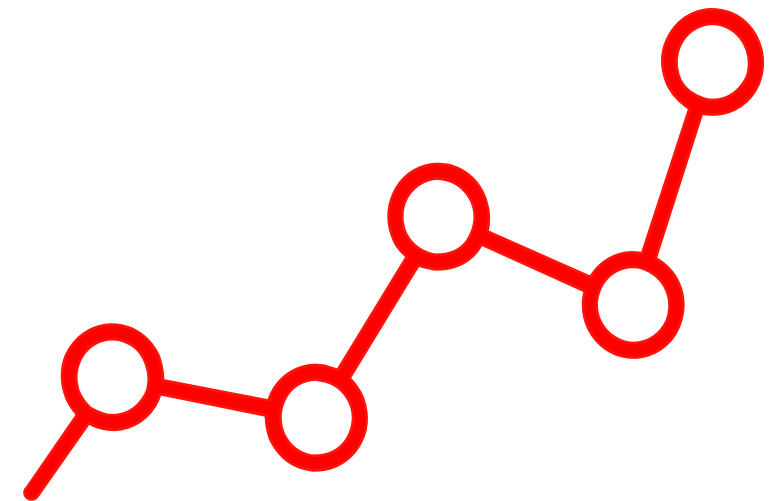
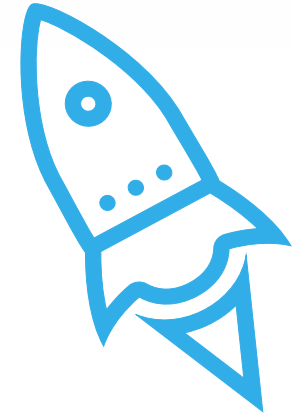


Optimizing GPU Usage for Deep Learning Workloads: Insights and Strategies for Enhanced Efficiency

13th June 2023

Pierrick Pochelu – PCOG

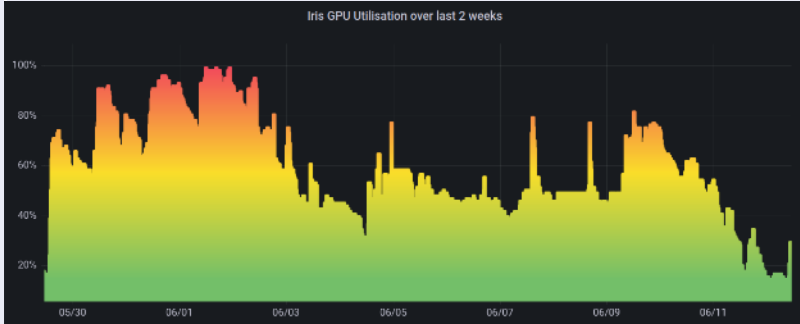
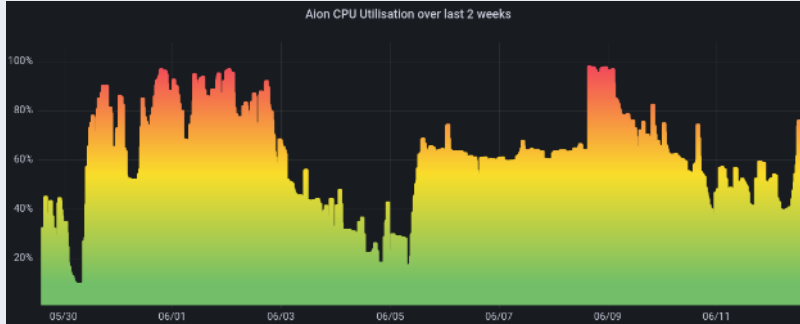
1. Importance of GPU workload (data from SLURM & Survey)
2. GPU workload challenges & potential methods
3. Accelerator trends and first feedbacks with IPU
4. User learning activity (User perspective)
5. Conclusion & My actions



1. Importance of GPU workload

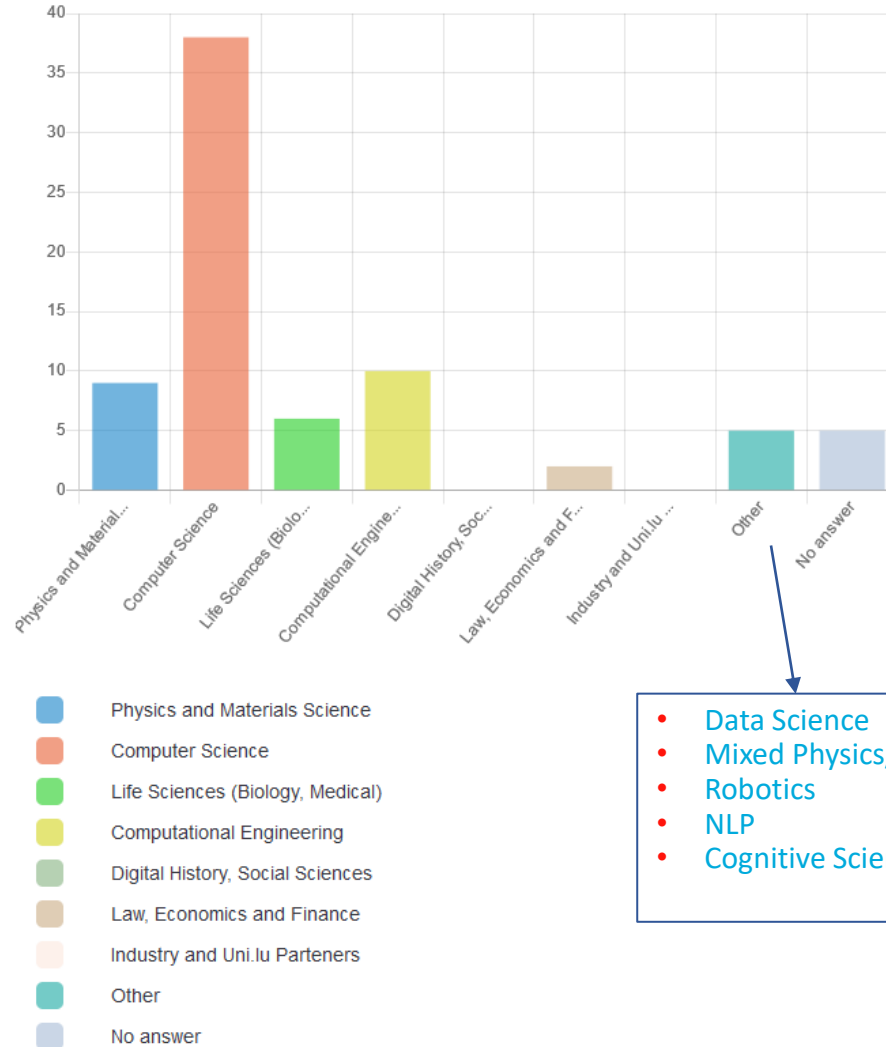
GPU more and more useful in HPC centers for dense computing

Statistics from UL HPC SLURM:

	Iris GPU	Aion CPU
From 1-1-2023 to 12-6-2023	64.3%	57%
Last 2 weeks	 <p>Mean:57.6% Std:19.8%</p> <p>Mean real utilization: 20.5%</p>	 <p>Mean:59.2% Std:20.6%</p>

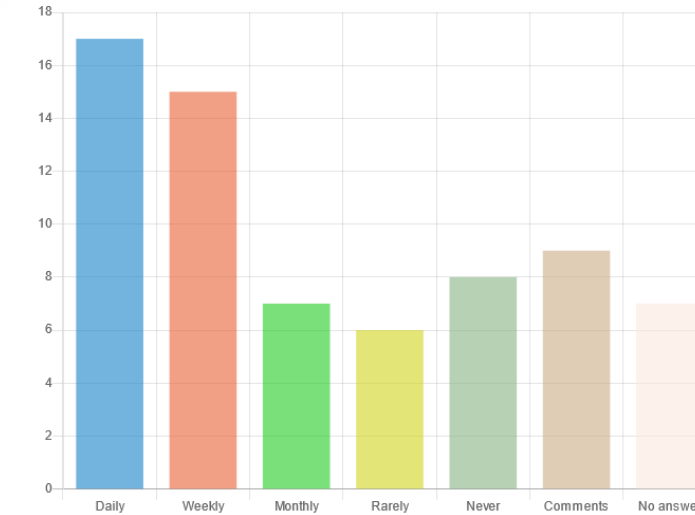
Source: Data from ULHPC Sys Admin

What best describes your background ?



- Data Science
- Mixed Physics/CS
- Robotics
- NLP
- Cognitive Science

How often do you use GPU?



