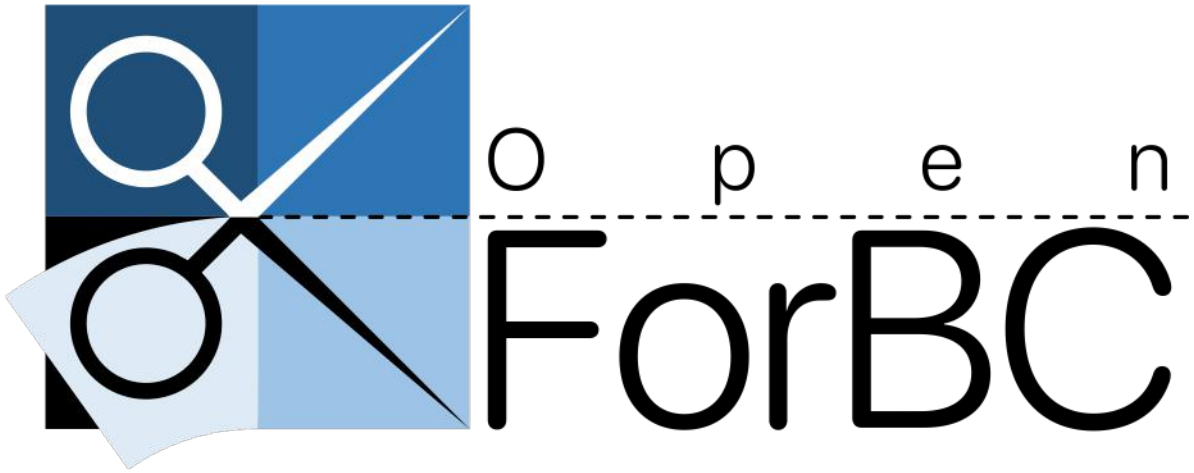
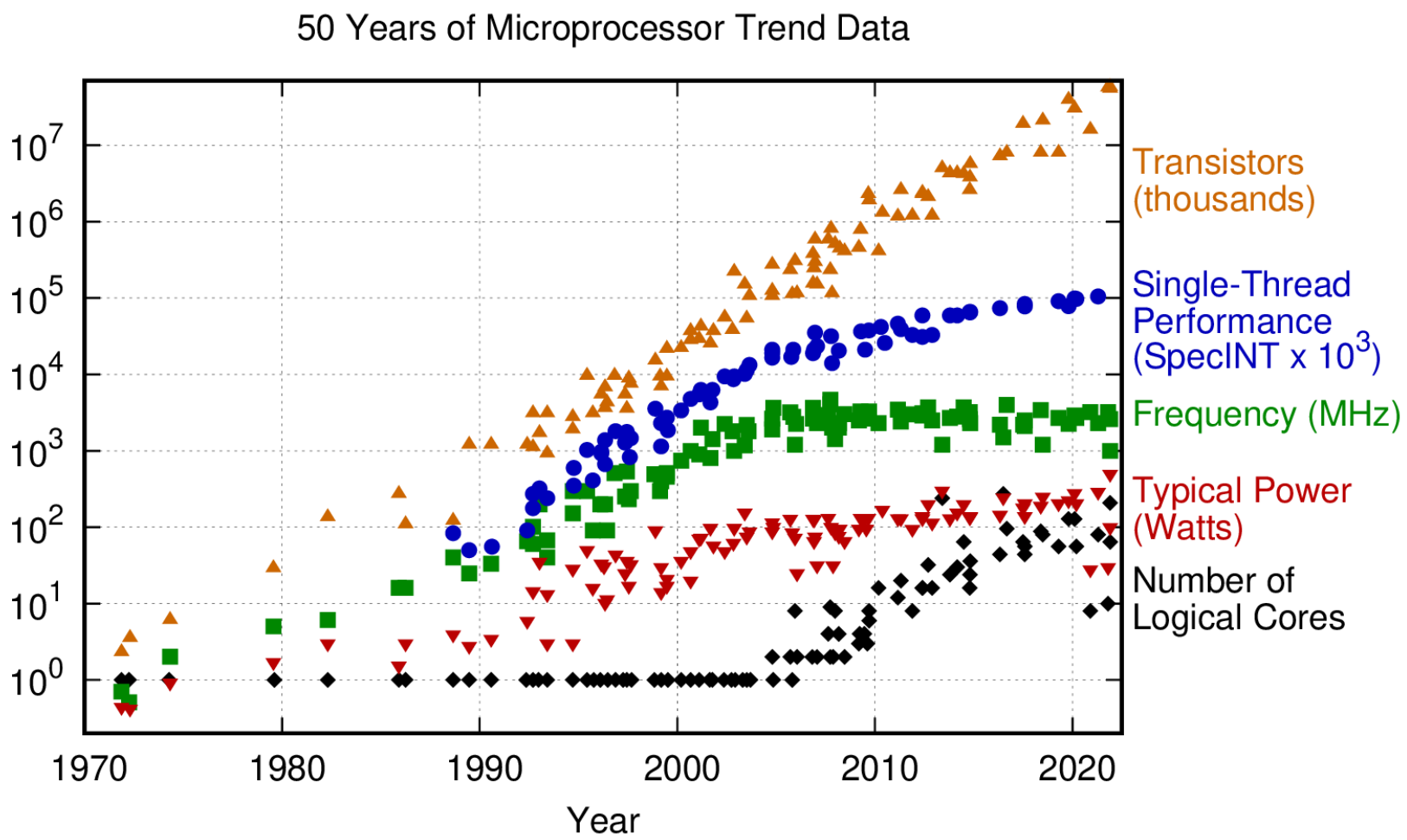


the definitive partitionable GPU interface

Alessio Borriero, **Gabriele Gaetano Fronzé**, Federica Legger and Daniele Monteleone

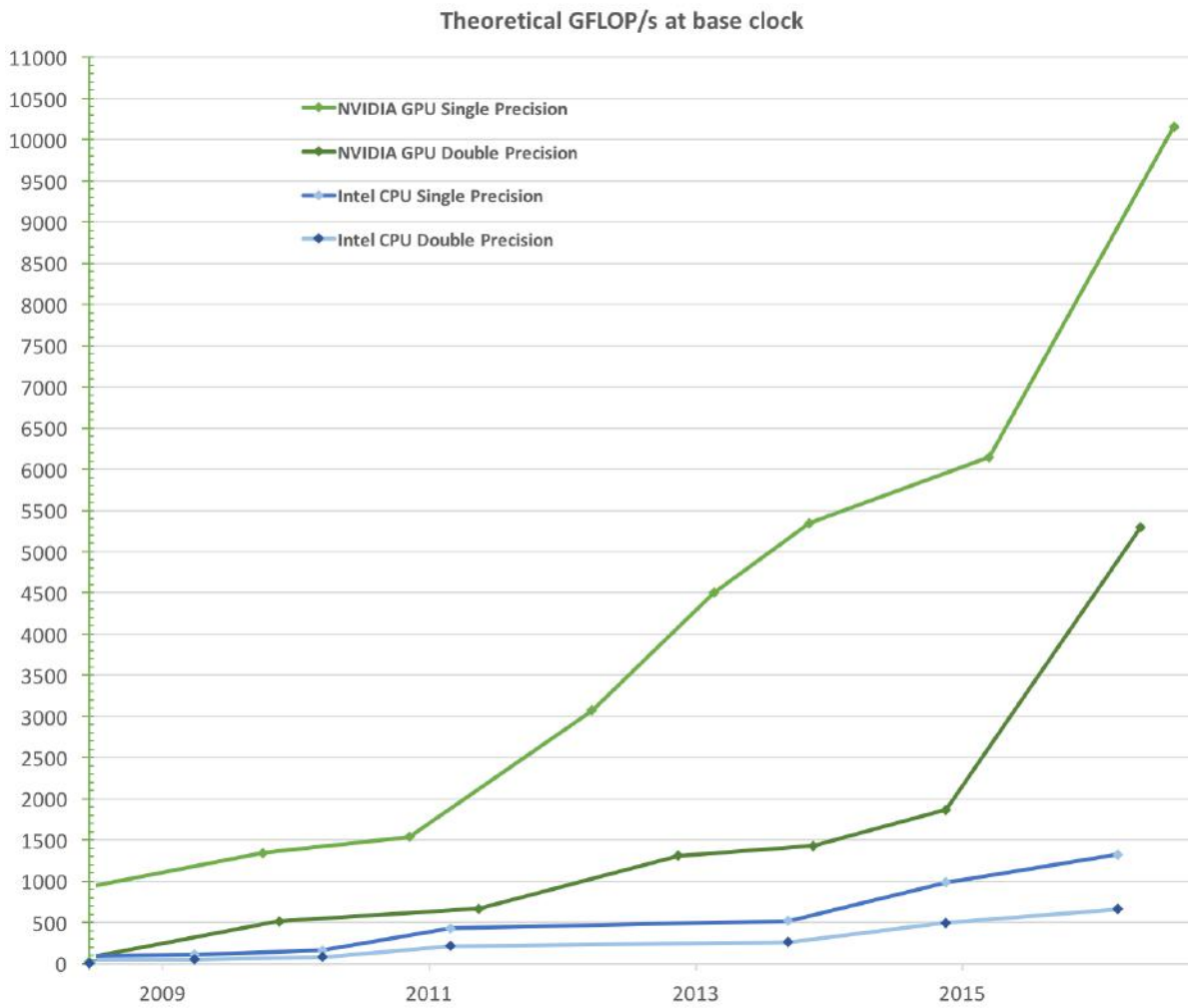
INFN (Italian National Institution for Nuclear Physics) - Research for Innovation grant - Torino

