



UNIVERSITAT
JAUME I



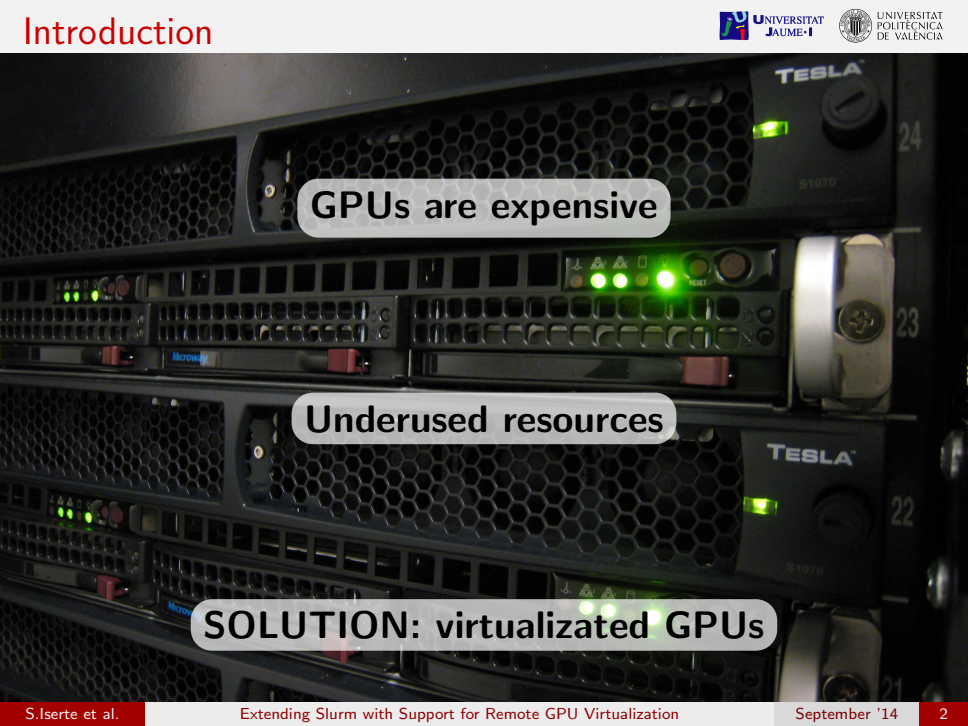
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Extending Slurm with Support for Remote GPU Virtualization

Sergio Iserte, Adrián Castelló, Rafael Mayo,
Enrique S. Quintana-Ortí, Federico Silla, Jose Duato

Slurm User Group Meeting 2014

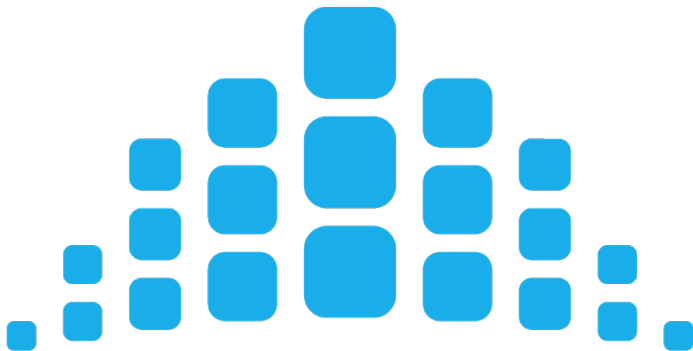
23 September 2014, Lugano (Switzerland)