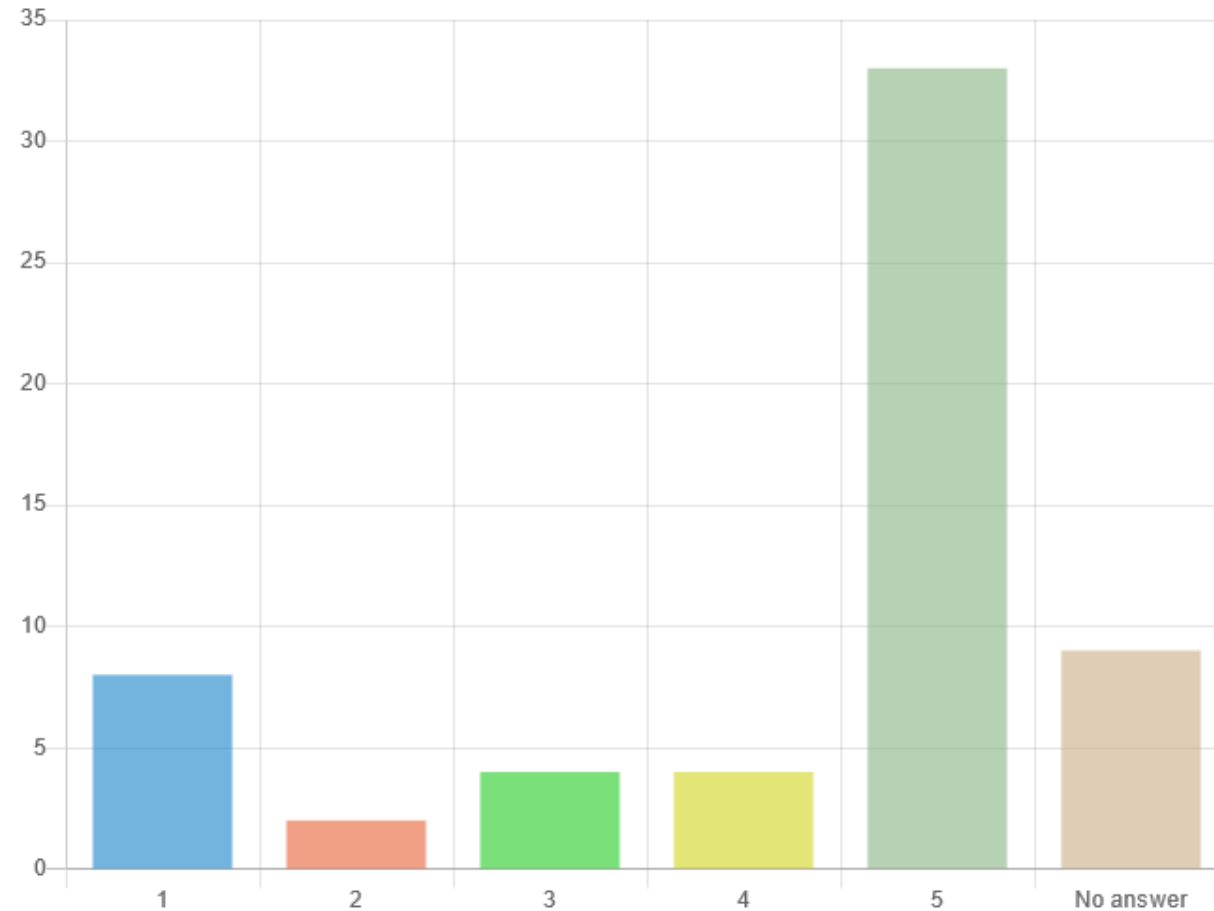
- Intermittent but intensive
- Depends
- Daily in certain periods
- I use it when i need to submit a paper.

Survey submitted to UL HPC users, 75 participants, no personally identifiable information was disclosed, conducted with [with survey.uni.lu](https://survey.uni.lu)

At which extent those claim are true ? 5 means "strongly agree" and 1 "strongly disagree".

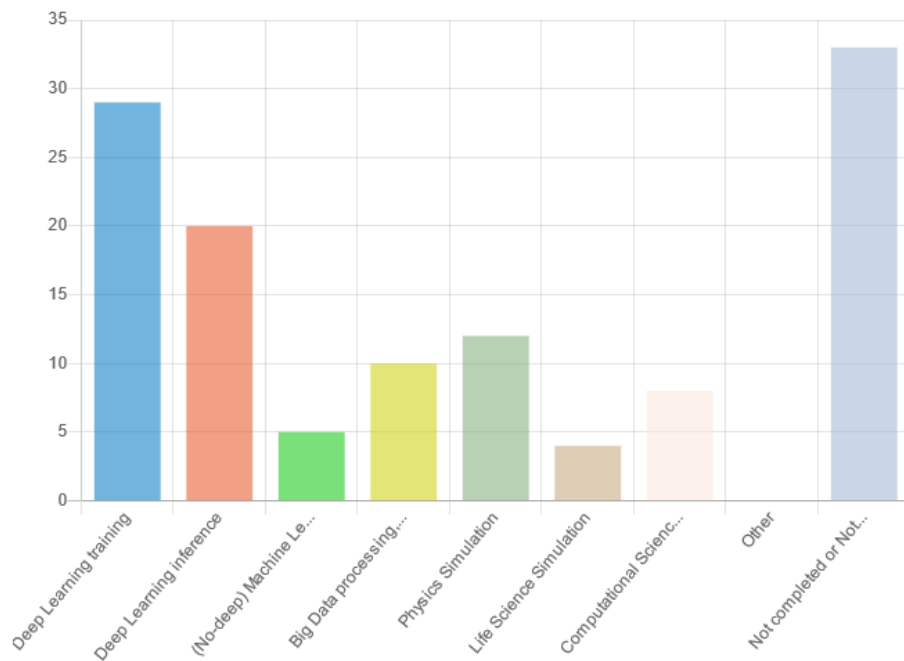
[GPUs are essential to my work]

Arithmetic mean 4.02 Standard deviation 1.53



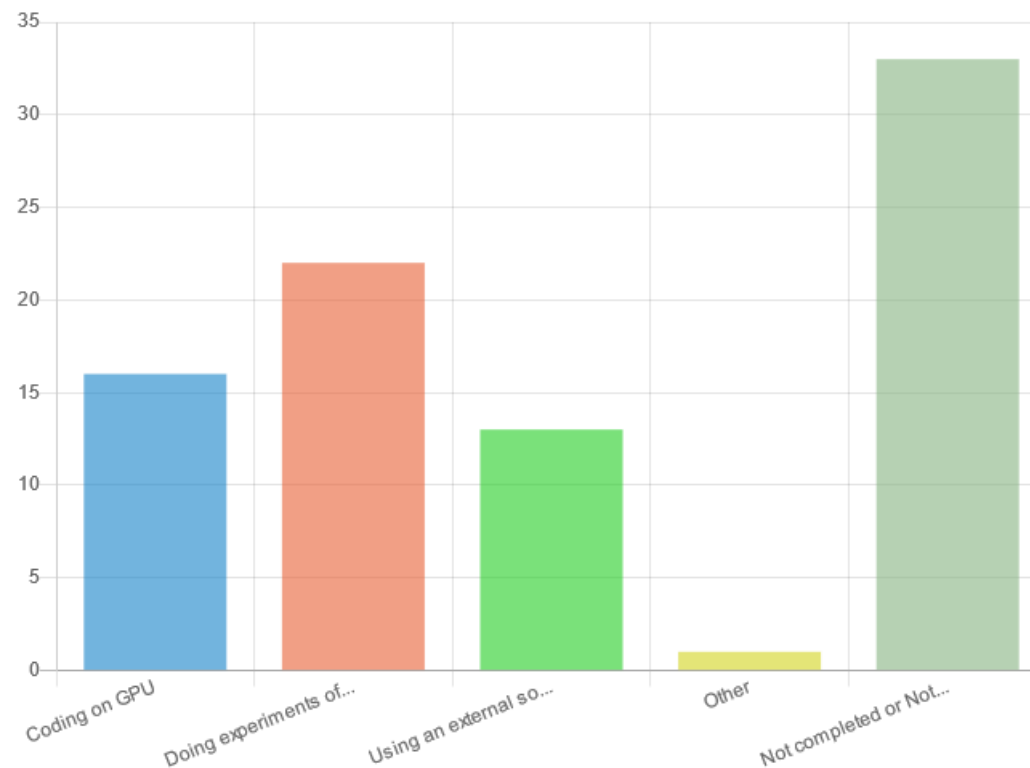
2. GPU workload challenges & potential methods

What type of tasks do you primarily perform that require significant computational resources? Please select the appropriate option:



- Deep Learning training
- Deep Learning inference
- (No-deep) Machine Learning or Artificial Intelligence
- Big Data processing, High Performance Data Analytics
- Physics Simulation
- Life Science Simulation
- Computational Science Simulation
- Other
- Not completed or Not displayed

How do you define your activity on UL HPC GPU ?

