Using remote GPUs with rCUDA

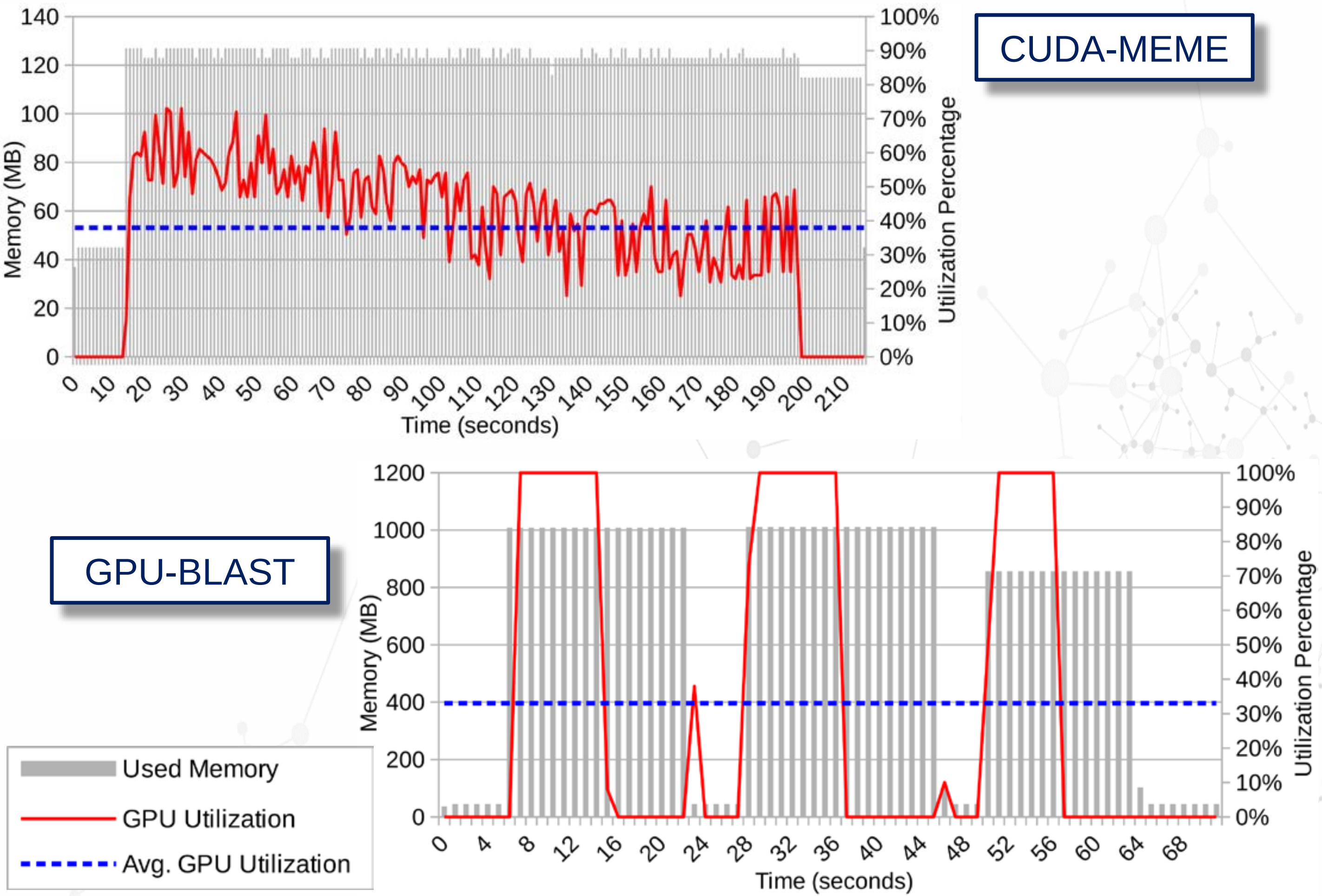
Federico Silla

Universitat Politècnica de València

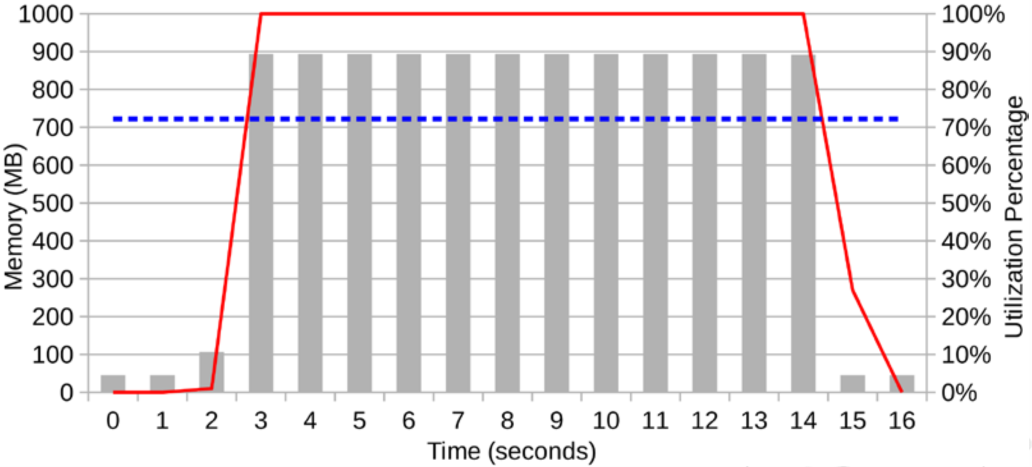