GPUs are expensive

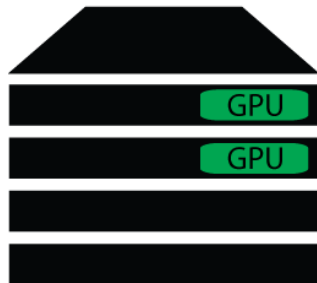
Underused resources

SOLUTION: virtualized GPUs



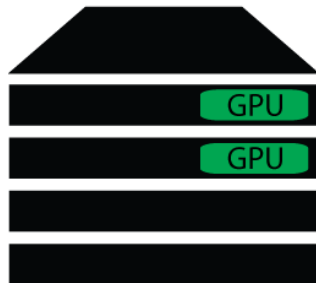
slurm

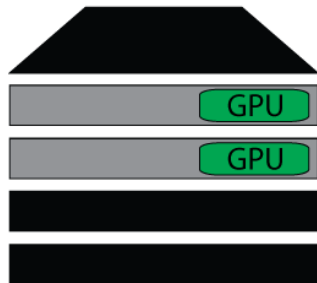
workload manager





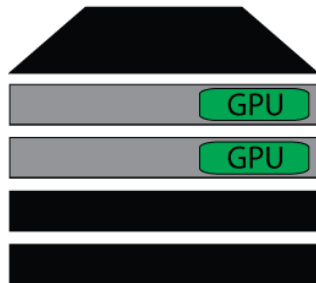
2 nodes







2 nodes
2 GPUs

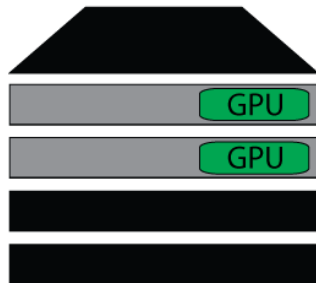




2 nodes
2 GPUs



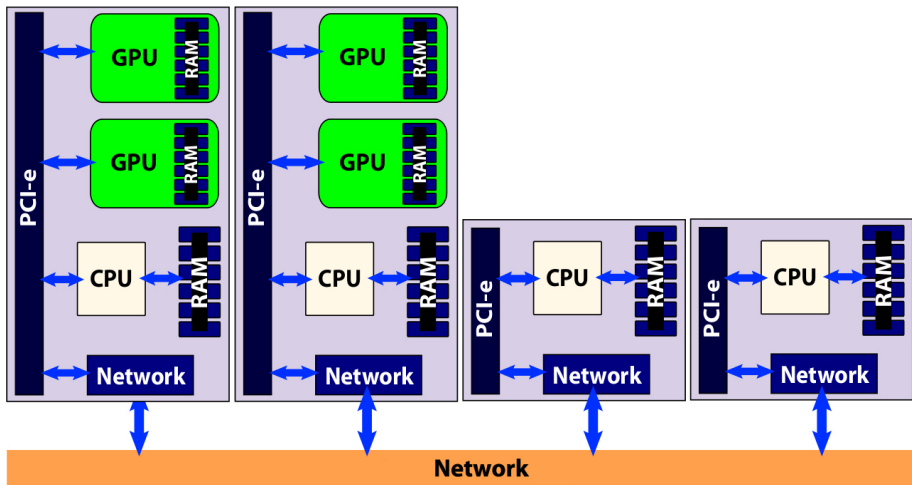
???

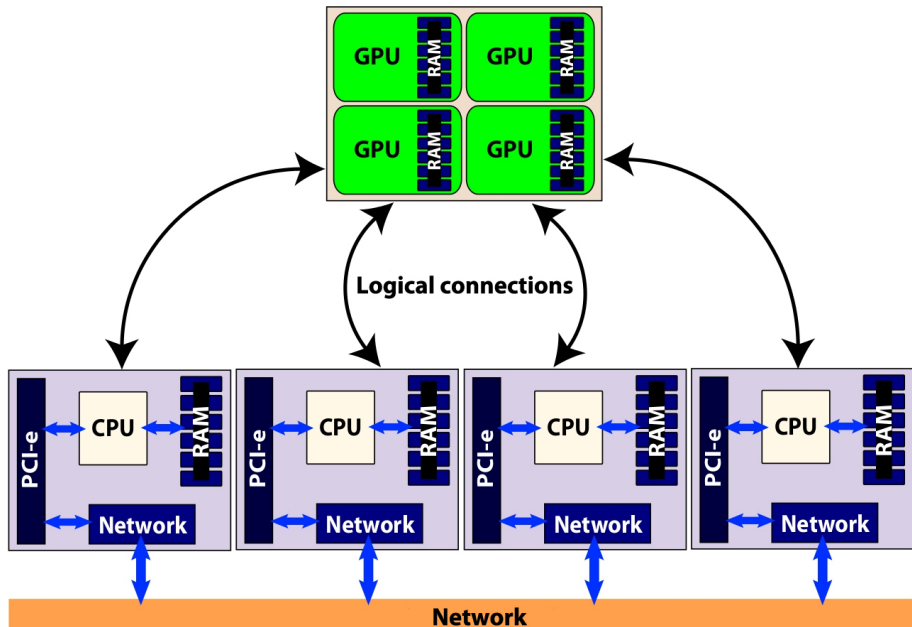


The logo for rCUDA features the word "rCUDA" in a large, bold, sans-serif font. The "r" is blue, while "CUDA" is black. A horizontal blue gradient bar with a white center passes through the middle of the letters. Below this, the words "remote CUDA" are written in a smaller, blue, italicized sans-serif font.

