

OpenForBC, the GPU partitioning framework



Federica Legger

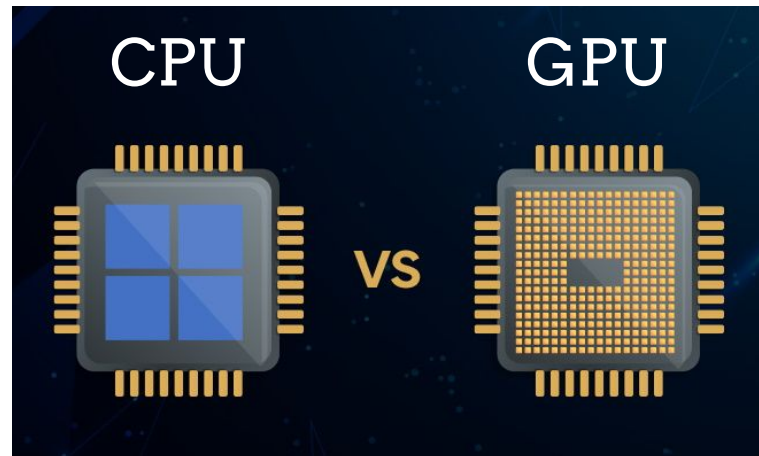
Alessio Borriero, Daniele Monteleone, Gabriele Gaetano Fronzé,
Sara Vallero, Stefano Bagnasco, Stefano Lusso



Istituto Nazionale di Fisica Nucleare

GPU: what?

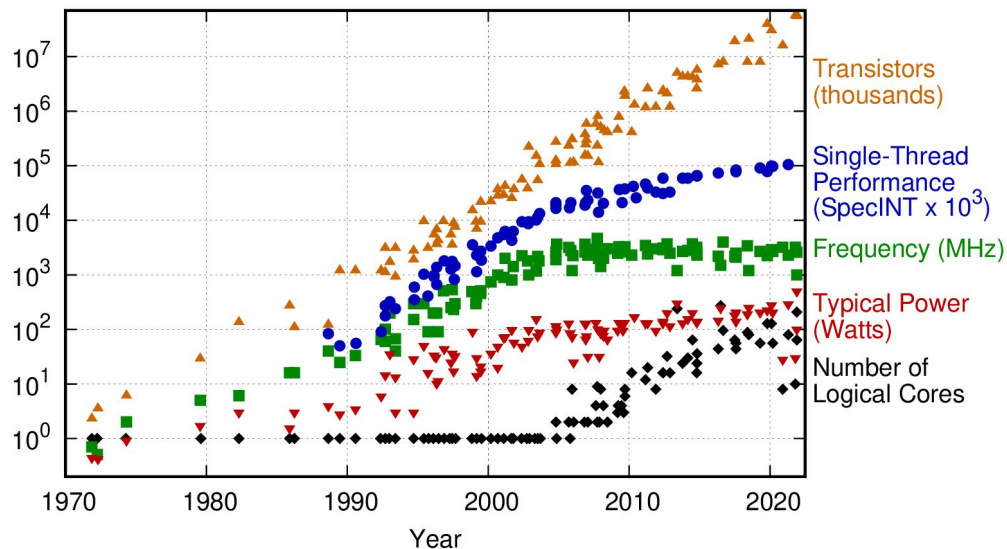
- CPU - Central Processing Unit
- GPU - Graphical Processing Unit
- Intensive computations may be offloaded to GPU from CPU
- Needs design and implementation of efficient data-parallel algorithms



- AI and data science
- Data Center and Cloud computing
- Design and Virtualization
- Edge computing
- High performance computing
- Self Driving vehicles