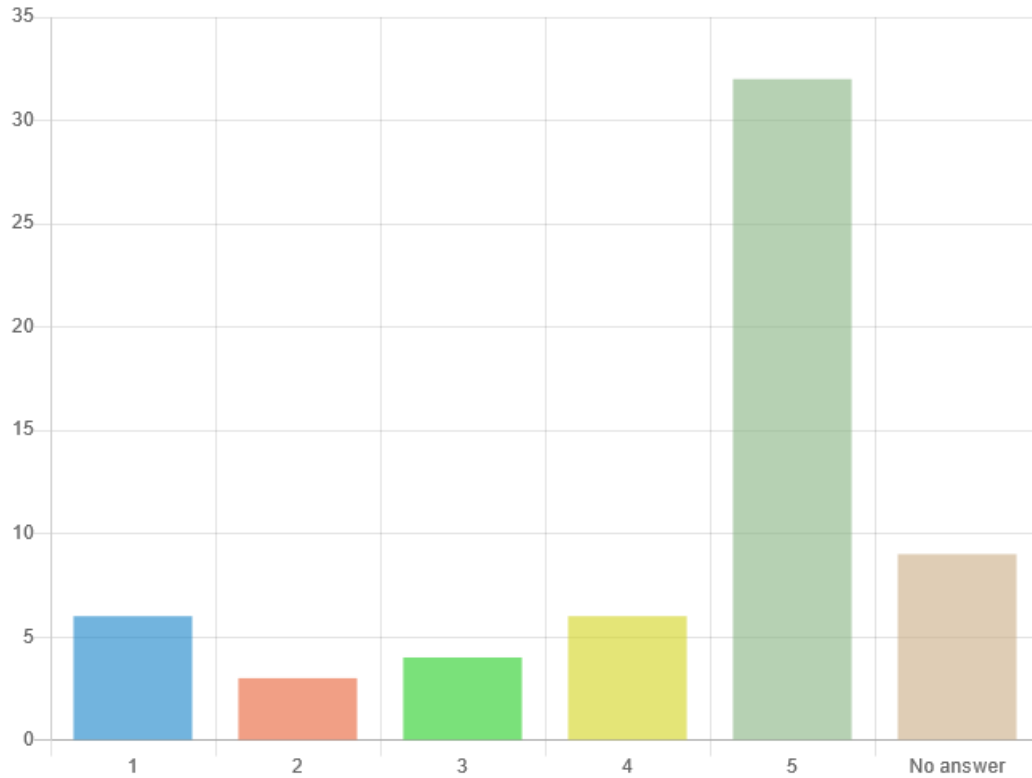


- Coding on GPU
- Doing experiments of in-house code
- Using an external software
- Other
- Not completed or Not displayed

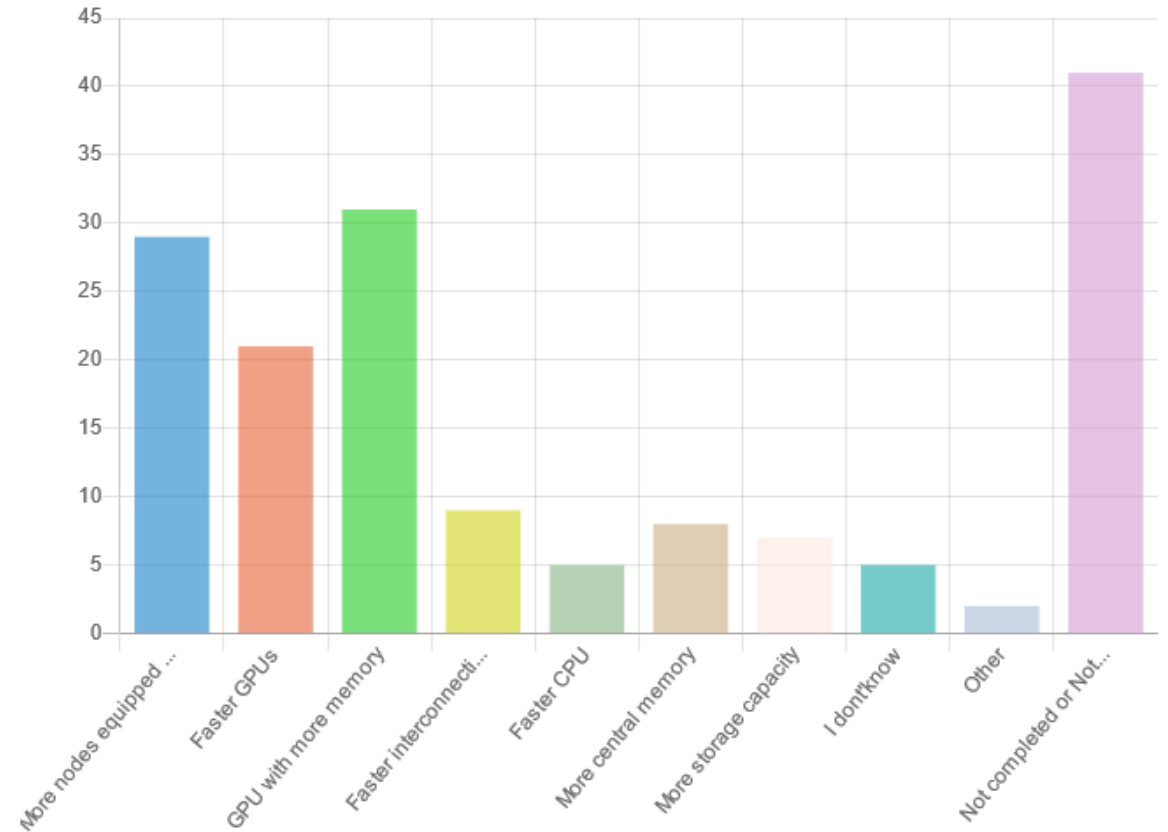
At which extend those claim are true ? 5 means "strongly agree" and 1 "strongly disagree".

[Faster and GPU with more memory would improve my productivity]

Arithmetic mean 4.08 Standard deviation 1.43

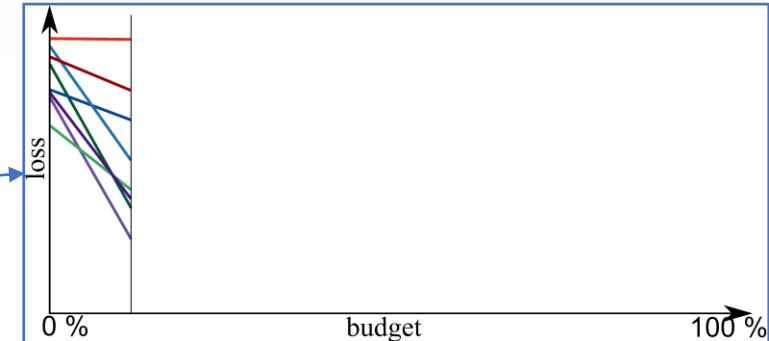


In what ways do you believe the HPC hardware could be enhanced to better suit your specific tasks or requirements?



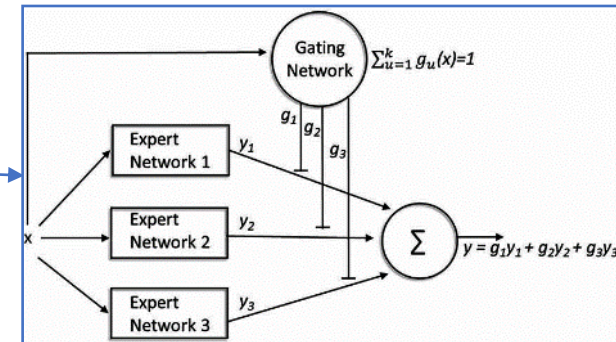
1. Tuning the ML project more efficiently

1. Explore/Exploit
2. Early-Stopping



2. Improving the training speed

- Solution: Data-parallel SGD
 - Problem: Scalability issue
 - Solution: Mixture of Experts