rCUDA
remote CUDA

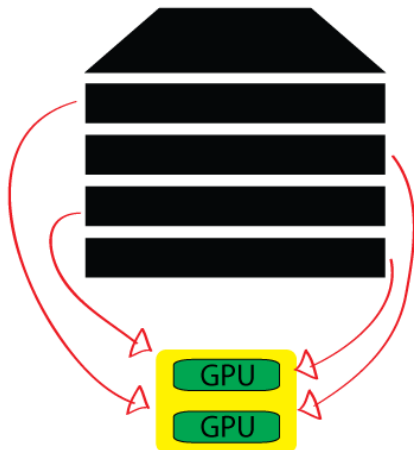
Cluster without GPU Virtualization

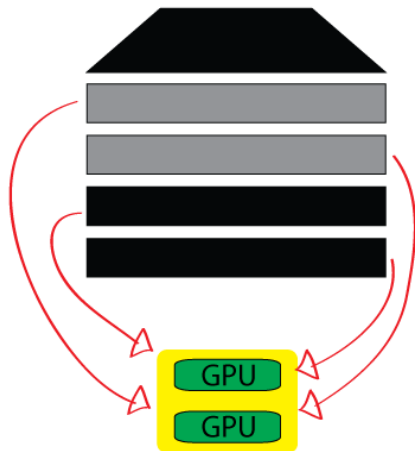






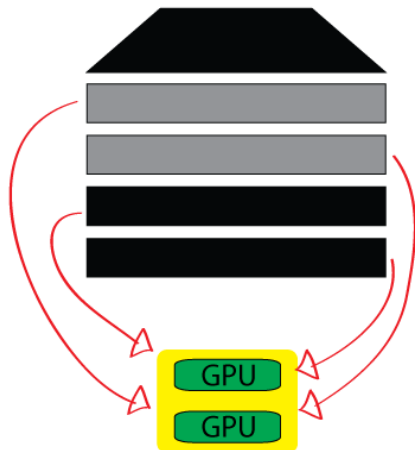
2 nodes

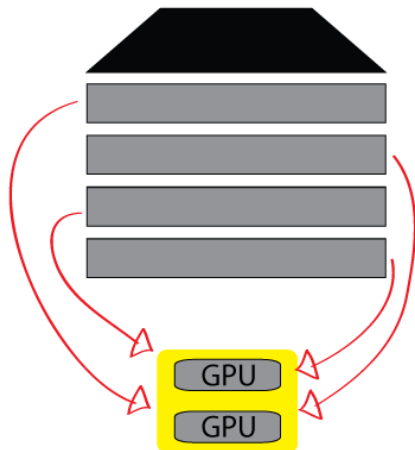


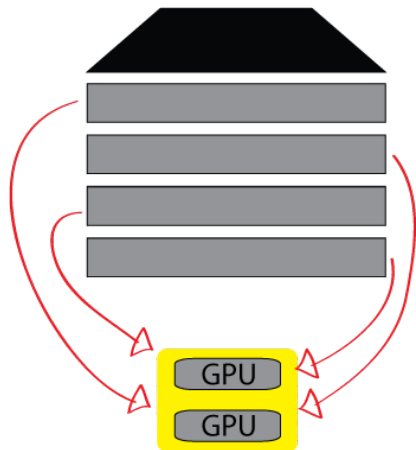


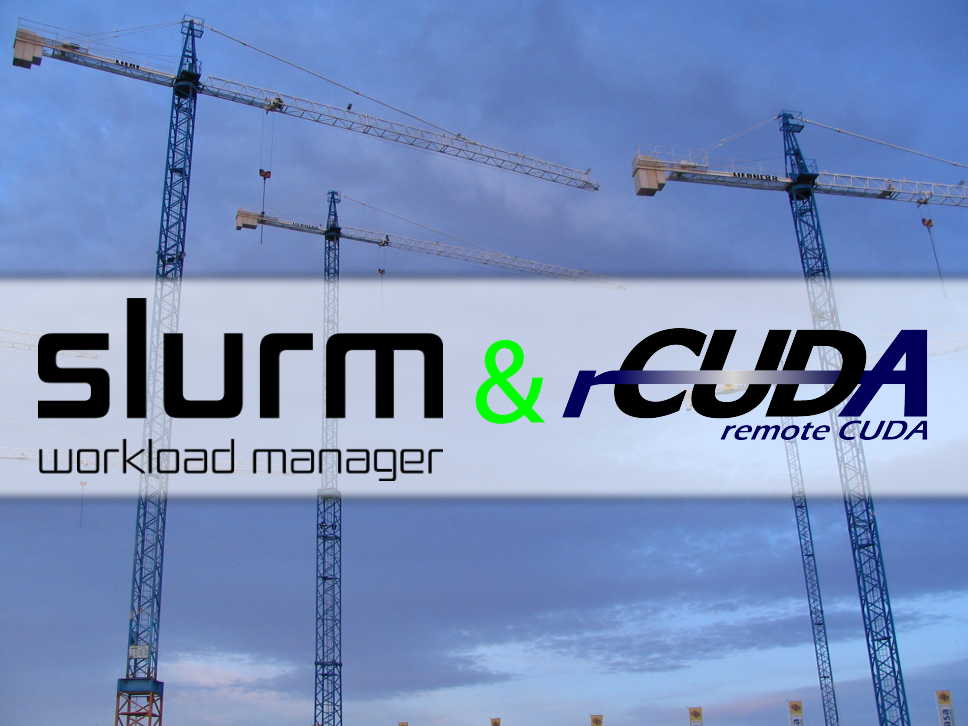


2 nodes
2 GPUs







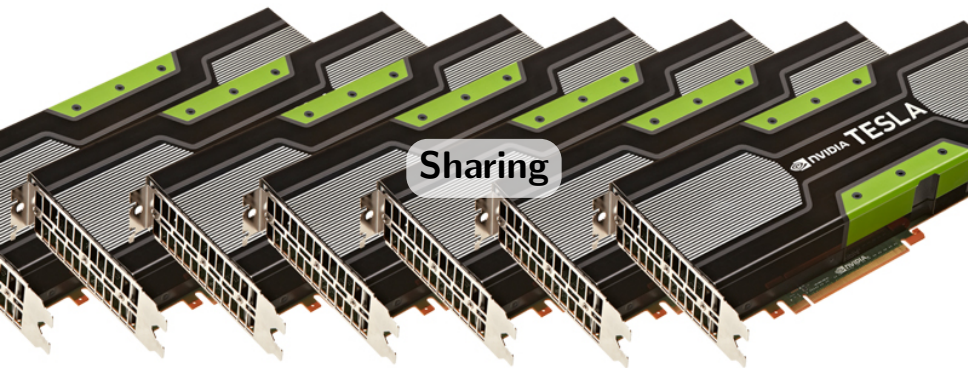


slurm
workload manager

&

rCUDA
remote CUDA

Special resource: rGPU




Remote Usage

`char`, `bitstr_t`, `gres_state_t`, `uint64_t`, `bool`

Partitions, Nodes, Jobs and Job Steps

Keep information about rGPUs



```
char *rgpulist;
```

```
packstr(msg->rgpulist, buffer);
```

```
safe_unpackstr_xmalloc(msg->rgpulist, &uint32_size, buffer);
```


`LD_LIBRARY_PATH`

`RCUDAPROTO`

`RCUDA_DEVICE_COUNT`

`RCUDA_DEVICE_X`

Global Counting of rGPUs

Isolate rGPUs from the job

rGPU selection plugin

Based on the *Consumable Resources* policy

Iterate the nodes searching for rGPUs

GRes driver is in charge of (de)allocations

Build the rGPU list of the nodes

Parse the rGPU-requirements of the jobs

Update rGPU resource information

slurm.conf

```
SelectType = select/cons_rgpu
SelectTypeParameters = CR_CORE
GresTypes = rgpu[,gpu]

NodeName=node1 NodeHostname=node1
  CPUs=12 Sockets=2 CoresPerSocket=6
  ThreadsPerCore=1 RealMemory=32072
  Gres=rgpu:1[,gpu:1]
```

gres.conf

```
Name=rgpu File=/dev/nvidia0 Cuda=3.5 Mem=4726M  
[Name=gpu File=/dev/nvidia0]
```

srun, sbatch, salloc

```
--cuda-mode=(shared|excl)
```

```
--gres=rgpu(:X(:Y)?(:Z)?)?
```