RATIONALE



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

RATIONALE

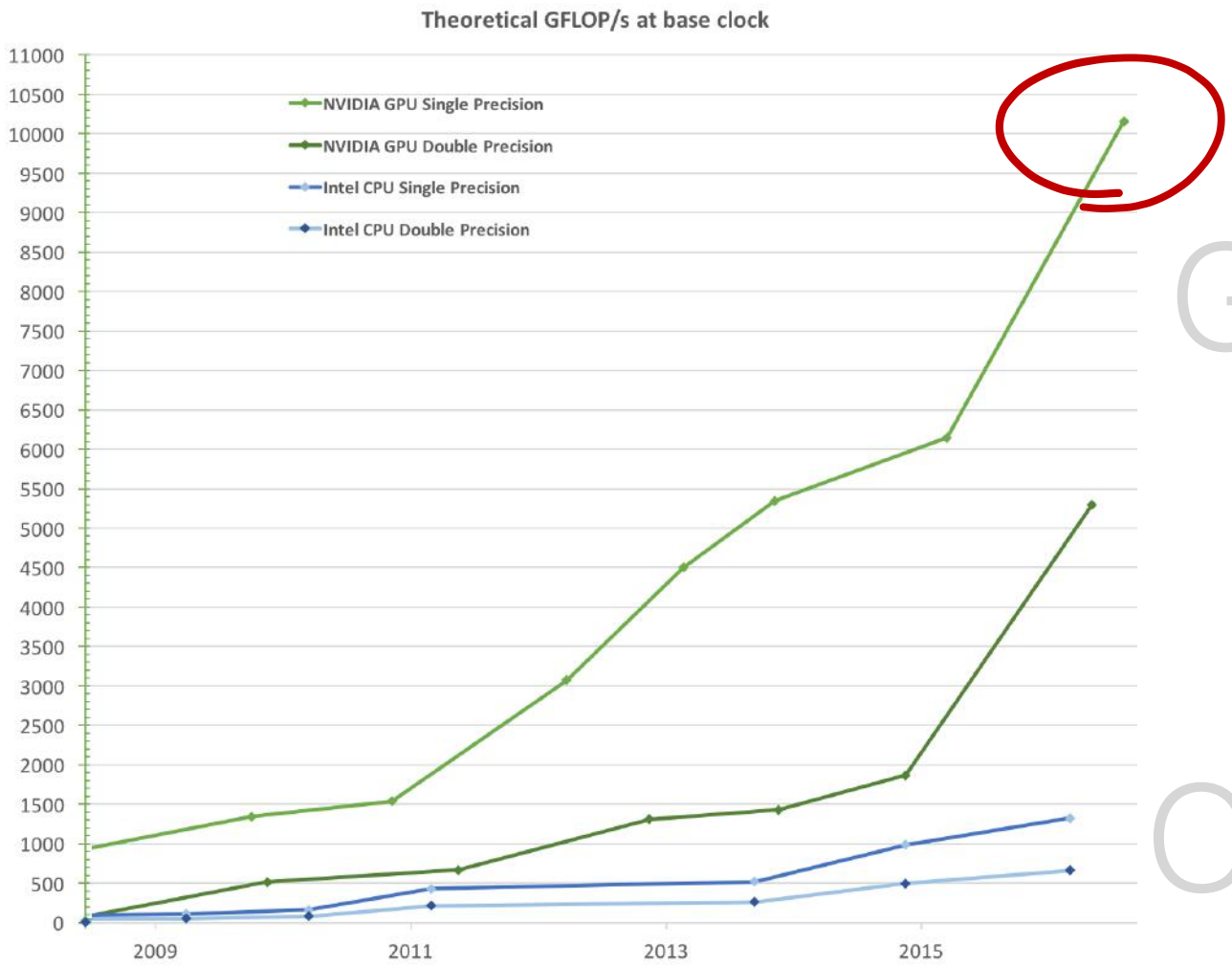


GPUs

CPUs

From Understanding GPU Architecture: Performance: GPU vs. CPU, Steve Lantz, Cornell Center for Advanced Computing

RATIONALE

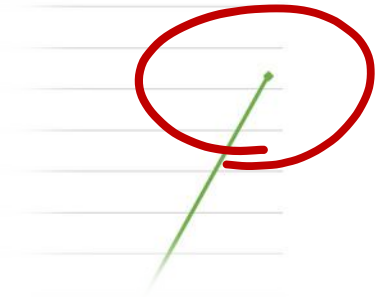


GPUs

CPUs

From Understanding GPU Architecture: Performance: GPU vs. CPU, Steve Lantz, Cornell Center for Advanced Computing

RATIONALE



Is that **too much**?

Can we **exploit** that power?

Is that **efficient**?

From Understanding GPU Architecture: Performance: GPU vs. CPU, Steve Lantz, Cornell Center for Advanced Computing

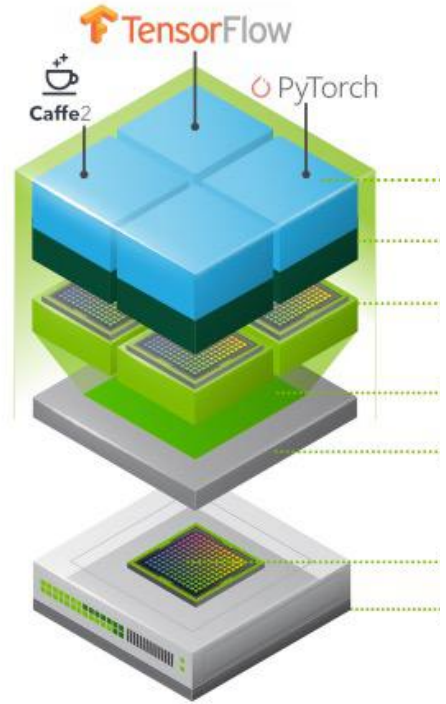
STATE OF ART

DIVIDE ET IMPERA!

- Smaller GPUs are easier to use **efficiently**
- Developing on a small GPU is **accessible**
- Some tasks are by design **inefficient on huge GPUs**
- Large GPUs can be seen as a **set of smaller ones**

STATE OF ART

DIVIDE ET IMPERA!

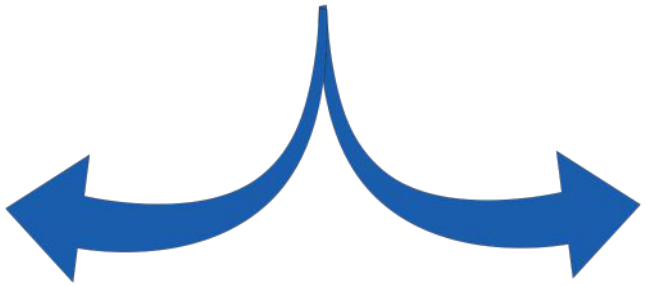


- Virtual machine or Container
- Guest driver
- GPU segment
- Host driver
- Hypervisor
- GPU
- Metal (the hardware)

STATE OF ART

PCI EXPRESS®

Single Root I/O Virtualization



NVIDIA

GRID, vGPU, MIG

closed source

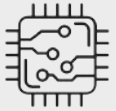
AMD

MxGPU

open source

Everyone is adopting its own "**standard**" based on the only real one

STATE OF ART

	Nvidia vGPU	Nvidia MIG	AMD MxGPU	PCIe SR-IOV
 Complete and constant API support across possible profiles e.g. all profiles support a complete set of API for compute and graphics	⚠	⚠	⚠	N/A
 P2P communications between partitions e.g. peer-to-peer communications between multiple virtual partitions for compute tasks	✅	⛔	N/A	N/A
 Free and easy licensing model e.g. the license is “included” in the hardware vs the licenses are an additional cost/require additional procedures	⛔	⚠	✅	✅
 Trivial compatibility matrix e.g. delegated to the OS flavor with no limitations wrt an equivalent physical GPU	⚠	⚠	✅	✅
 Certified on any physically compatible host system e.g. the compatibility is based on low-enough standards, therefore any physically and electrically supporting hardware works	⛔	⛔	✅	✅

Both within a vendor’s techs or between all vendors!