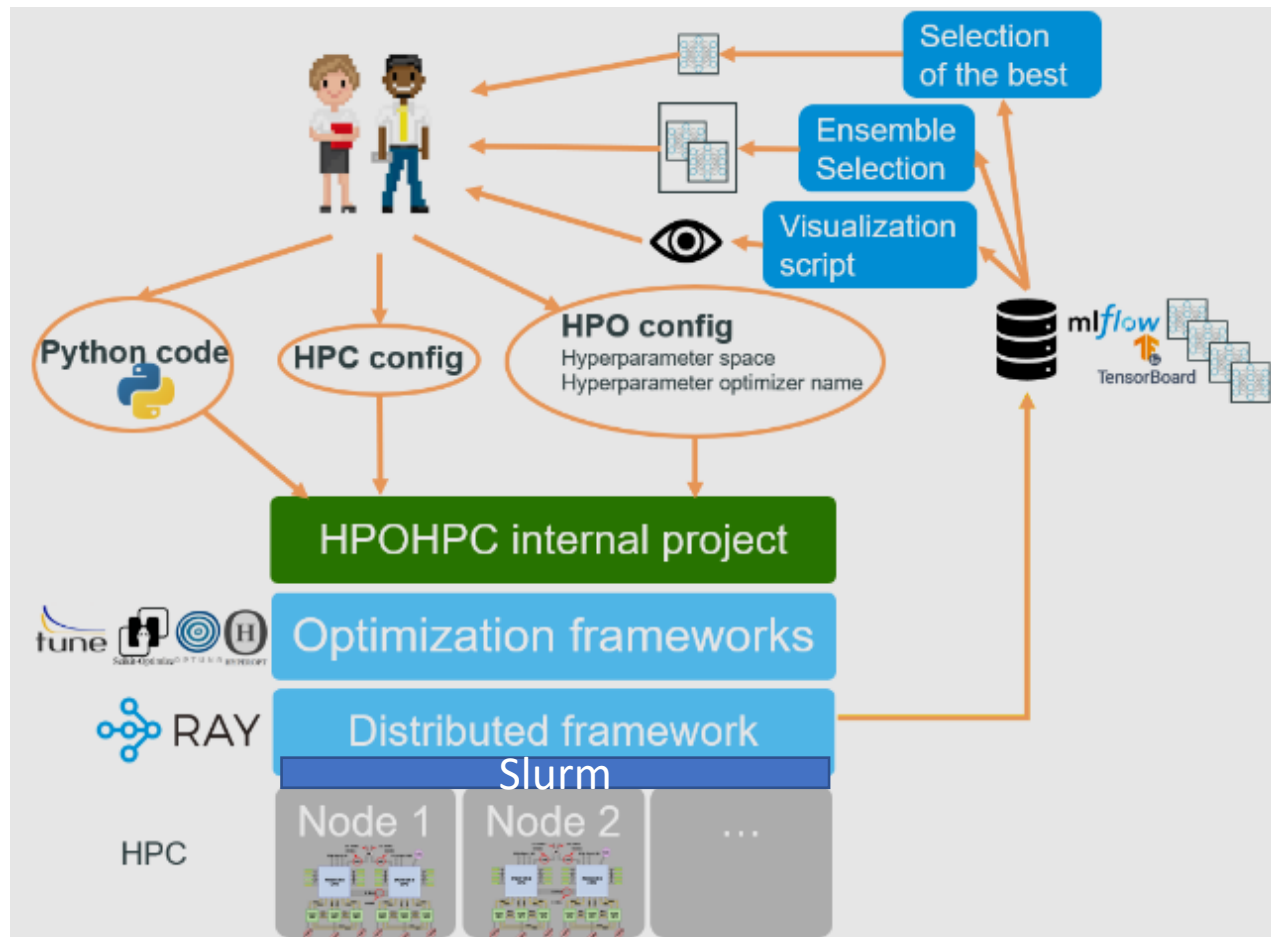
3. Not enough memory

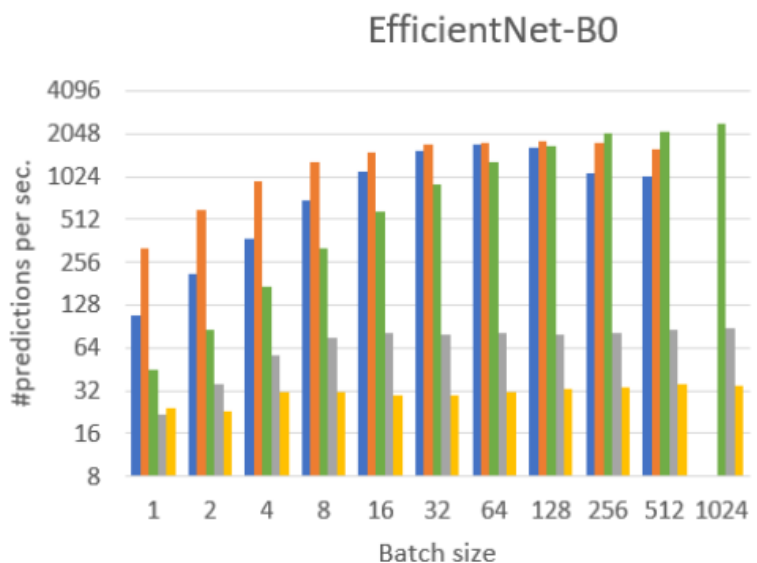
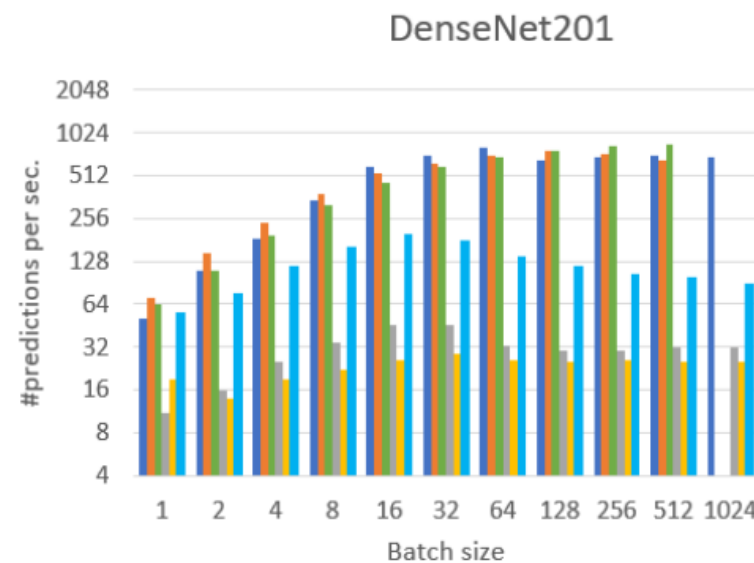
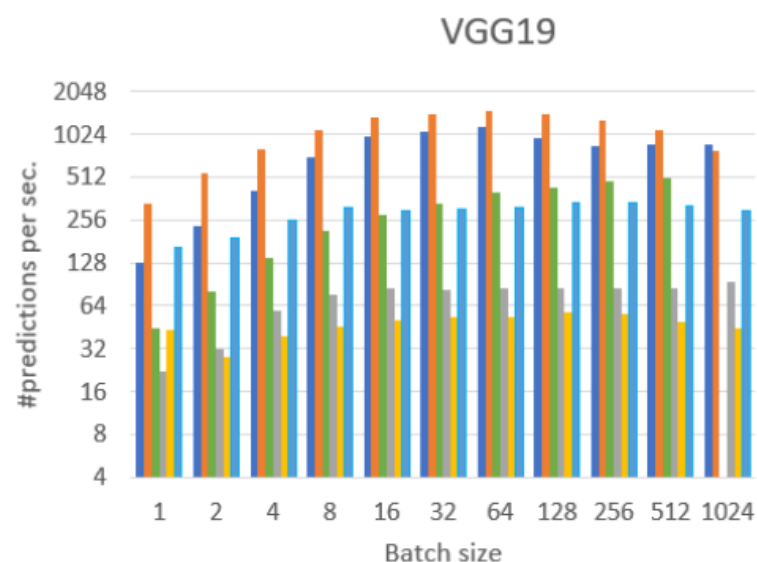
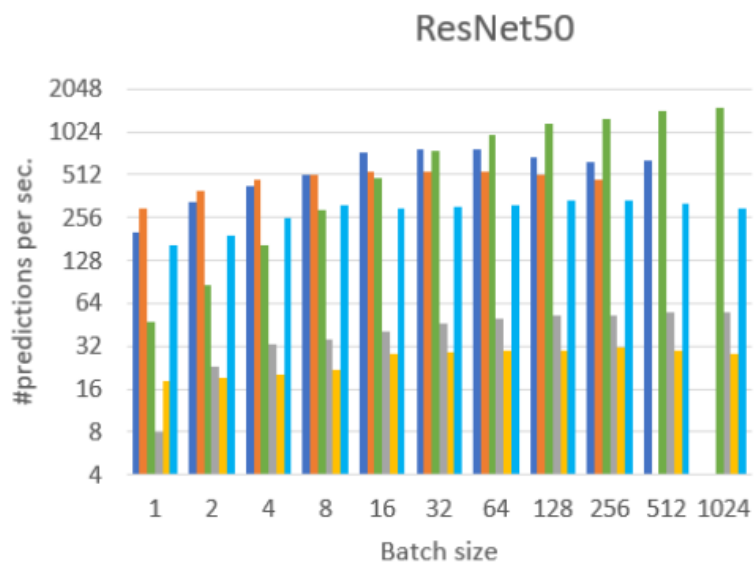
1. Automatic Model Parallelism (GPIPETorch)
2. Compute nodes on GPU \Leftrightarrow DAG stored on CPU (LMS by IBM)