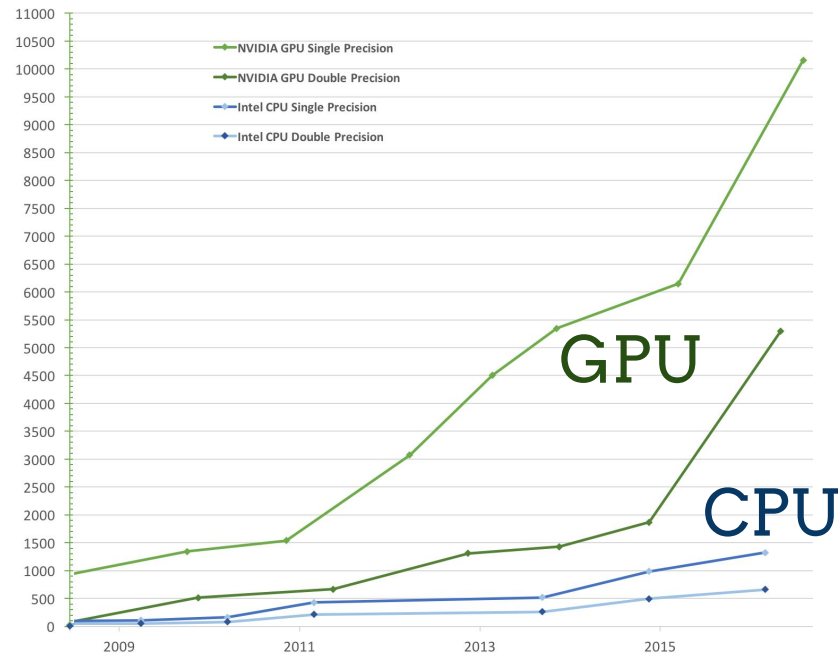
GPU: why?

50 Years of Microprocessor Trend Data



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

Theoretical GFLOP/s at base clock



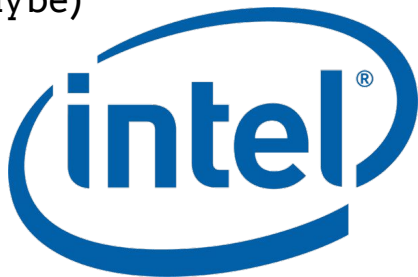
GPU: who?



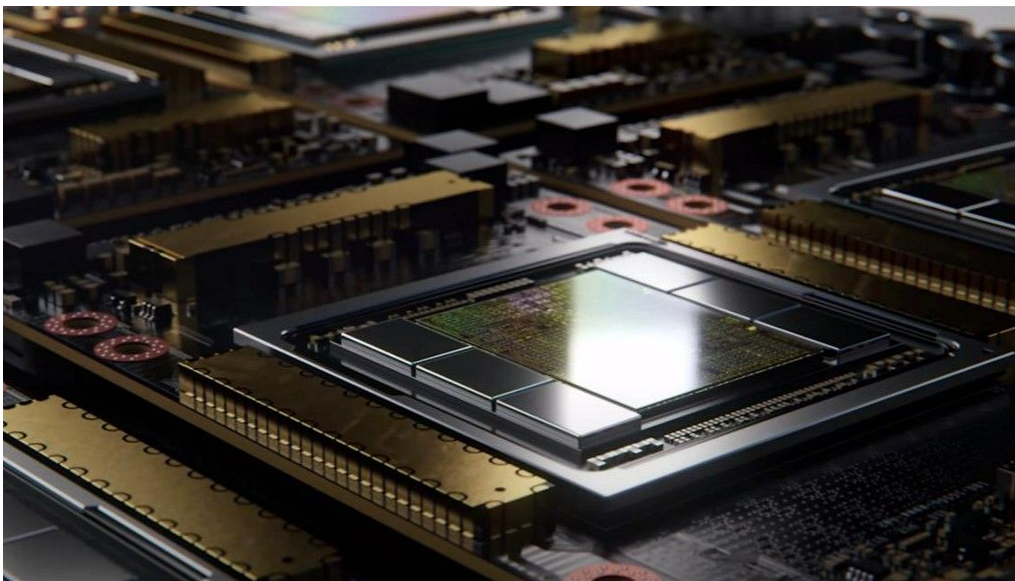
NVIDIA®



(Maybe)