STATE OF ART

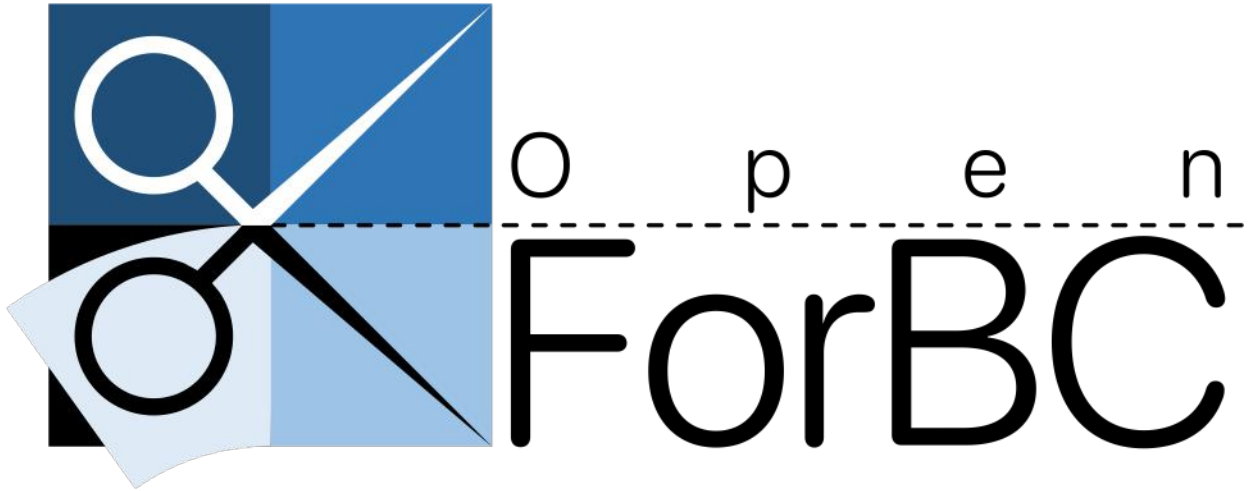
 Same underlying **boilerplate**

 Same ideal **operations** and **procedures**

 **Different** top level **semantics**

Proprietary “standards” only partially supported in Linux

Both within a vendor’s techs or between all vendors!



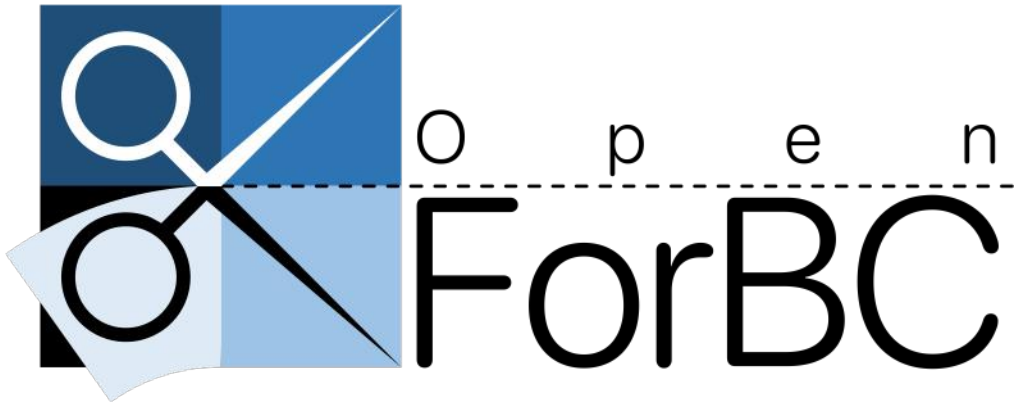
The independent inititive to make
GPU partitioning:

as easy as using a pair of scissors

vendor independent

mainstream

OpenForBC

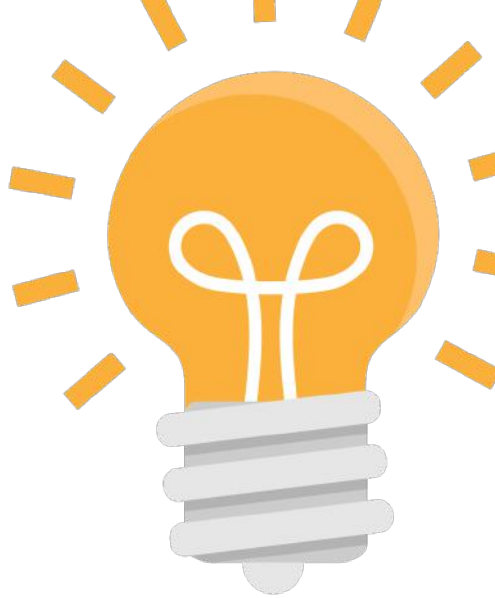


= Open scissors in 🇮🇹!

stands for

Open For Better Computing

OpenForBC



PCI EXPRESS®

Single Root I/O Virtualization

NVIDIA®

GRID, vGPU, MIG

AMD

MxGPU

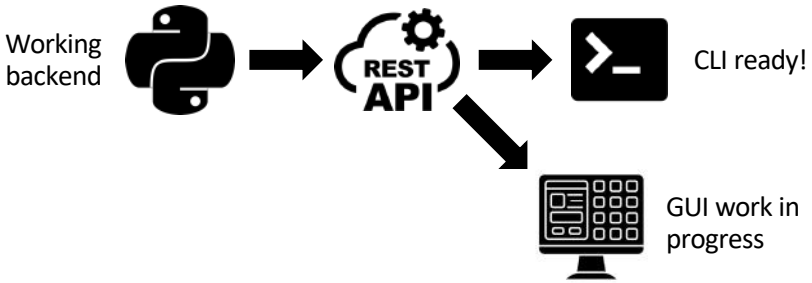
Open For Better Computing

Open ForBC

- **Uniform interface** for GPU partitioning
- Expandable toolset for **future new technologies**
- **No vendor specificity**
- Improved **Linux Compatibility**

openforbc

Open ForBC



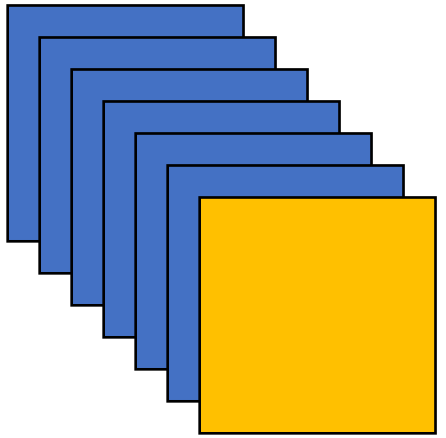

```
gpu list
gpu types
gpu partition create
gpu partition list
gpu partition get
```

openforbc

Open ForBC

gpu list

Lists the available physical GPUs compatible with any partitioning technology

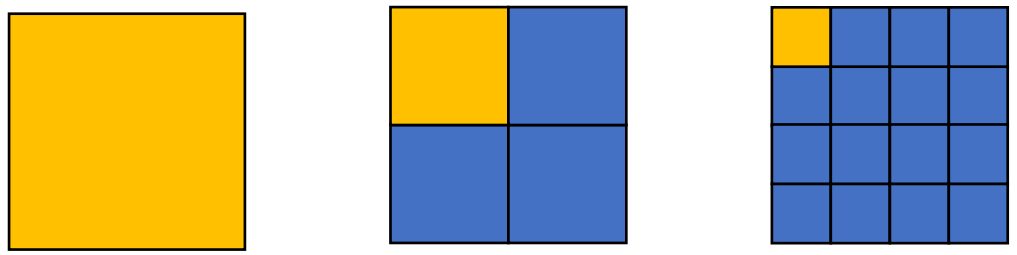


openforbc

Open ForBC

gpu types

Lists the available virtual GPU profiles, with plenty of information about memory size (WIP: peak performance).



1/1 GPU

1/4 GPUs

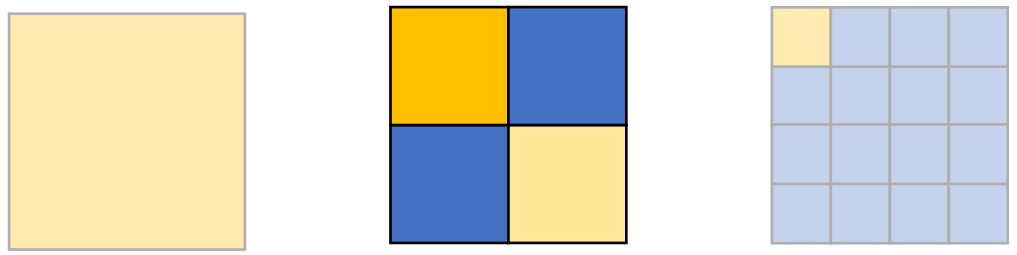
1/16 GPUs

openforbc

Open ForBC

gpu types list -c

Lists the **creatable** virtual GPU profiles, handling the complex compatibility matrices automatically.