4. Improving the inference speed

1. DAG compiler&compiler such as Tensor-RT, ONNX-RT
2. High-asynchronosity server or IoT

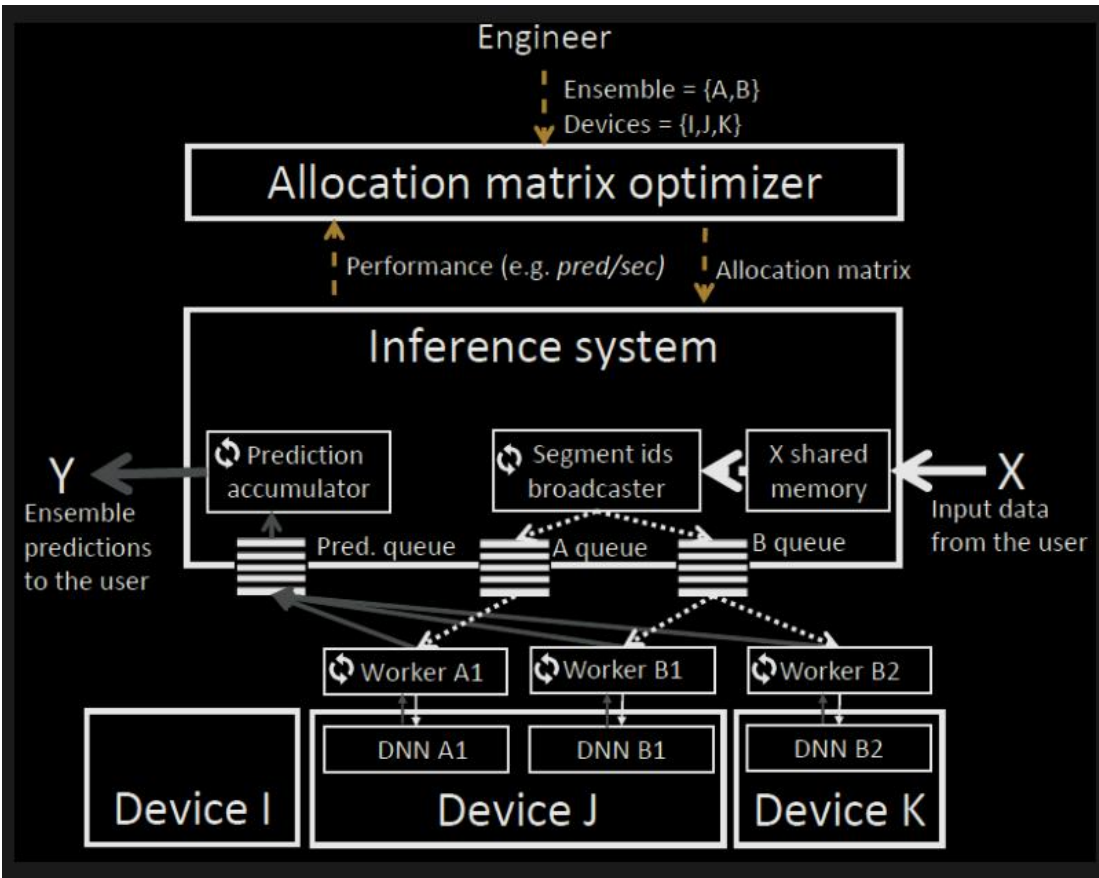
Example of system for optimizing users workload



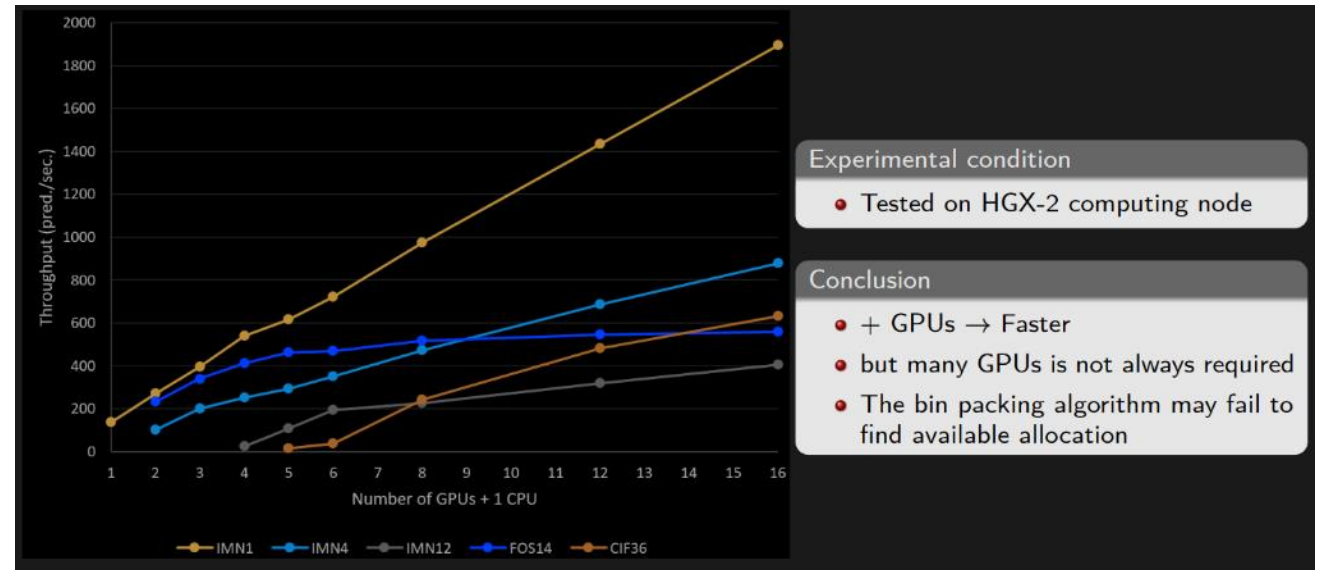


Legend

- Tensorflow GPU
- ONNX-RT GPU
- TensorRT GPU
- Tensorflow CPU
- ONNX-RT CPU
- OpenVINO CPU



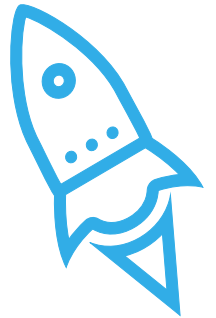
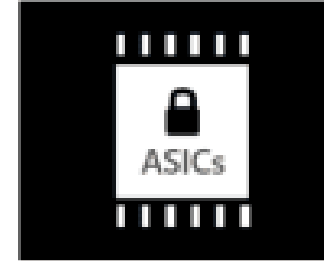
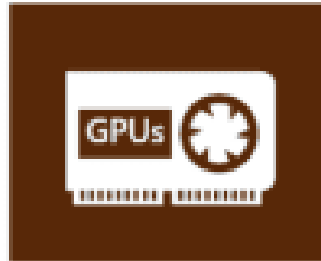
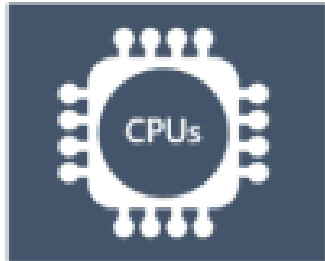
Weak Scalability Analysis



Many comments on the lack of GPU memory and GPUs:

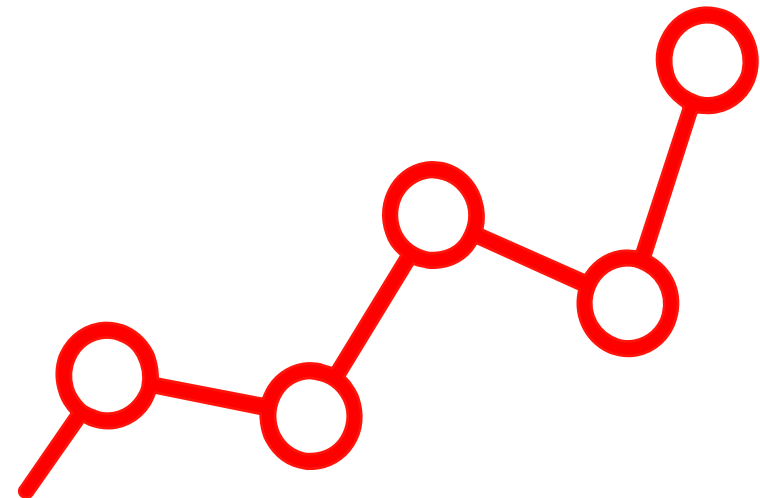
- Not enough GPUs, not enough memory. I am thus using Colab or Lambdacloud instead.
- I experienced a time when the research was limited by the memory limitation of the GPU, even though I used the Tesla v100 32GB. Especially when I want to train a large graph-based deep learning model, I failed to train the network on big-size data.
- I cannot train/finetune very large language model due to memory constraint (A100/H100 x4 80Go required)
- Sometimes, I have to wait quite a long for my submitted jobs to be allocated