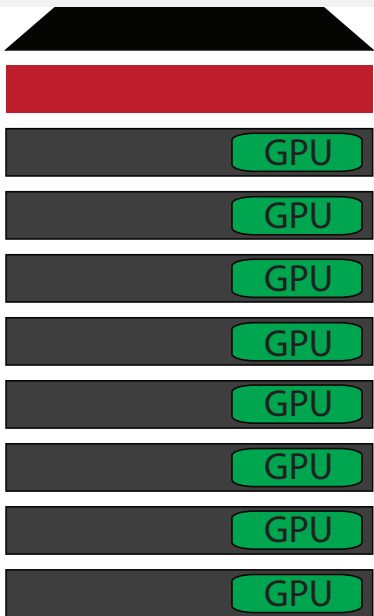
```
    X = [1-9]+[0-9]*
```

```
    Y = [1-9]+[0-9]*[ kKmMgG]
```

```
    Z = [1-9]\.[0-9](cc|CC)
```



experimentation



CentOS 6.4

2 x Intel Xeon E5-2620 Hexacore

NVIDIA Tesla K20 GPU

Mellanox SX6025 (FDR)

Application	Multi-process	Multi-thread	GPU Computational Load
GPU-Blast	-	X	Medium
LAMMPS	X	-	High
MCUDA-MEME	X	X	Medium
GROMACS	X	X	None, only CPU

25% of each application



2 hours



4 hours



8 hours

Comparison between GPU and rGPU

Improvement of the global throughput

Reduction of GPU devices

Maximum Performance

Application	CUDA submission	rCUDA submission
GPU-Blast	-c6 -gres=gpu:1	-c6 -gres=rgpu:1:1686M
LAMMPS	-N5 -n5 -gres=gpu:1	-N5 -n5 -gres=rgpu:5:3275M
MCUDA-MEME	-N4 -n4 -gres=gpu:1	-N4 -n4 -gres=rgpu:4:163M
GROMACS	-N2 -n2 -c12	-N2 -n2 -c12

High Performance Computing

HPC → **HTC**

High Throughput Computing

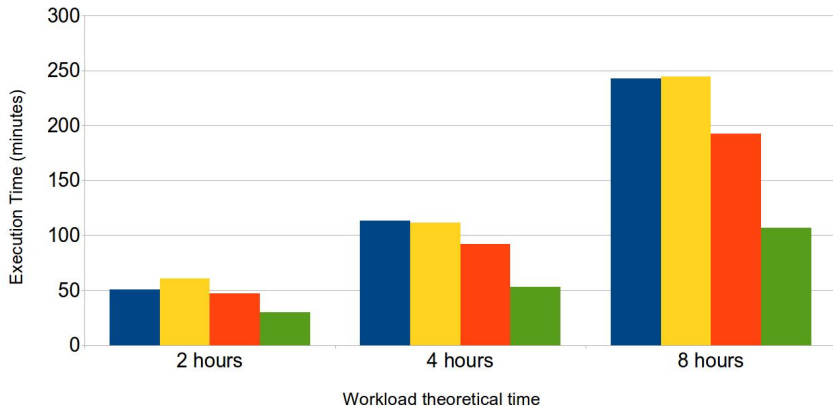
Maximum Throughput

Application	CUDA submission	rCUDA submission
GPU-Blast	-c6 -gres=gpu:1	-c6 -gres=rgpu:1:1686M
LAMMPS	-N5 -n5 -gres=gpu:1	-N5 -n5 -gres=rgpu:5:3275M
MCUDA-MEME	-N4 -n4 -gres=gpu:1	-N4 -n4 -gres=rgpu:4:163M
GROMACS	-N2 -n2 -c12	-N2 -n2 -c12