1/1 GPU

1/4 GPUs

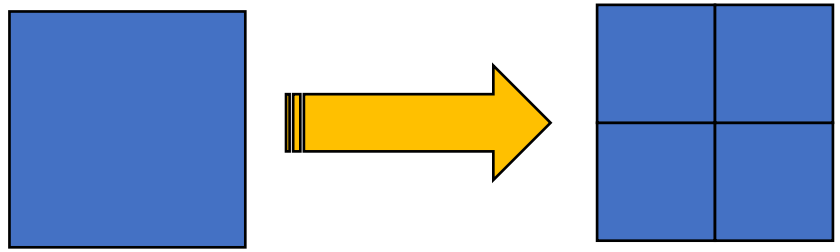
1/16 GPUs

openforbc

Open ForBC

gpu partition create

Applies one of the available profiles performing the required procedures and ensuring not to disrupt existing instances, providing understandable errors in case of wrong selection



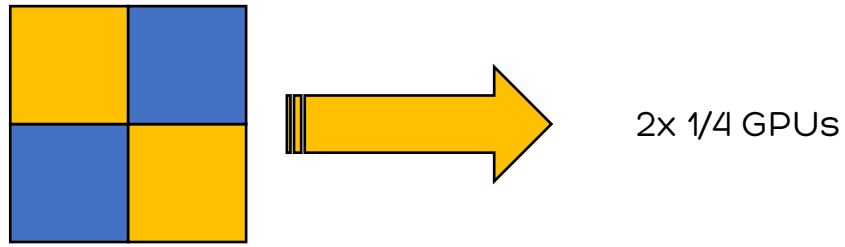
gpu partition destroy

openforbc

Open ForBC

gpu partition list

Returns the mode currently set up on the GPU.

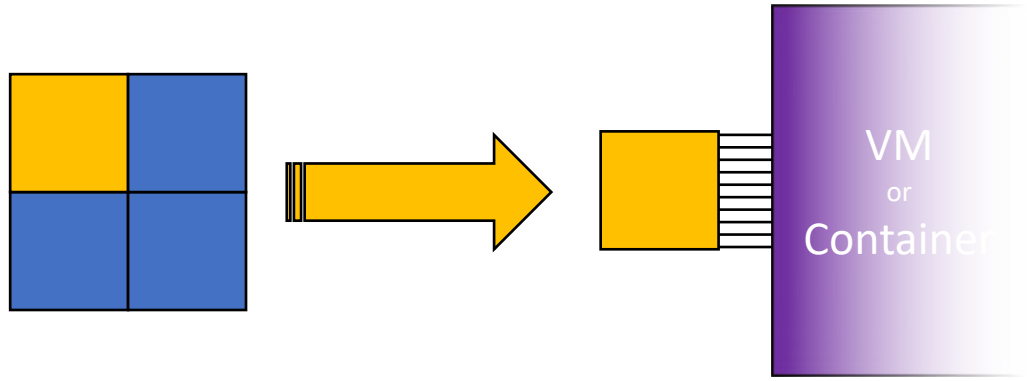


openforbc

Open ForBC

gpu partition get

Retrieves the information needed to instanciate a new VM or container attached to the virtual GPU instance.



OpenForBC

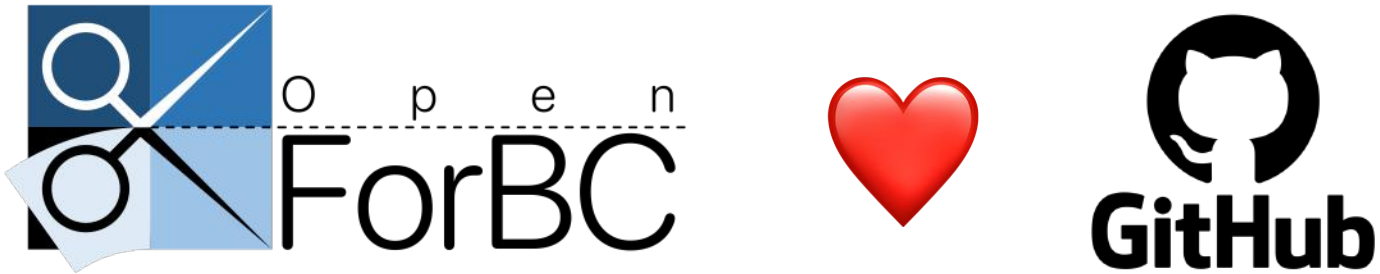

⌥⌘3
fish /home/monteleo/openforbc
$ openforbc gpu list
[nvidia:a100-0] 54c2f5e1-6865-3a7b-93c9-3a6e051ac3f0: NVIDIA A100-PCIE-40GB
$ openforbc gpu -i nvidia:a100-0 types -c
468: GRID A100-4C (4.0GiB)
469: GRID A100-5C (5.0GiB)
470: GRID A100-8C (8.0GiB)
471: GRID A100-10C (10.0GiB)
472: GRID A100-20C (20.0GiB)
473: GRID A100-40C (40.0GiB)
$ openforbc gpu -i nvidia:a100-0 partition create 471
f74efc9f-d5ea-46db-bf00-ab0a15ecee88
$ openforbc gpu -i nvidia:a100-0 partition get f74efc9f-d5ea-46db-bf00-ab0a15ecee88
NOTE: please ensure that PCI domain:bus:slot.function is not already used.
<hostdev mode='subsystem' type='mdev' managed='no' model='vfio-pci' display='on'>
<source>
<address uuid='f74efc9f-d5ea-46db-bf00-ab0a15ecee88'/>
</source>
<address type='pci' domain='0x0000' bus='0x00' slot='0x10' function='0x0'/>
</hostdev>
$ openforbc gpu -i nvidia:a100-0 partition destroy f74efc9f-d5ea-46db-bf00-ab0a15ecee88
$