3. Accelerator trends

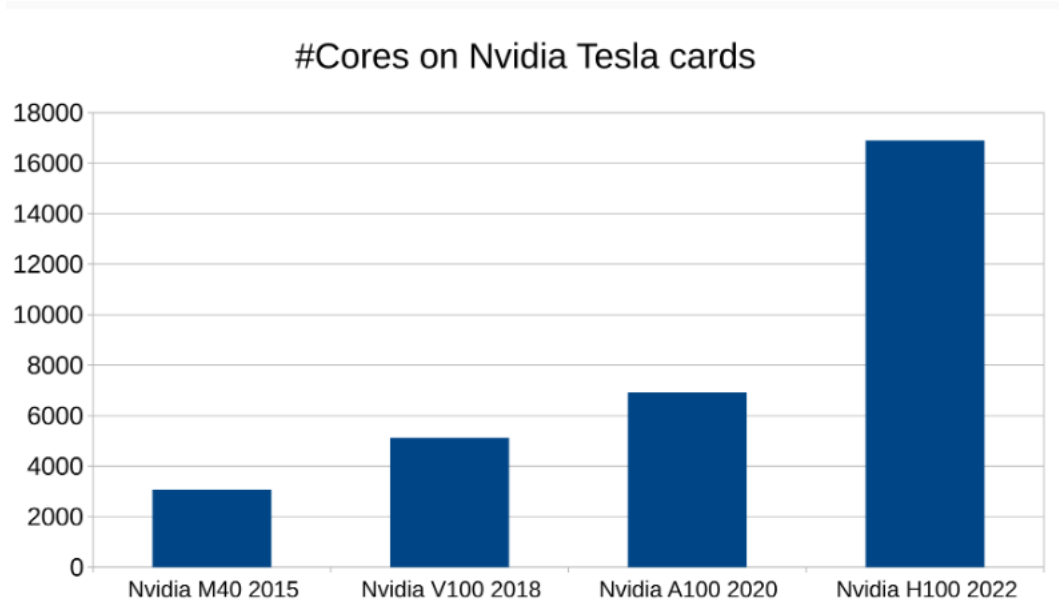


Other hardware includes:

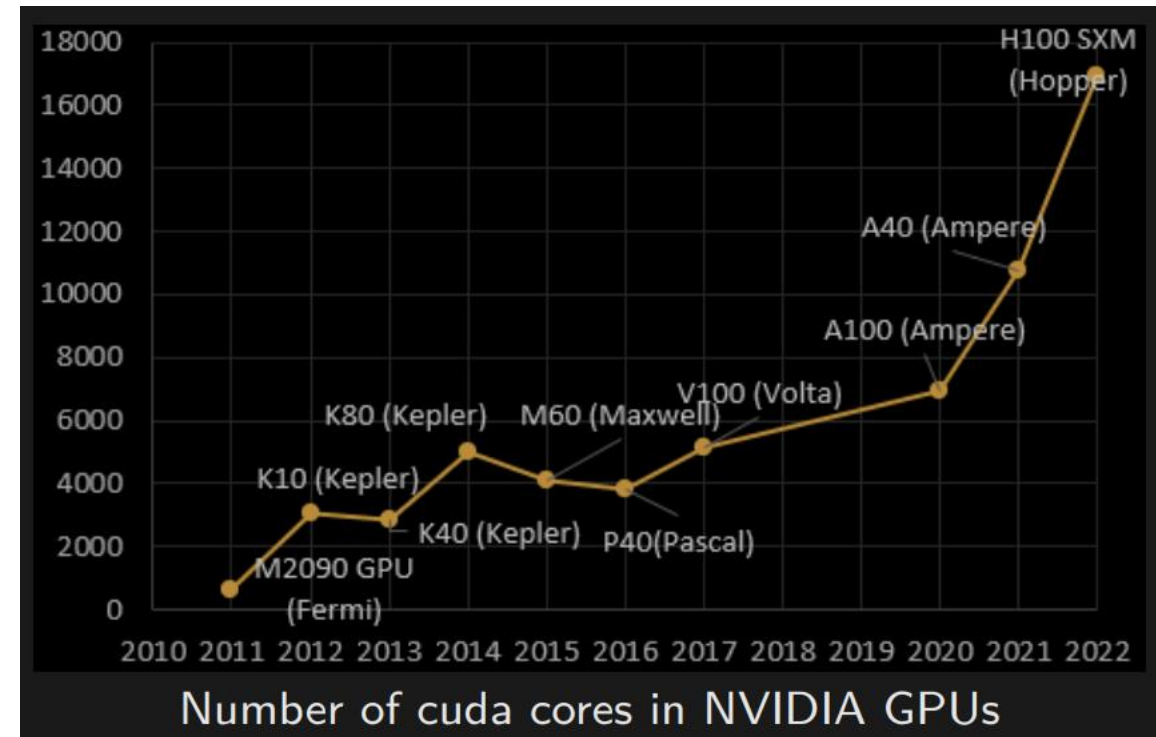
- Quantum computing
- Opto-electronic computing
- Neuromorphic for Spiking Neural Network (E.g., SpiNNaker)



3. Accelerator trends



Retrieved from *P. Talbot. Introduction to CUDA Programming 2023. UL HPC School 2023.*



Retrieved from *P. Pochelu. PhD Defense 2022 at Paris.*

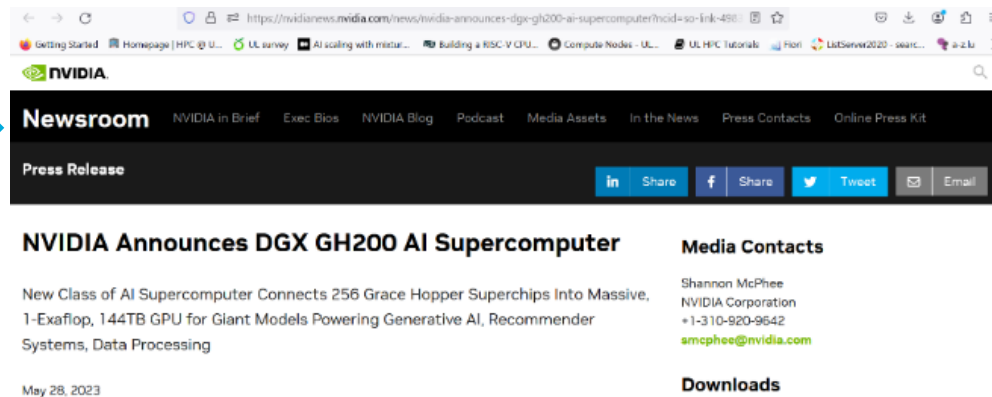
Challenge:

- The #cores will probably finish by plateau
- GPU-CPU connection may bottleneck some applications (e.g. short kernels)
- Scalability inter-GPU is still a challenge (Nvlink/Infiniband still not fast enough)

Solution:

- « This increases the bandwidth between GPU and CPU by 7x compared with the latest PCIe technology »
- « 48x more NVLink bandwidth than the previous generation »
- « [NVIDIA Quantum-2 InfiniBand](#) networking to supercharge data throughput for training large AI models »

SOURCE →



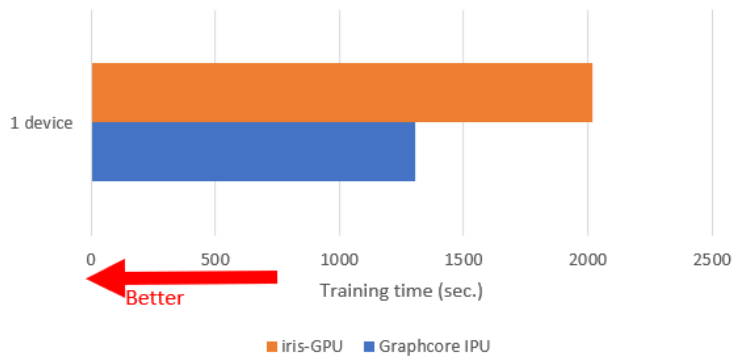
- From GPU code to IPU code:
 - From mono-GPU Tensorflow to mono-IPU (low code impact but version lag)
 - mono-GPU PyTorch → mono-IPU PopTorch (stronger impact : https://docs.graphcore.ai/projects/poptorch-user-guide/en/latest/pytorch_to_poptorch.html)
 - Multi-GPU Tensorflow → Multi-IPU Tensorflow
- Strong performance difference according the model (computing graph):
 - URL: <https://docs.graphcore.ai/projects/memory-performance-optimisation/en/latest/optimising-performance.html>

Good for	Training large models
	Training with large global batch sizes
Not recommended	Low-latency inference