**CONFIDENTIAL
EXAMINATION
RESULTS**

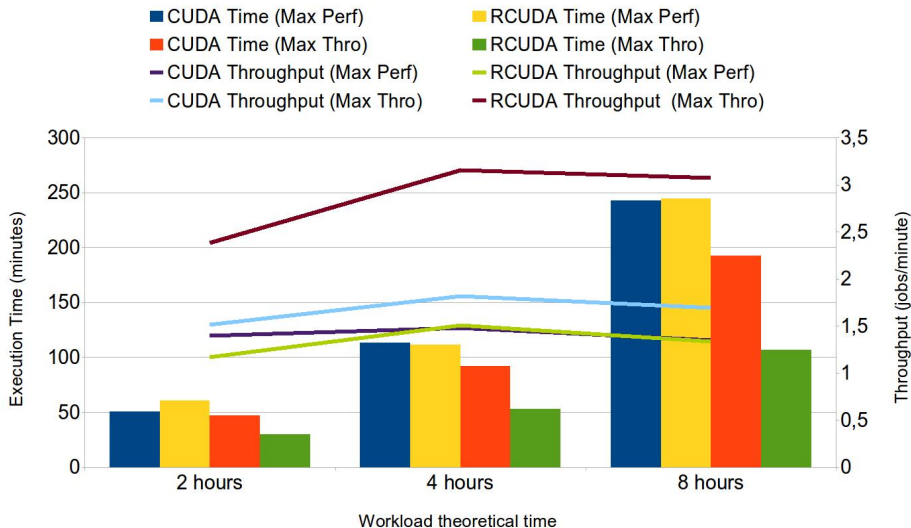
8-GPU Cluster

- CUDA Time (Max Perf)
- RCUDA Time (Max Perf)
- CUDA Time (Max Thro)
- RCUDA Time (Max Thro)



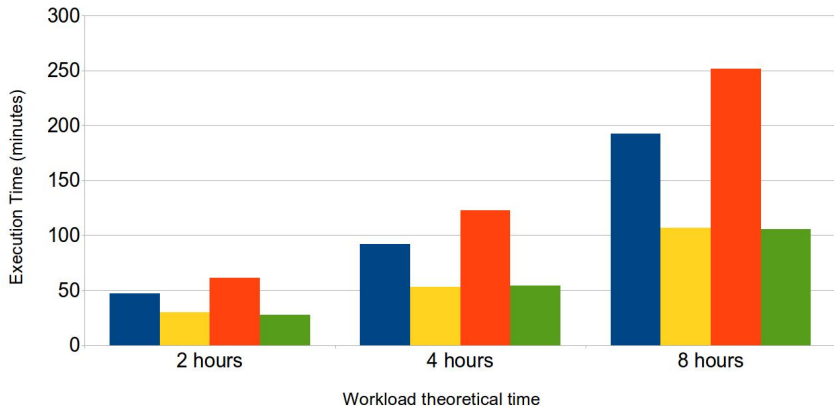
Max. Performance VS Max. Throughput

8-GPU Cluster



Maximum Throughput

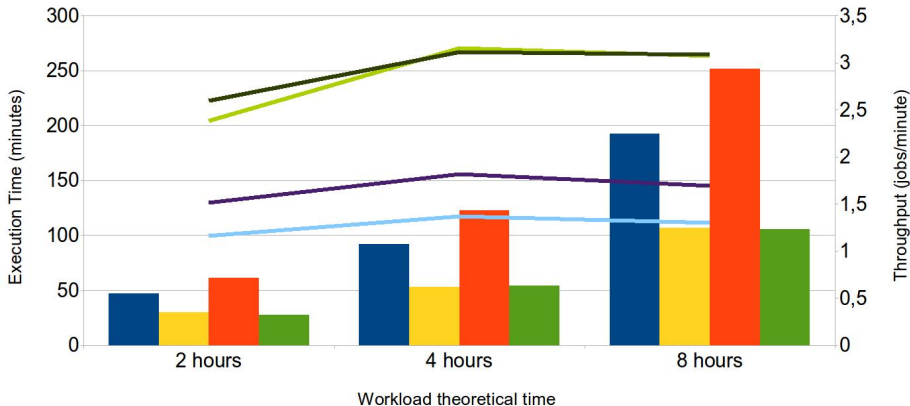
■ CUDA 8 GPUs Time ■ RCUDA 8 GPUs Time
■ CUDA 4 GPUs Time ■ RCUDA 4 GPUs Time