OpenForBC

One simple line to install it:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Open-ForBC/OpenForBC/install.sh)"
```

Every line is open source and hosted on GitHub!



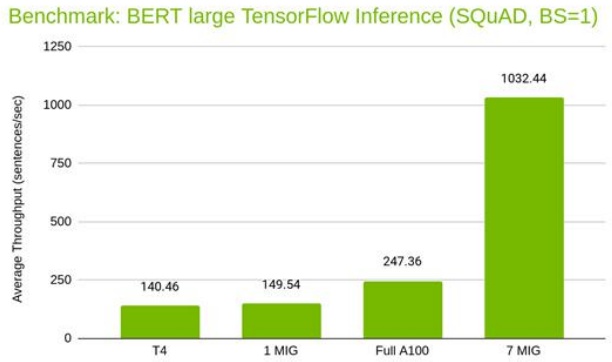
https://github.com/Open-ForBC/OpenForBC/tree/main

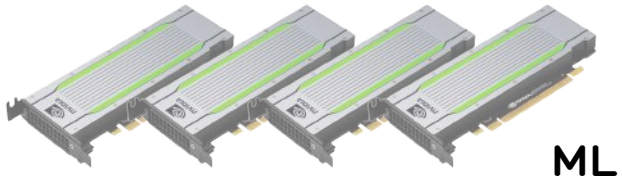
Tested and working on our A100-40GB PCIe

Working on RHEL and RHEL-equivalent. More models to be tested as soon as we get them!

Hardware choice:
Nvidia A100 40GB GPU

ML training:
whole GPU with lots of memory





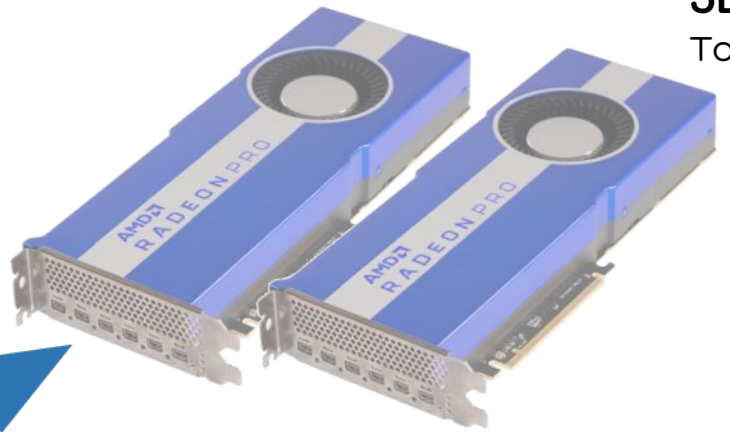
ML inference:
smaller GPUs for higher throughput

Planned in our roadmap

We got from AMD an S7150x2 to start thinkering with.
Looking forward for v320/v340/v520/v540

Hardware choice:
AMD Radeon PRO V340

3D tasks:
Top of the line vGPU instances



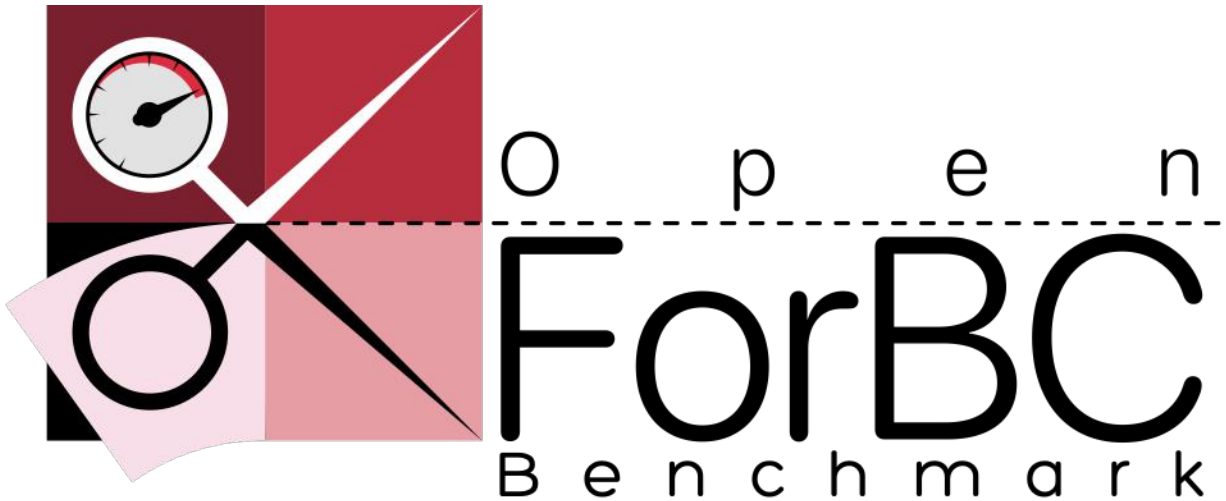
Up to 128 users per server
Down to 1/6 power consuption

2D tasks:
Up to 32 vGPUs (32 users!)

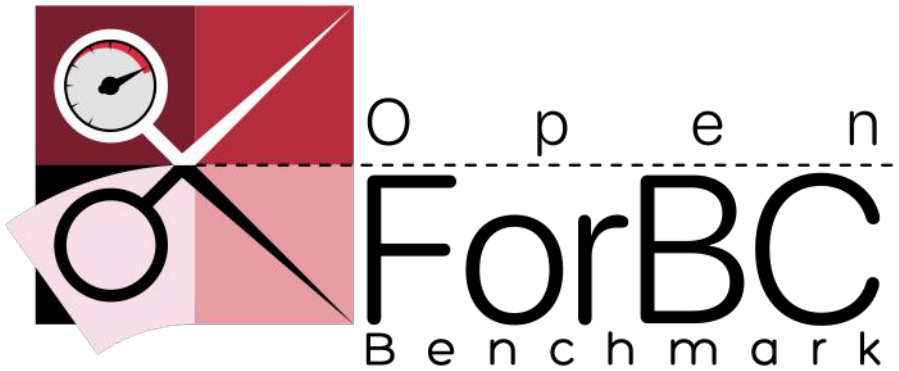
BENCHMARKS

Is GPU partitioning actually worth it?

We've created a tool to answer this question



BENCHMARKS



Our modular benchmark suite

It includes our **custom benchmarks**

Is compatible with **Phoronics benchmarks**