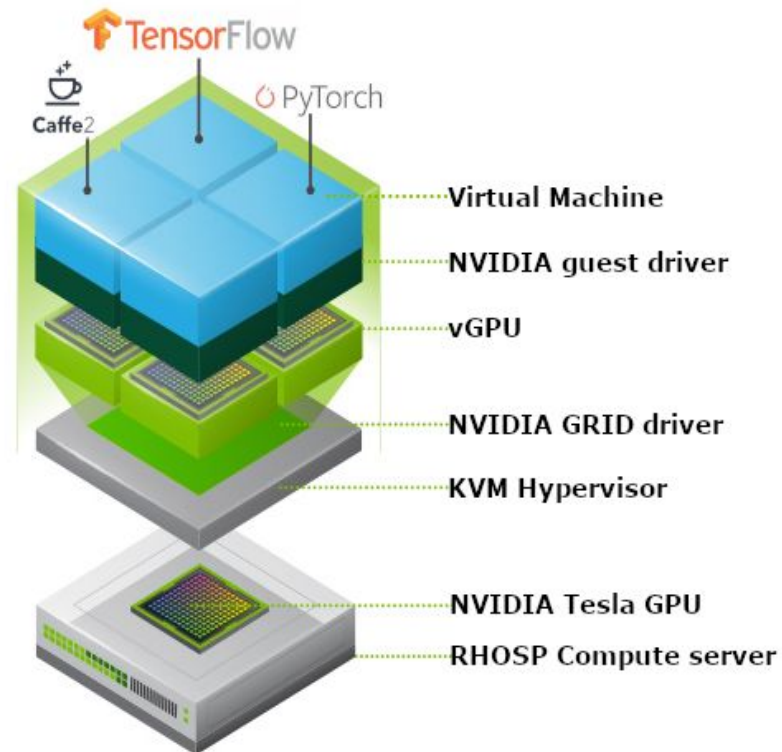


- CUDA (nVidia)
- ROCm (AMD)
- OpenCL SDK (all)

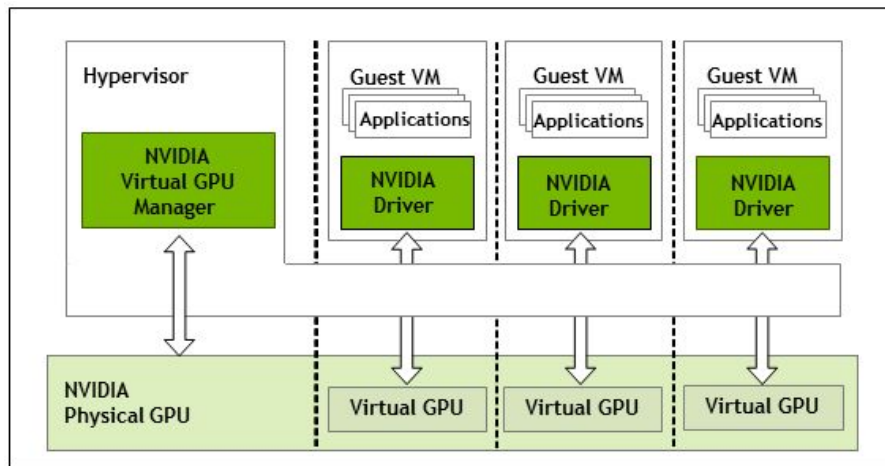


GPU: how?

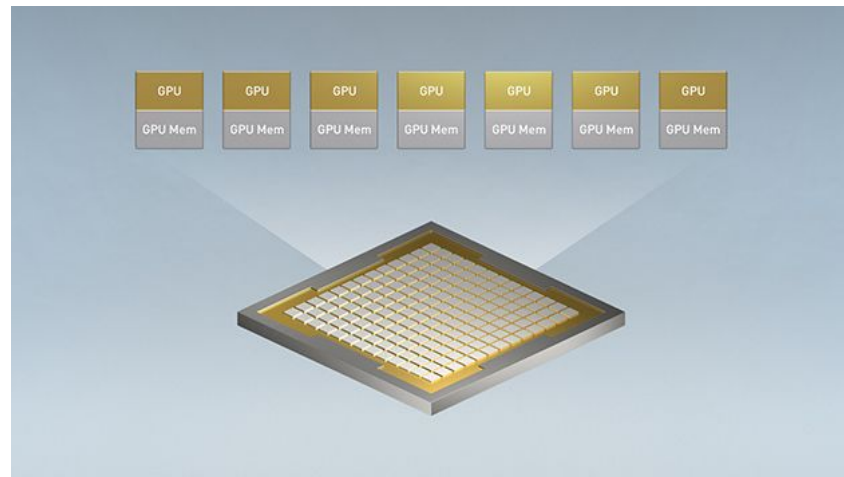
- Modern GPU extremely powerful:
 - FLOPS, memory -> expensive!
- **GPU partitioning!**
 - Not all workflows require 100% GPU resources
 - Share GPU with other users and/or applications

