Max. Throughput with 8 and 4 GPUs

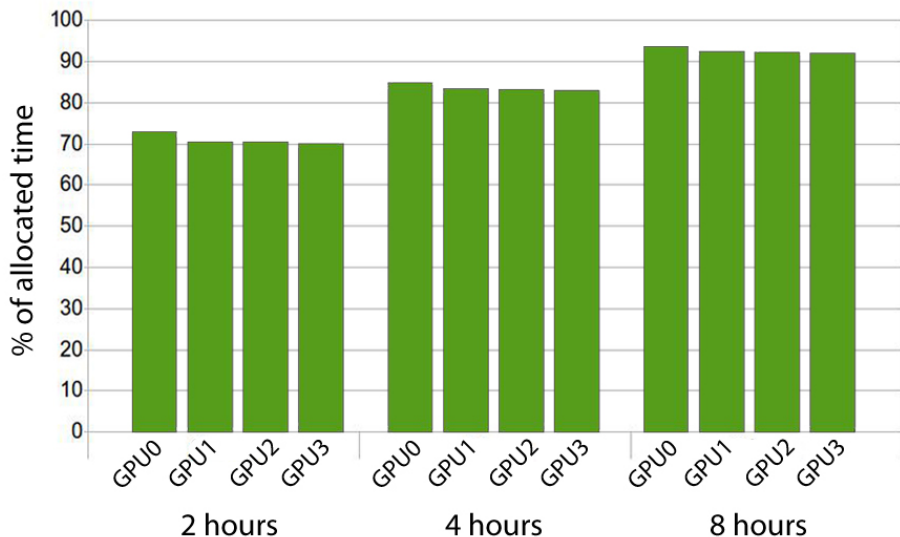
Maximum Throughput

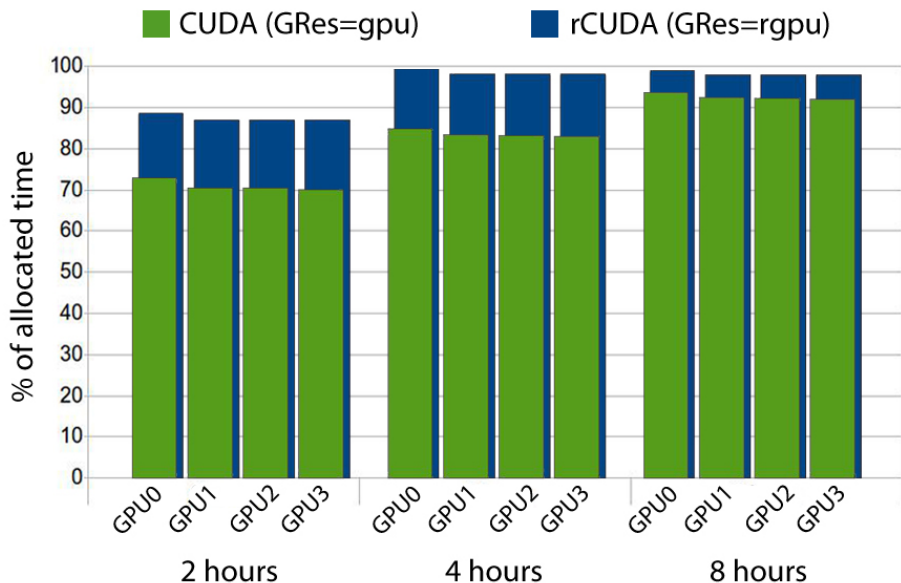
- CUDA 8 GPUs Time
- CUDA 4GPUs Time
- CUDA 8 GPUs Throughput
- CUDA 4 GPUs Throughput
- RCUDA 8 GPUs Time
- RCUDA 4 GPUs Time
- RCUDA 8 GPUs Throughput
- RCUDA 4 GPUs Throughput



■ CUDA (GRes=gpu)

■ rCUDA (GRes=rgpu)





C
o
n
c
l
u
s
i
o
n
s



Higher throughput even with less resources

THE
FUTURE
IS NOW

Higher throughput saving energy and money

rGPUs in production clusters is NOW available

Make the most of your GPUs with



REFERENCE:

S. Iserte, A. Castelló, R. Mayo, E. S. Quintana-Ortí, F. Silla, J. Duato, C. Reaño, J. Prades. C. Reaño, J. Prades.

SLURM Support for Remote GPU Virtualization: Implementation and Performance Study, in SBAC-PAD, 2014 (accepted).