GEOSCIENCE AND REMOTE SENSING SOCIETY (GRSS)

One of the 39 societies of IEEE

Mission



Develop concepts and techniques of RS of the Earth, oceans, atmosphere, and space, as well as processing, interpretation, and dissemination of this information for the benefit of society

Facts

Founded in 1961

~4,200 members in 94 countries

69 chapters, 22 student chapters, and 11 ambassadors all over the world

GRSS TECHNICAL COMMITTEES

Earth Science Informatics

Frequency allocations in Remote Sensing

Geoscience Spaceborne Imaging Spectroscopy

Image Analysis and Data Fusion

Instrumentation and Future Technologies

Modeling in Remote Sensing

Remote sensing Environment, Analysis and Climate Technologies

GRSS Standards for Earth Observations

EARTH SCIENCE INFORMATICS (ESI) TECHNICAL COMMITTEE

Objectives

Advance application of informatics to geoscience and remote sensing

https://www.grss-ieee.org/technical-committees/earth-science-informatics/

Chairs

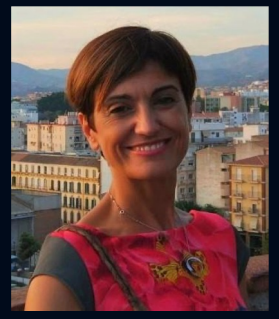
Peter Baumann

Manil Maskey

TWO WORKING GROUPS

High-performance and Disruptive Computing in Remote Sensing (HDCRS)

Gabriele Cavallaro



Dora Blanco Heras

Jin Sun

Databases in Remote Sensing (DBRS)

Dai-Hai Ton That

Kesheng (John) Wu

Khalid Belhajjame

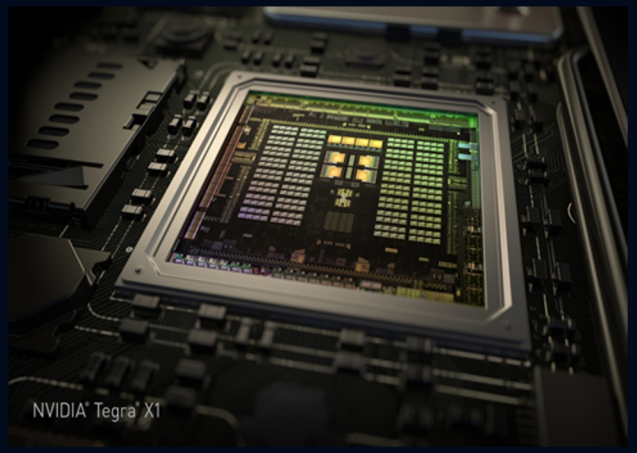
High Performance and Disruptive Computing in Remote Sensing Working Group

Main Objective:

Connect and support the community of interdisciplinary researchers in remote sensing who are specialized in emerging computing paradigms

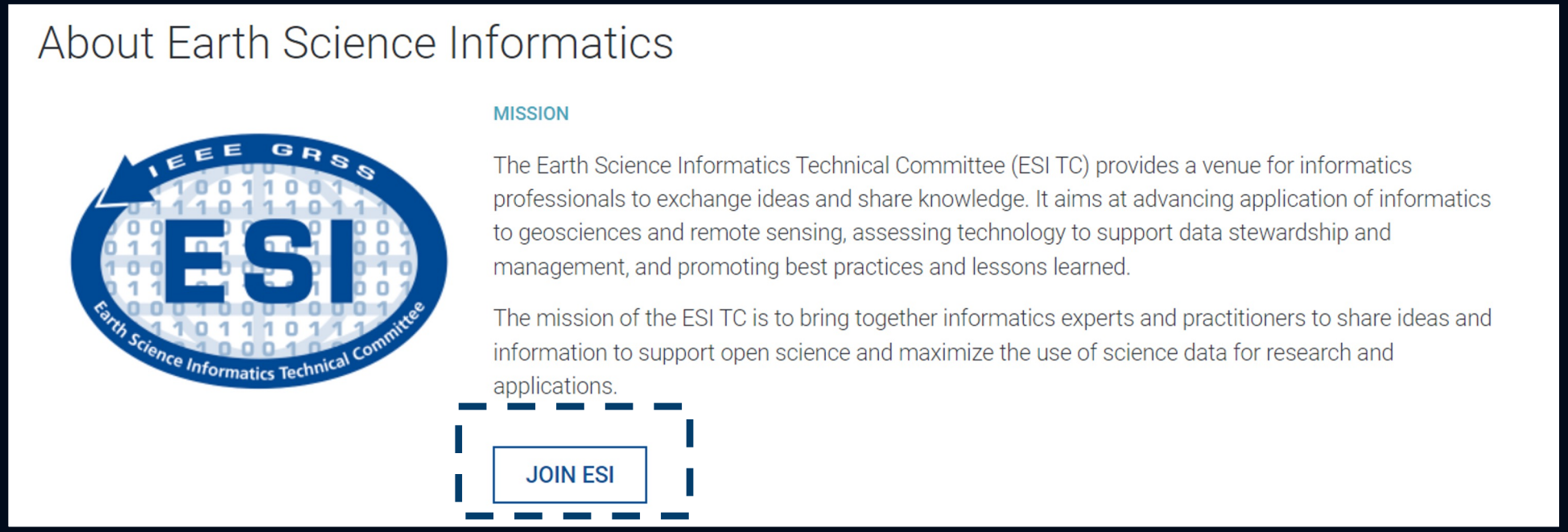
The idea:

Innovative computing technologies applied to efficient computation of remote sensing problems



https://www.grss-ieee.org/community/groups-initiatives/high-performance-and-disruptive-computing-in-remote-sensing-hdcrs/

COME ABOARD



https://www.grss-ieee.org/technical-committees/earth-science-informatics/

Activities

HDCRS SUMMER SCHOOL - 2023

Teaching material and videos will be soon available

https://www.grss-ieee.org/community/groups-initiatives/high-perfomance-and-disruptive-computing-in-remote-sensing-hdcrs/hdcrs-summer-school-2023/

ACTIVITIES AT IGARSS 2023

Tutorial on End-to-End Machine Learning with Supercomputing and in the Cloud

Hands-on experience in Supercomputer Systems and Cloud Computing Services for Remote Sensing Applications

Community Contributed Sessions

(1) Scalable Parallel Computing for Remote Sensing
(2) Quantum computing next generation HPC
(3) Quantum Machine Learning algorithms for EO

IGARSS
Pasadena, California 2023

https://2023.ieeeigarss.org/

Thank you for your attention