- Temporal partitioning: **vGPU**
 - On NVIDIA A100 (40 GB) up to 10 vGPUs with 4 GB memory allocated per VM
- Spatial partitioning: **MIG**
 - Up to 7 fully isolated instances with 5 GB memory each on an A100



vGPU



MIG

		Nvidia VGPU	Nvidia MIG	AMD MxGPU	PCIe SR-IOV
	Full API support across profiles complete set of API for compute and graphics				N/A
	P2P communications between partitions connects multiple virtual partitions for computing			N/A	N/A
	Free and easy licensing model license included or requires additional costs/procedures				
	Trivial compatibility matrix delegated to OS with no limitations wrt an equivalent physical GPU				
	Certified on any compatible host system Compatible with any physically and electrically supporting hardware				

- GPU partitioning technologies are based on one underlying standard: **single root input/output virtualization (SR-IOV)**, a specification that allows the isolation of PCI Express resources for manageability and performance reasons

PCI 
EXPRESS[®]

Single Root I/O Virtualization



GRID, vGPU, MIG



MxGPU

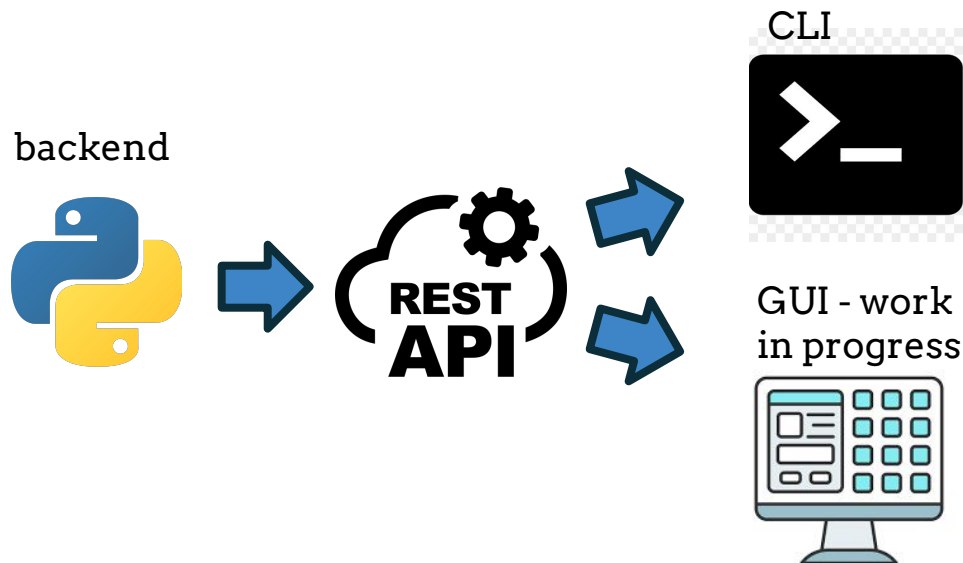
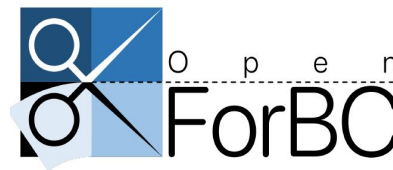


- **Open For Better Computing:** uniform interface for GPU partitioning
 - Same underlying boilerplate (SR-IOV)
 - Same operations and procedures to partition GPUs from different vendors
 - Expandable toolset for future new technologies
 - No vendor specificity
 - Improved Linux Compatibility



Winner of 2021 R4I (Research For Innovation) INFN grant for technology transfer

OpenForBC: how?



- > gpu list
- > gpu types
- > gpu partition create
- > gpu partition list
- > gpu partition get



<https://github.com/Open-ForBC/OpenForBC>

1. > openforbc gpu list

```
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)

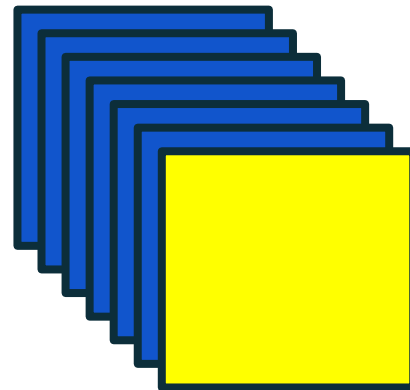
$ openforbc gpu -i nvidia:a100-0 partition create 471
f74efc9f-d5ea-46db-bf00-ab0a15ecee88

$ 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
$
```