Is easily **expandable** with additional benchmark definitions

BENCHMARKS

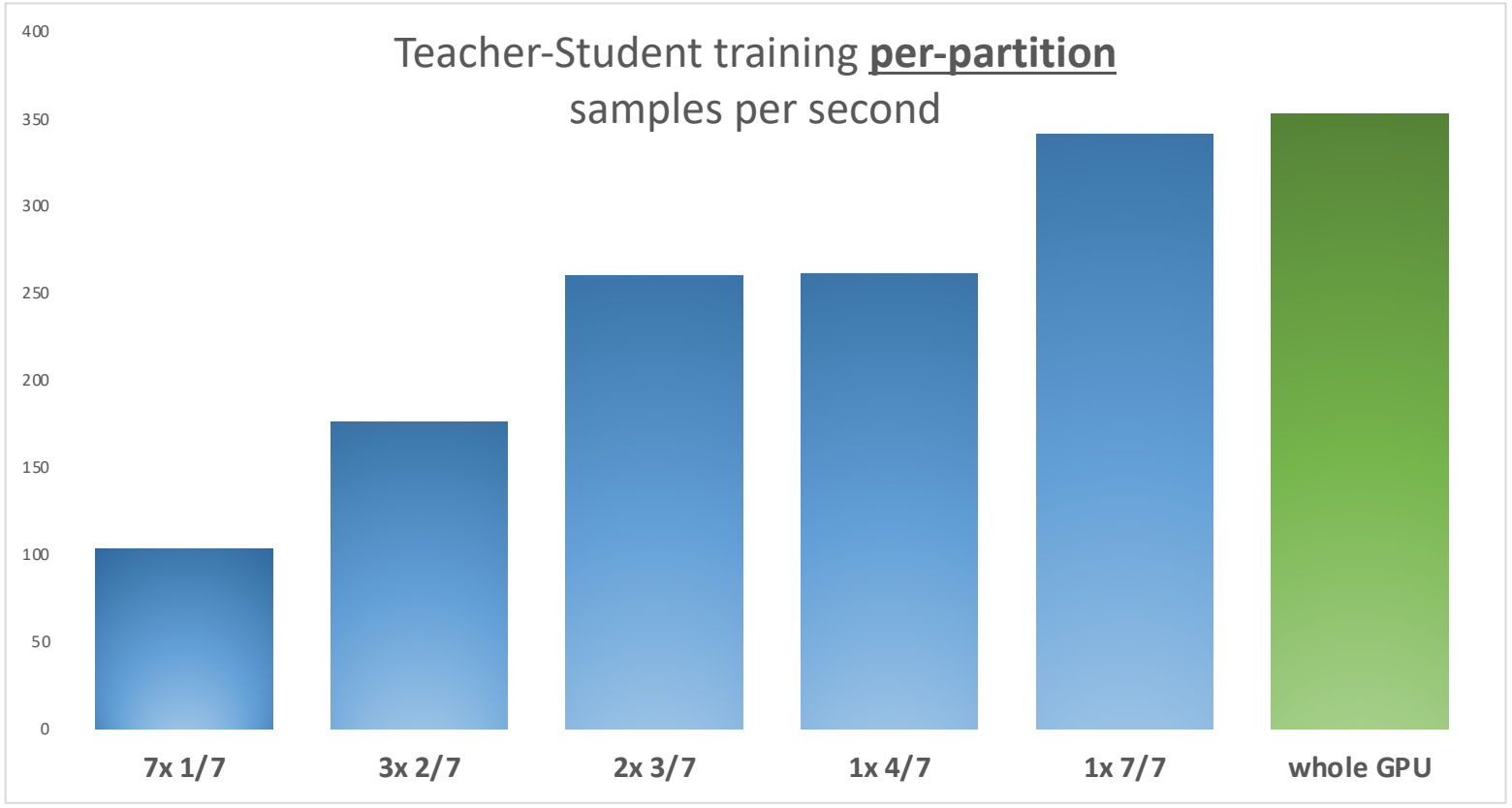
Every line is open source and hosted on GitHub!

Feel free to submit a PR with your own benchmarks!

https://github.com/Open-ForBC/OpenForBC-Benchmark

BENCHMARKS

Let's look at a typical ML task...



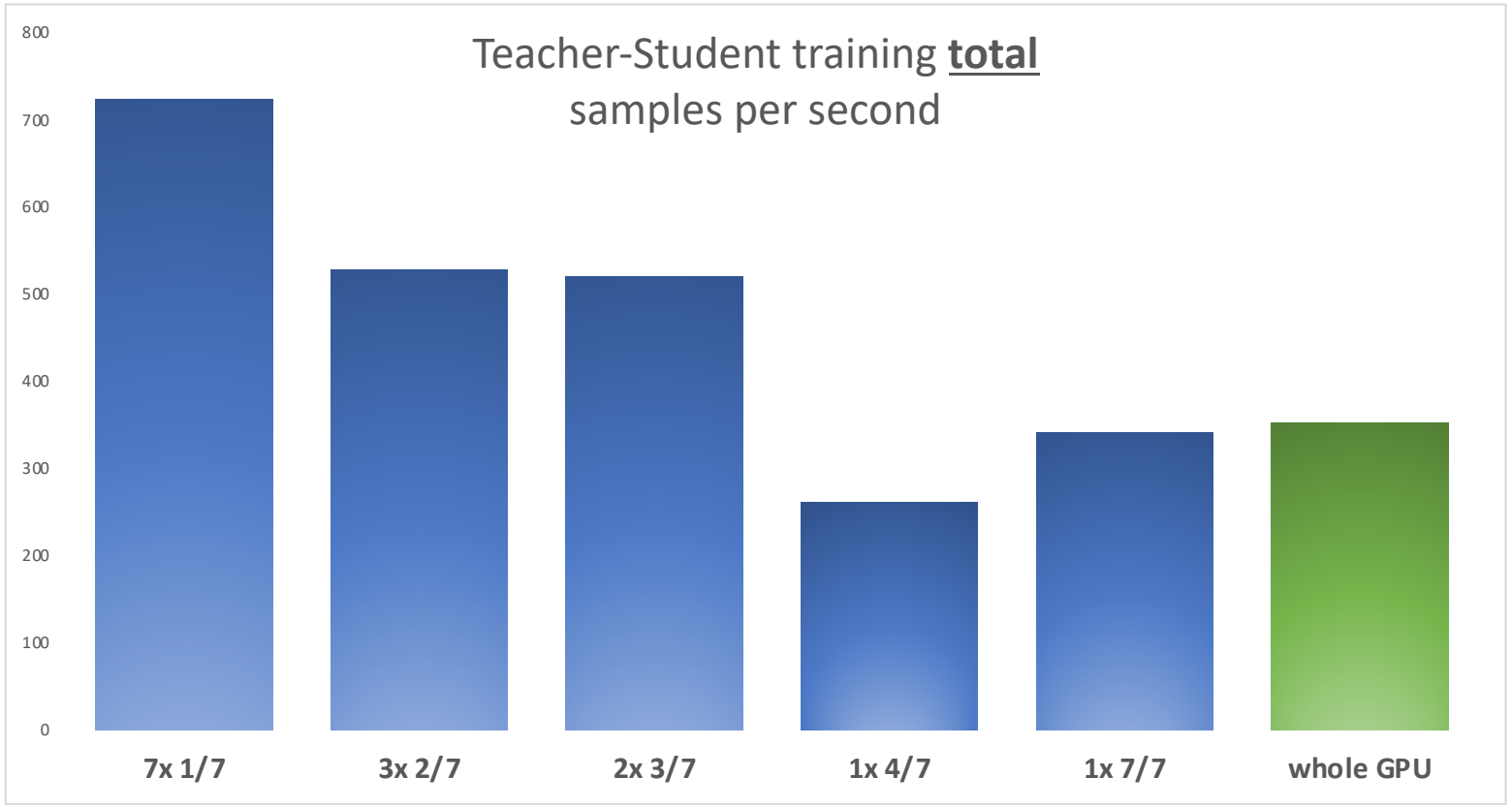
In green the peak throughput of an A100-40GB PCIe.

In blue the average peak throughput of all the creatable* partitions for a given profile.

* All creatable partitions have been allocated and loaded with computation.

BENCHMARKS

...and rescale the peak performance



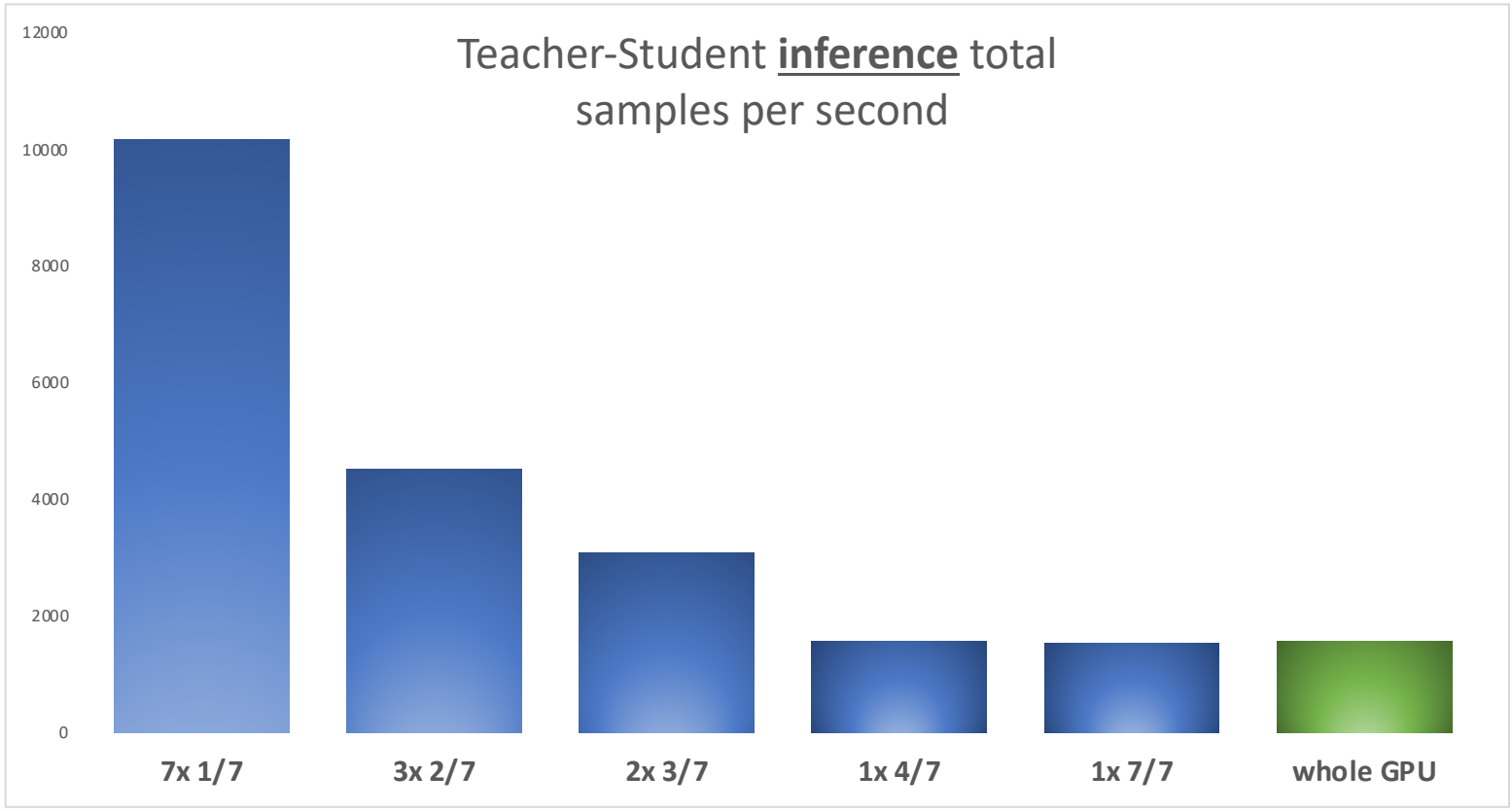
In green the peak throughput of an A100-40GB PCIe.

In blue the peak throughput of each partition topology, computed as the sum of the average throughput of all creatable* partitions given a specific profile.

Maximum speedup of 105%.

* All creatable partitions have been allocated and loaded with computation.

What happens with inference?



In blue the peak throughput of each partition topology, computed as the sum of the average throughput of all creatable* partitions given a specific profile.

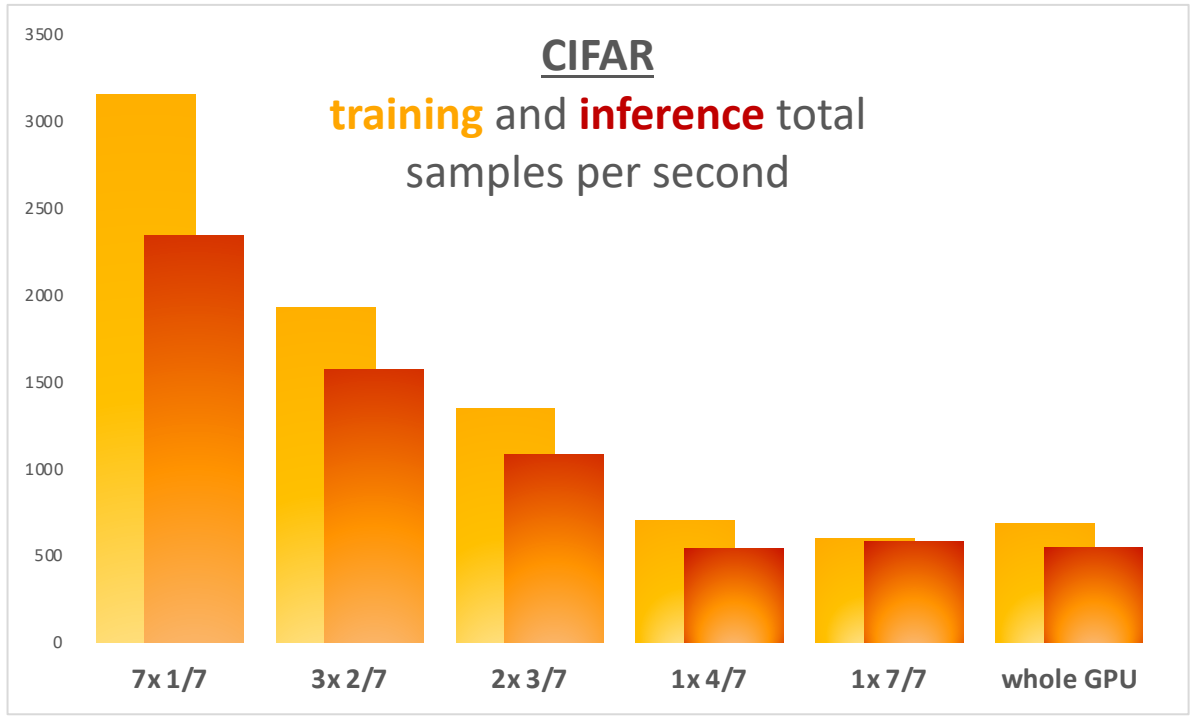
Maximum speedup of 655%.