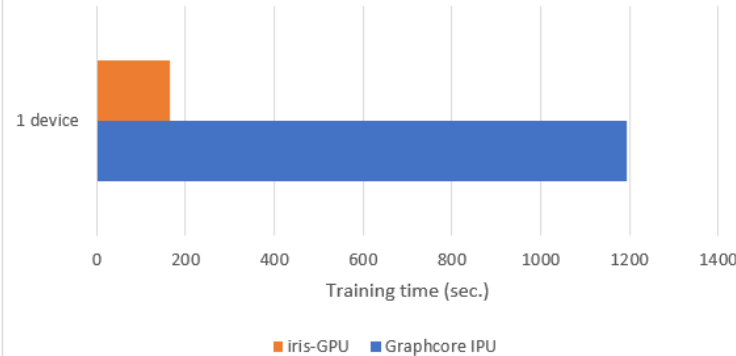
- Compilation from a few seconds (GPU) to a few minutes (IPU) → Caching mechanisms

- Standard Deep Learning operation + CPU RNG produce similar results (diff.<1E-6)
- Preliminary results: IPU can be from x2 faster to x7 slower according the DAG

ResNet50 batch_size=16



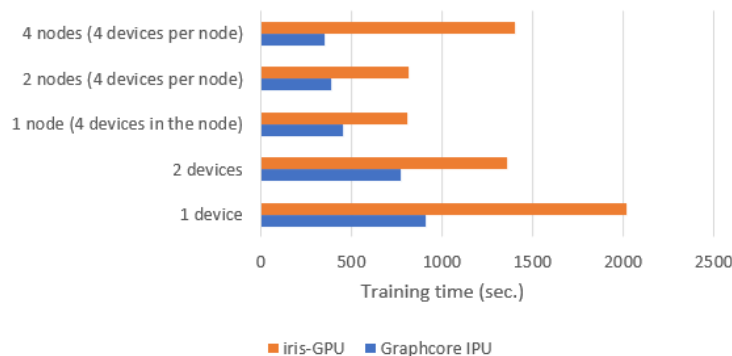
GNN on Cora batch_size=16



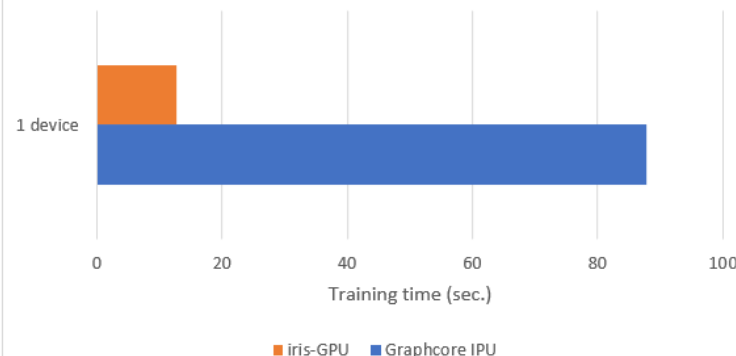
Official code for testing:

- `tf.keras.applications.ResNet50` (20M param.)
- https://keras.io/examples/graph/gnn_citations/ (60K param.)

ResNet50 batch_size=256



GNN on Cora batch_size=256

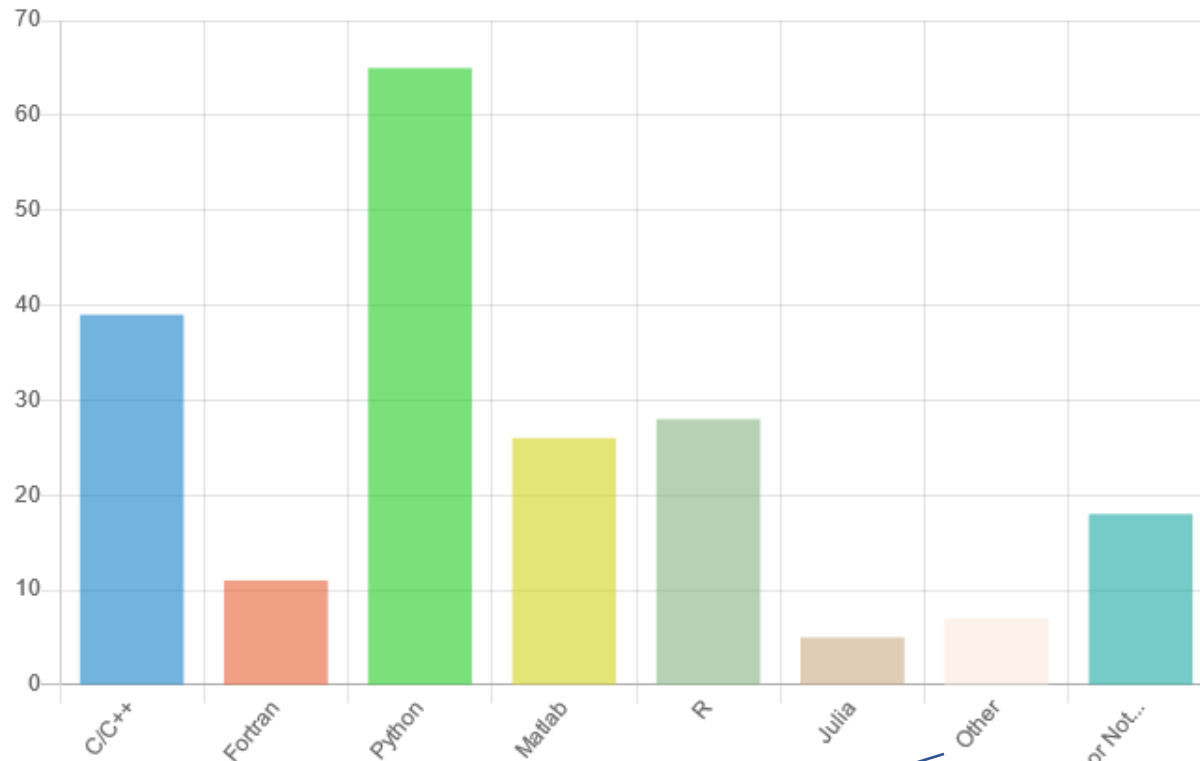


Preliminary conclusion:

- IPU may scale better
- IPU may be significantly faster or slower according the application
- Any heuristics to know if a DAG will be faster on IPU than GPU ? (nodes, parameters, ...)

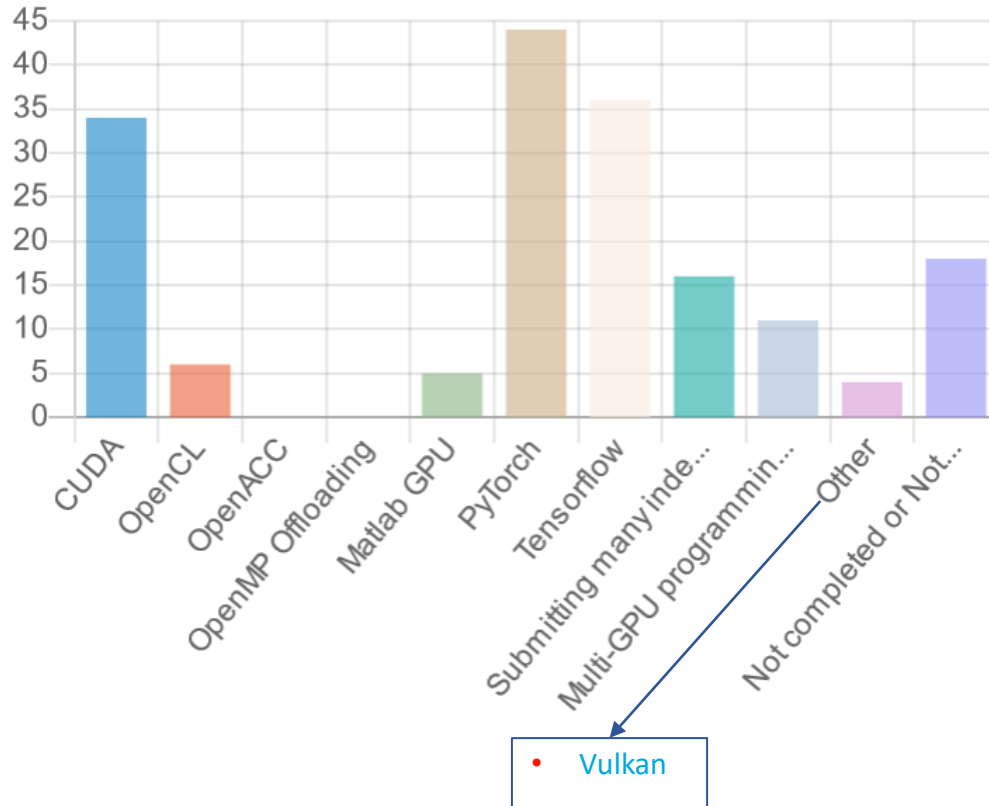
4. User learning activity (User perspective)

Which programming language have you already used ?