- Lists the available physical GPUs compatible with any partitioning technology



2. > openforbc gpu types -c

```
fish /home/monteleo/openforbc

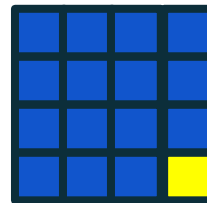
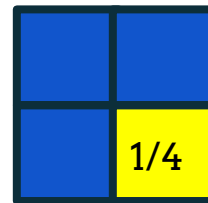
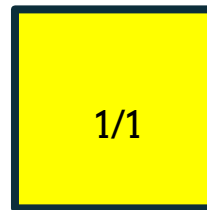
$ openforbc gpu list
[nvidia:a100-0] 54c2f5e1-6865-3a7b-93c9-3a6e051ac3f0: NVIDIA_A100-PCIE-40GB

$ 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
$
```

- Lists the **creatable** virtual GPU profiles



1/16

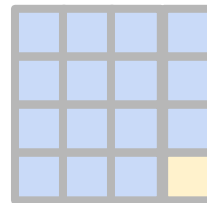
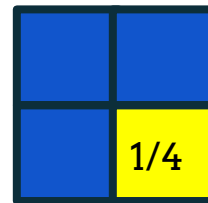
3. > openforbc gpu partition create

```
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='0x00' />
</hostdev>
$ openforbc gpu -i nvidia:a100-0 partition destroy f74efc9f-d5ea-46db-bf00-ab0a15ecee88
$
```

- Creates one of the available profiles



1/16

4. > openforbc gpu partition get

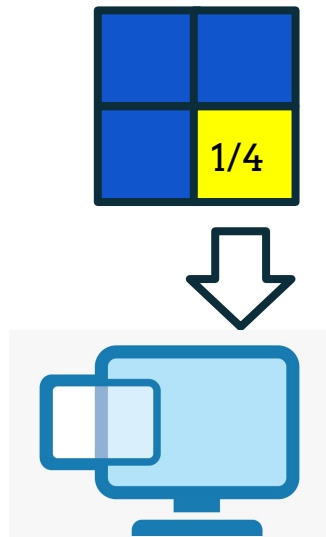
```
fish /home/monteleo/openforbc

$ 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.

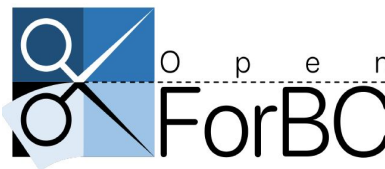
<hostdev mode='subsystem' type='mdev' managed='no' model='vfio-pci' display='on'>
  <source>
    <address uuid='f74efc9f-d5ea-46db-bf00-ab0a15ecee88' />
  </source>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x10' function='0x0' />
</hostdev>

$ openforbc gpu -i nvidia:a100-0 partition destroy f74efc9f-d5ea-46db-bf00-ab0a15ecee88
$
```

- Retrieves the info needed to attach the virtual GPU instance to a VM



5. > openforbc gpu partition destroy



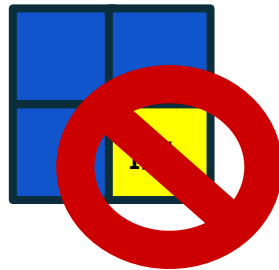
```
fish /home/monteleo/openforbc

$ 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='0x00' />
</hostdev>

$ openforbc gpu -i nvidia:a100-0 partition destroy f74efc9f-d5ea-46db-bf00-ab0a15ecee88
```

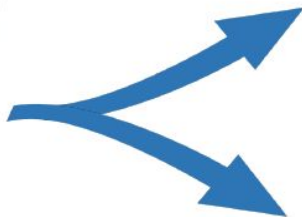
- Destroys the virtual GPU profile



Is it really a good idea?