GPU power consuption merely rises from 130W to 225W.

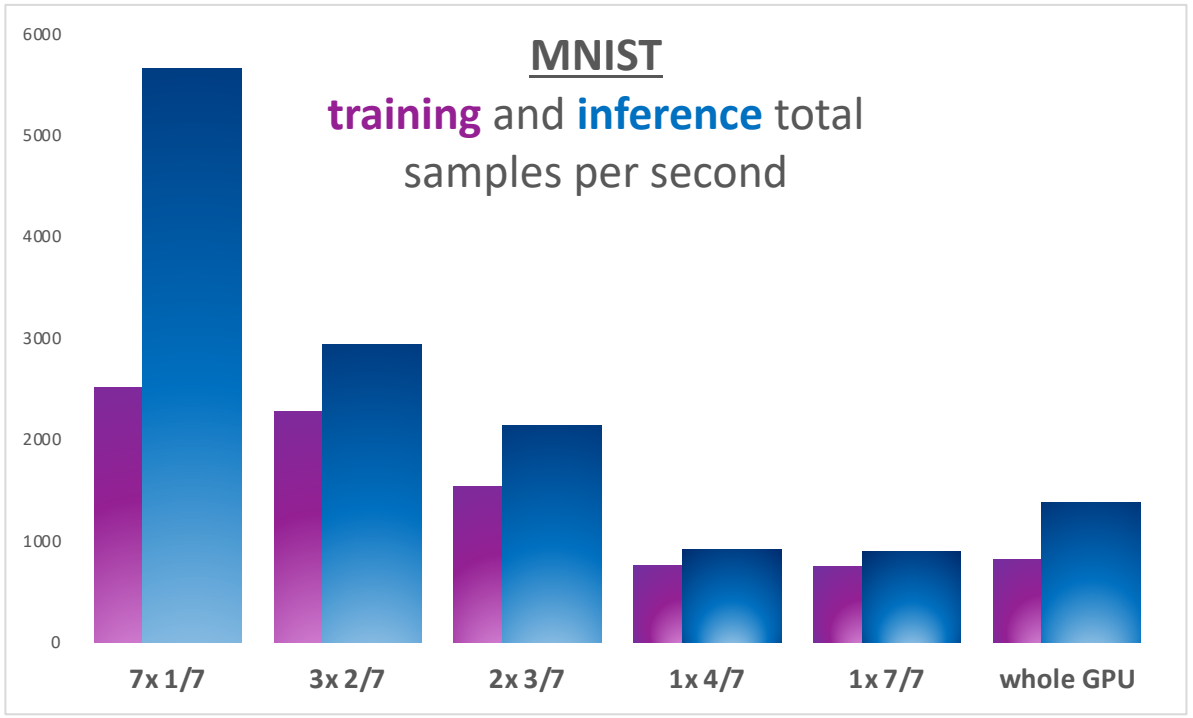
* All creatable partitions have been allocated and loaded with computation.

BENCHMARKS

More examples:



Speedup 461% and 428%



Speedup 305% and 409%

Partners and supporters

WHO ARE WE

OpenForBC is an INFN's Research for Innovation 2021 grant

Federica Legger
Senior researcher – PhD in Physics

Gabriele Gateano Fronzé
Senior researcher – PhD in Physics

Alessio Borriero
Physics Master's Student

Daniele Monteleone
Computer Engeneering Bachelor's Student

And the whole INFN Torino Computing Group

OpenForBC

GPU partitioning shows clear advantages:
huge speedups and reduced relative power consuption

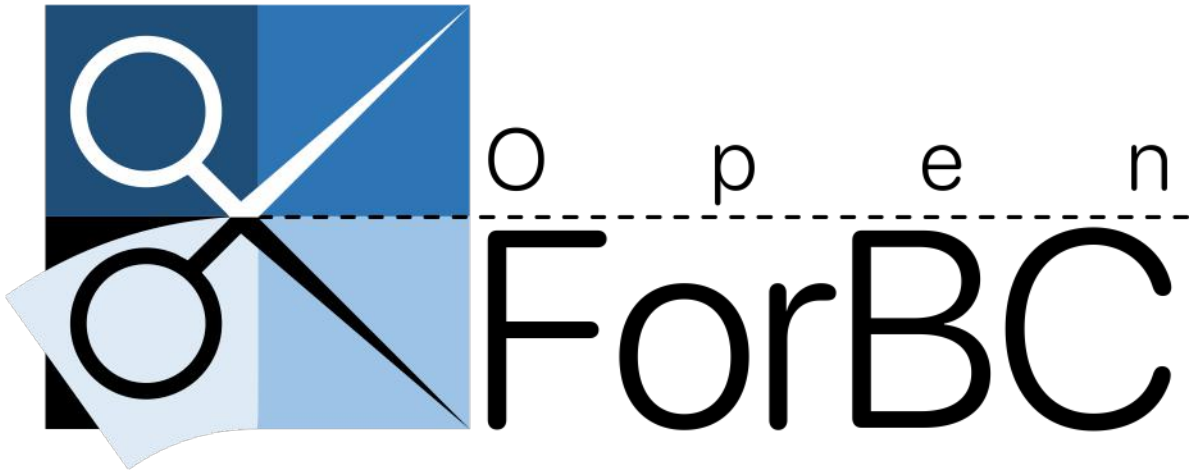
Open ForBC introduces a smarter way to use partitionable GPUs on Linux KVM.

Open ForBC handles the complexity due to vendors' choices providing users with a powerful yet simple toolset.

Open ForBC is open source, provides a CLI and a REST API, supports Nvidia GPUs (MIG and vGPU) with AMD support coming next.

Open ForBC Benchmark is an expandable modular benchmark framework.

INFN
Istituto Nazionale di Fisica Nucleare



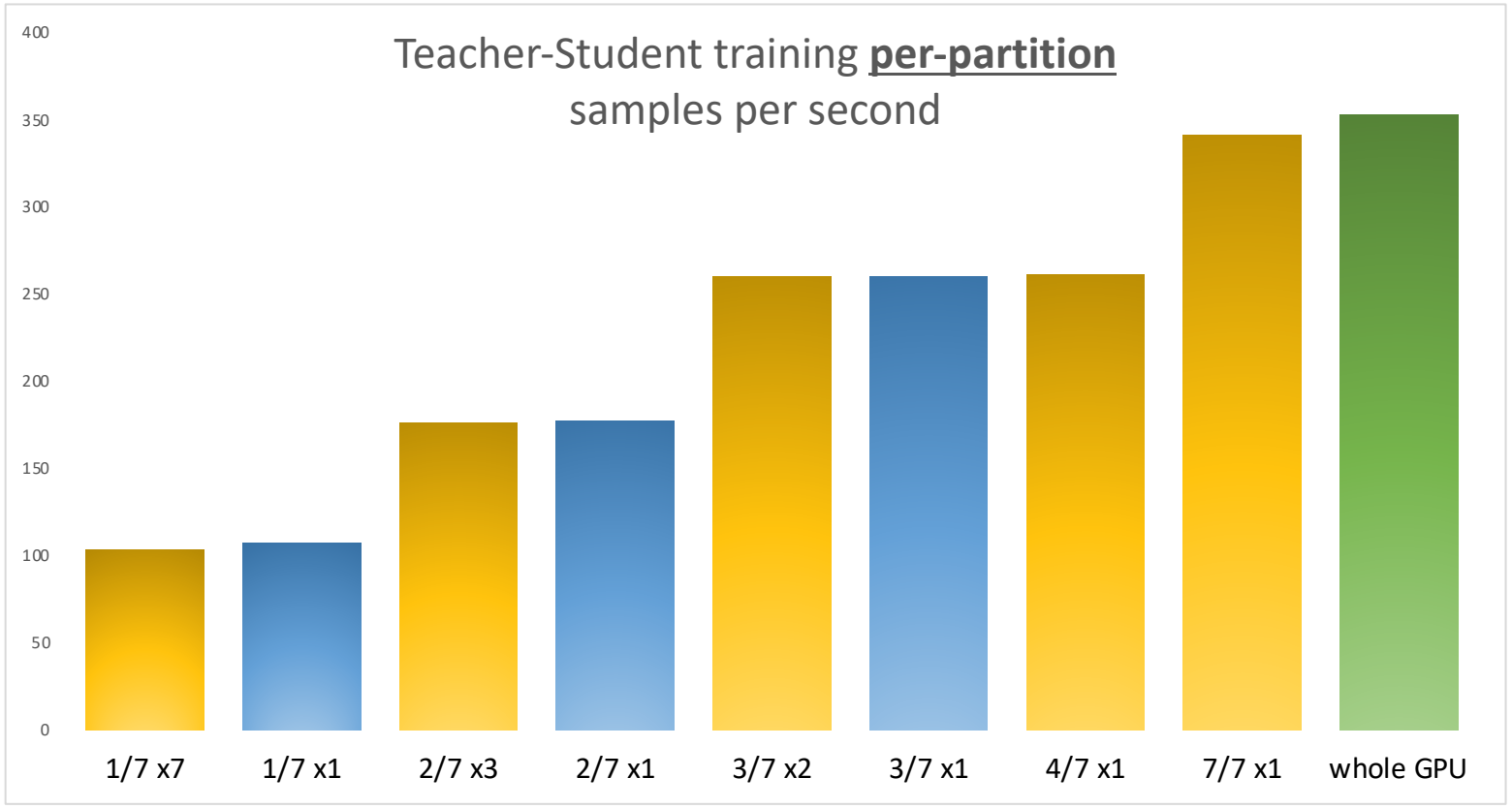
cut it apart!