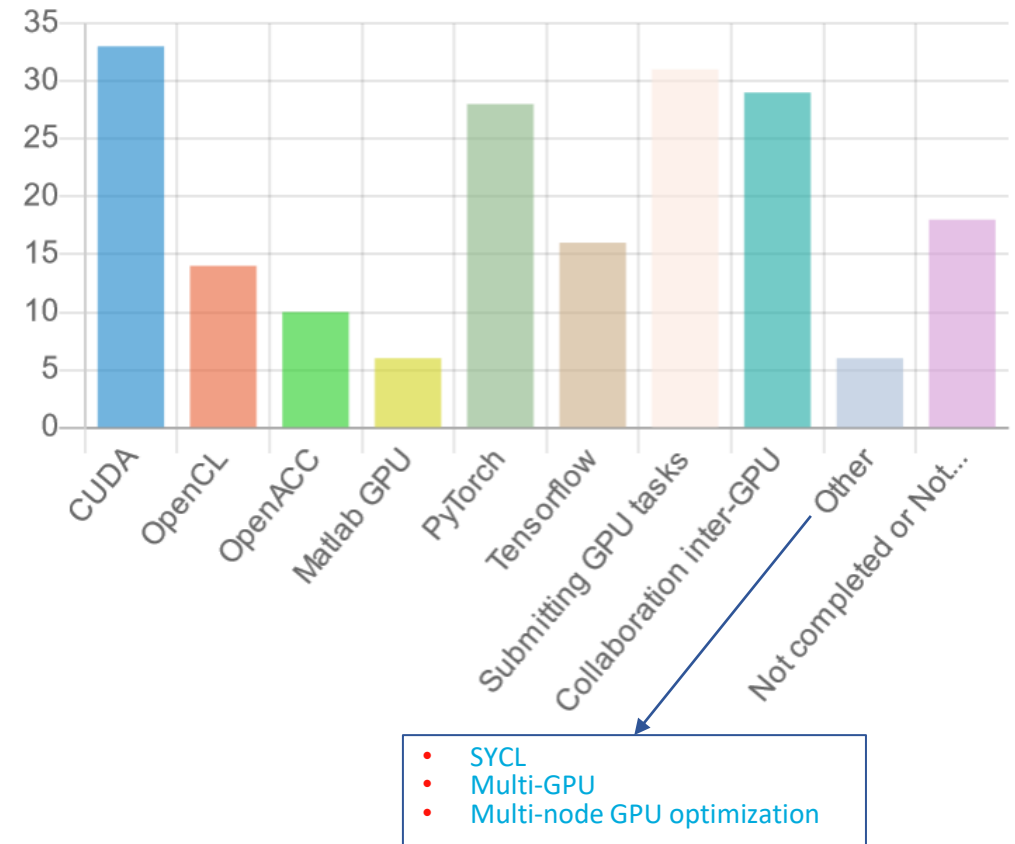


- 3x « Java »
- Perl/Lisp
- Haskell/scheme/prolog
- Mathematica

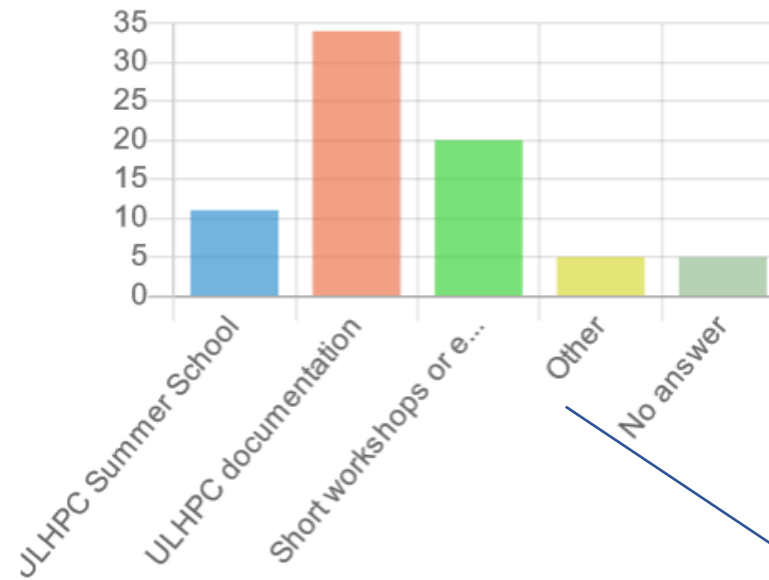
Which GPU-related skills have you already demonstrated?



What would you like to learn more ?



What learning option do you find most appealing when it comes to exploring new technologies?

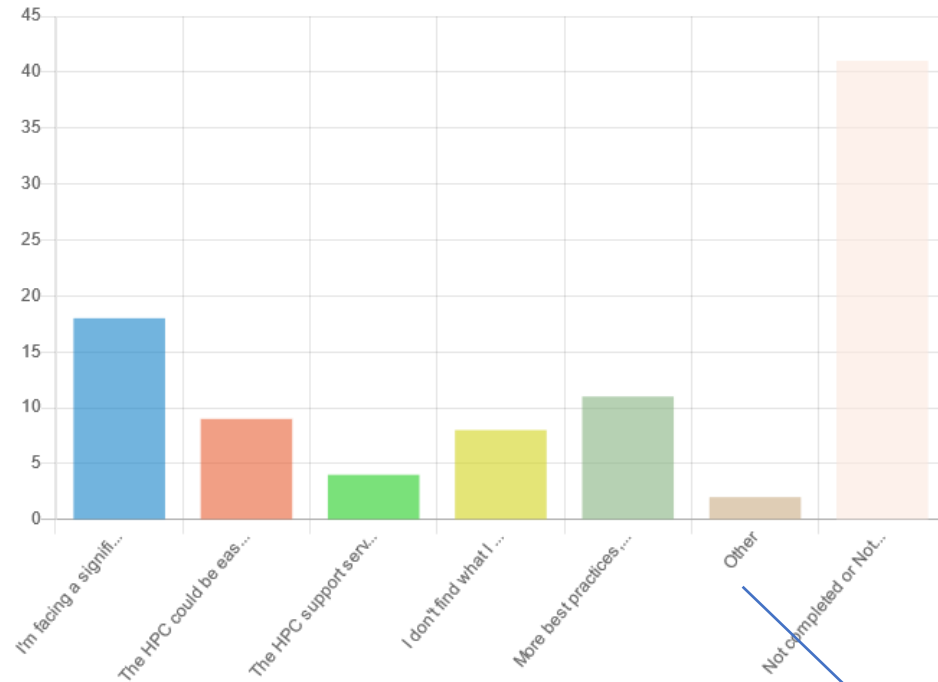









- ULHPC Summer School
- ULHPC documentation
- Short workshops or events
- Other
- No answer

• Summer School but more often

5. Conclusion & My actions

How can the HPC ecosystem be enhanced or improved?



-  I'm facing a significant delay in accessing the computing node I need due to the extended wait time in the SLURM queue.
-  The HPC could be easier to use
-  The HPC support service is not suitable
-  I don't find what I want on the UL HPC documentation
-  More best practices, training recommendations and guidance (e.g., UL HPC Summer School)
-  Other
-  Not completed or Not displayed

The support of the HPC staff is really good. For the future it would be interesting to hire IT experts to do programming work on a limited time basis for projects.

Conclusion:

- GPU are today fundamental for HPC users
 - Feels more GPUs is required
 - Feels more memory is required
 - Feels faster GPU would be great
- The future of accelerators seems:
 - Faster interconnections
 - Diversity in accelerators (IPU, FPGA, ...)
- Eager to learn GPUs
 - Find difficulties (eg memory issue) but does not open tickets

What I can do (or doing)

- Saving with workload optimization (such as Hyperband)
- For memory issue: Automatic Model Parallelism technics (GPipeTorch), GPU<->CPU callbacks
- For speed issue: Data-Parallel SGD (Horovod)
- Both: Mixture-of-Experts instead of 1 big model. Better for scaling.
- Tuto: Horovod, NCCL, GPU RDMA, ...
- IPU benchmark and doing recommandations
- Opening slack channel dedicated to GPU for research support (HLST team)
- Monitoring the userbase (sometimes what is need is not explicit, collecting data for research)

Thank you for your attention

Hungry ? 😊

- TITLE: Optimizing GPU Usage for Deep Learning Workloads: Insights and Strategies for Enhanced Efficiency
- ABSTRACT:
- Demand for GPUs in super computing has recently shown tremendous growth as the technology is spearheading advancements in AI, drug research, medical imaging, financial modeling and numerous other domains. It is therefore crucial to understand and optimize their usage for deep learning workloads.
- In response to this computational need, this talk focuses on key concepts related to GPU usage. First, we present the various types of deep learning workloads and the challenges they pose in terms of performance and programming. Second, we delve into state of the art parallelization and optimization techniques that can effectively enhance GPU usage. At last, we explore the latest hardware technologies that have the potential to significantly improve the efficiency of AI workloads.
- The insights shared in this talk are based on past experiences, recent experiments, a survey conducted among over 75 UL HPC users, and valuable information provided by the PCOG team. All of these elements will be showcased during the presentation.