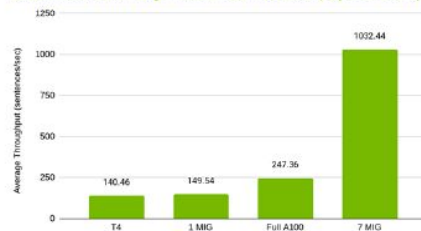


Hardware choice:
Nvidia V100 32GB GPU



ML training:
huge GPU with lots of memory

Benchmark: BERT large TensorFlow Inference (SQuAD, BS=1)



ML inference:
smaller GPUs for higher throughput

- GPU partitioning for workloads that do not fully saturate the GPU
- Test OpenForBC overhead

OpenForBC Benchmark



- modular benchmark suite for GPUs
 - Agnostic to GPU partitioning
 - Benchmarks may also run on CPU
 - includes our own custom benchmarks
 - compatible with Phoronics benchmarks
 - easily expandable with additional benchmark definitions
- Python codebase
 - run benchmark from CLI
 - automatic logging of test results

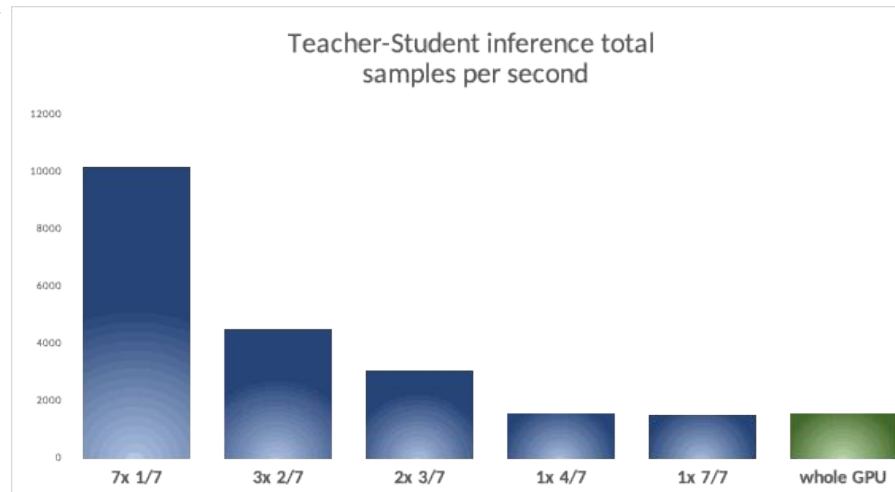
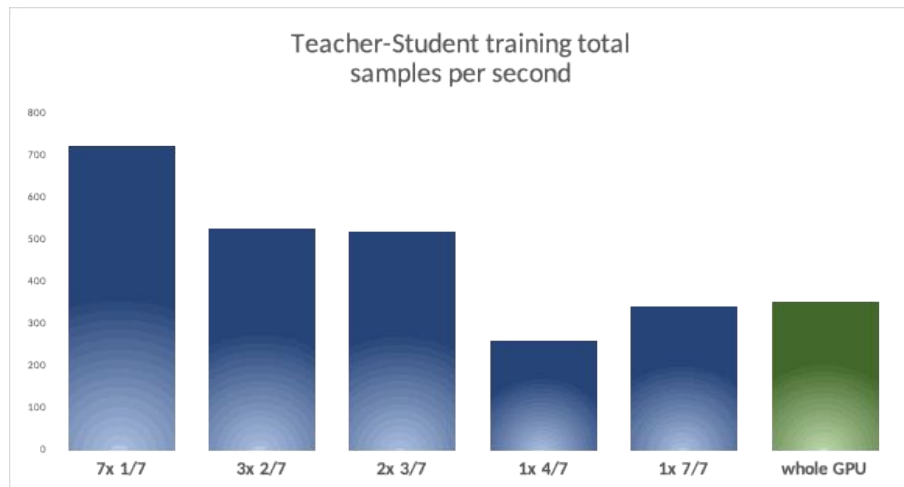
```
1. bash
bash #1 ssh #2
admin@h-34 /Users/admin/Work/OpenForBC/OpenForBC-Benchmark > o4bc-bench benchmark list
CIFAR_realtime_benchmark
TeacherStudent_realtime_benchmark
MNIST_FCNeuralNetwork
matmulCpp_benchmark
dummy_benchmark
matmul_benchmark
dummy_py_benchmark
MNIST_realtime_benchmark
admin@h-34 /Users/admin/Work/OpenForBC/OpenForBC-Benchmark > o4bc-bench benchmark run matmulCpp_benchmark
Running "matmulCpp_benchmark" setup commands
$ python3 -m venv .venv
(venv) $ ./setup.sh
(venv) $ chmod +x bin/matmulCppExe
Running "matmulCpp_benchmark" preset "matrix_20x30"
(venv) $ bin/matmulCppExe 20 30
Matrix multiplication time: 0.000054 s
Preset      Stat      Value
-----
matrix_20x30 matmul_time_s 5.4e-05
admin@h-34 /Users/admin/Work/OpenForBC/OpenForBC-Benchmark >
```