Thank you for your attention!

BACKUP

BENCHMARKS

Let's look at a typical ML task...



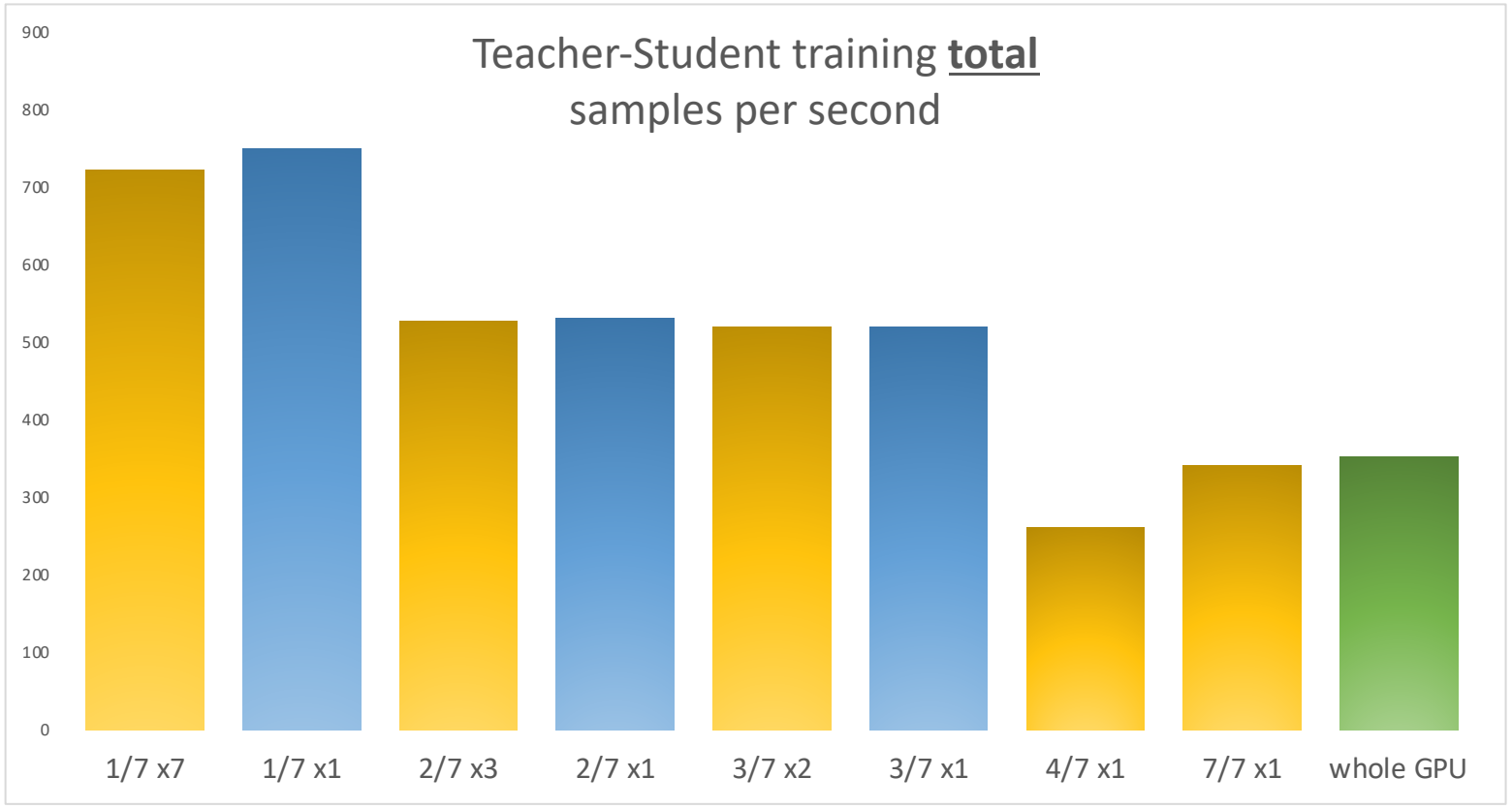
In green the peak performances of an A100-40GB PCIe

In yellow the peak performance of one of N partitions packing 1/N of the whole GPU resources, loaded simultaneously.

In blue the peak performance of a single partition packing 1/N of the whole GPU resources, without other running partitions.

BENCHMARKS

...and rescale the peak performance

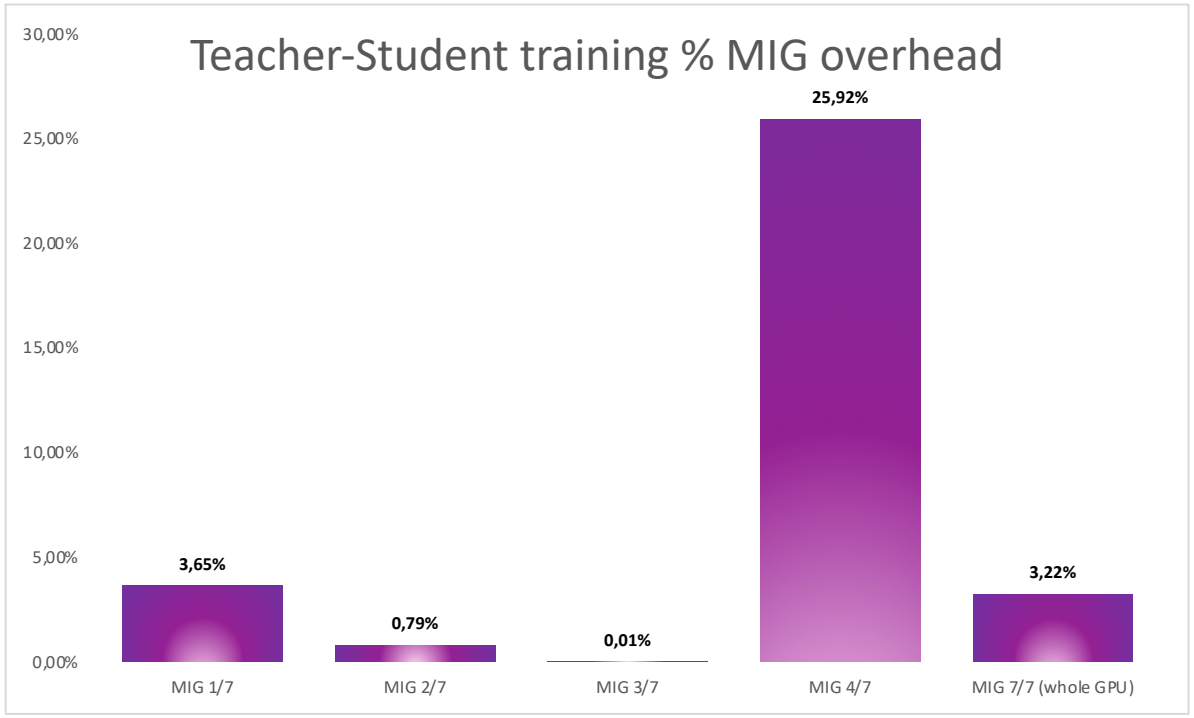


Same color code as before.

The peak performance of each partition topology has been rescaled by a factor equal to the maximum number of partitons of such kind runnable on a single physical GPU.

Seven 1/7 partitions reach up to 105% speedup with respect to a single A100-40GB PCIe.

Overhead evaluation



Overhead computed as difference between the peak performance of a partitioning topology when all N partitions are loaded simultaneously and when only one is loaded.

This measures the overhead caused by internal scheduling and handling of partitions in the worst case scenario.

Maximum overhead is around 3.7%, with the 4/7 case being caused by the impossibility of allocating more than one 4/7 partition.