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



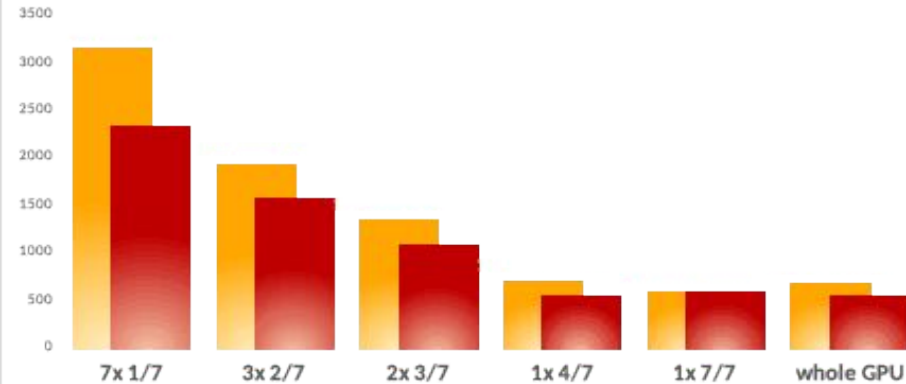
Teacher-Student ML Benchmark



- GPU power consumption merely rises from 130W to 225W
- peak throughput computed as the sum of the average throughput of all creatable partitions given a specific profile
- All creatable partitions have been allocated and loaded with computation

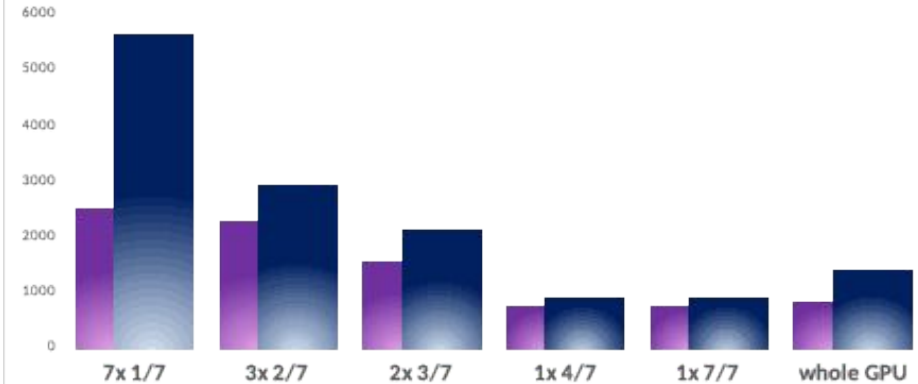
CIFAR and MNIST ML benchmarks

CIFAR
training and inference total
samples per second



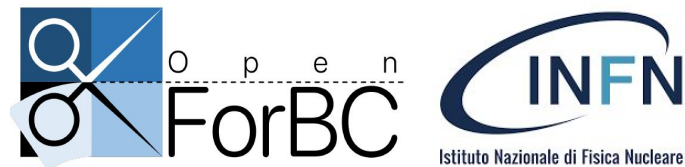
CNN for image recognition on CIFAR dataset

MNIST
training and inference total
samples per second



FFNN for hand-writing recognition on MNIST dataset

OpenForBC: who?



Federica Legger
Technologist INFN



Gabriele Gaetano Fronzé
UniTo Post-doc grant



Alessio Borriero
INFN Student grant



Daniele Monteleone
INFN Student grant

Sponsors



- GPU partitioning allows for more efficient resource utilisation
 - Reduced power consumption
 - Huge speedups for specific workloads
- OpenForBC makes it easy to use partitionable GPUs on Linux KVM
 - Simple toolset, open source, CLI and REST API
 - Tested with Nvidia GPUs
 - AMD support coming next
- OpenForBC Benchmark is an expandable modular benchmark framework for GPUs
 - Ready-to-run benchmarks
 - Easy to add your own benchmarks