High Performance Container Workshop @ ISC 2018

Some motivation …

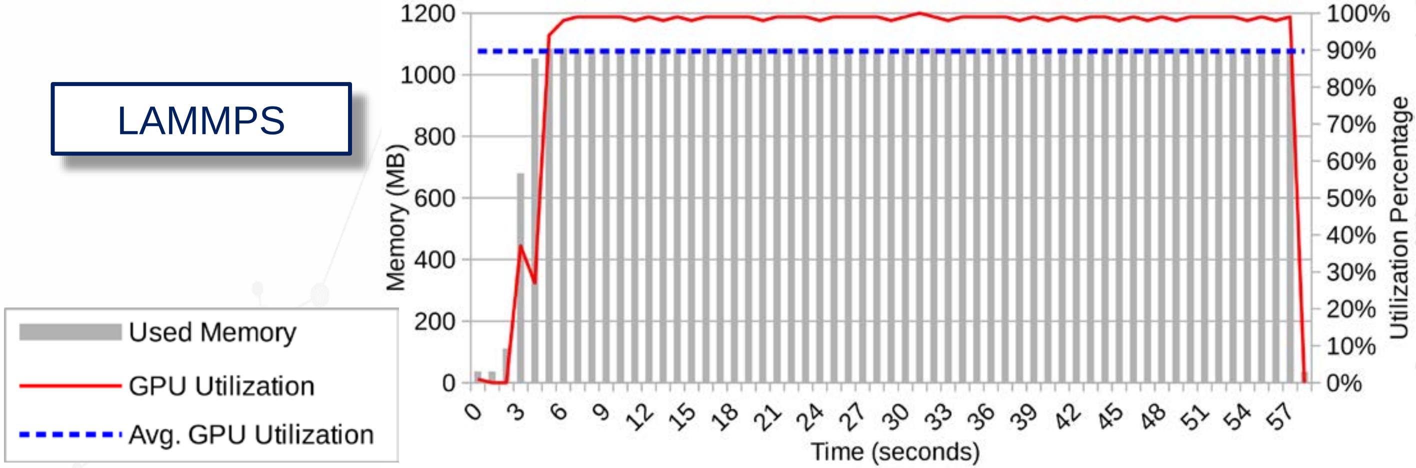
Are we making good use of GPUs?



Are we making good use of GPUs?

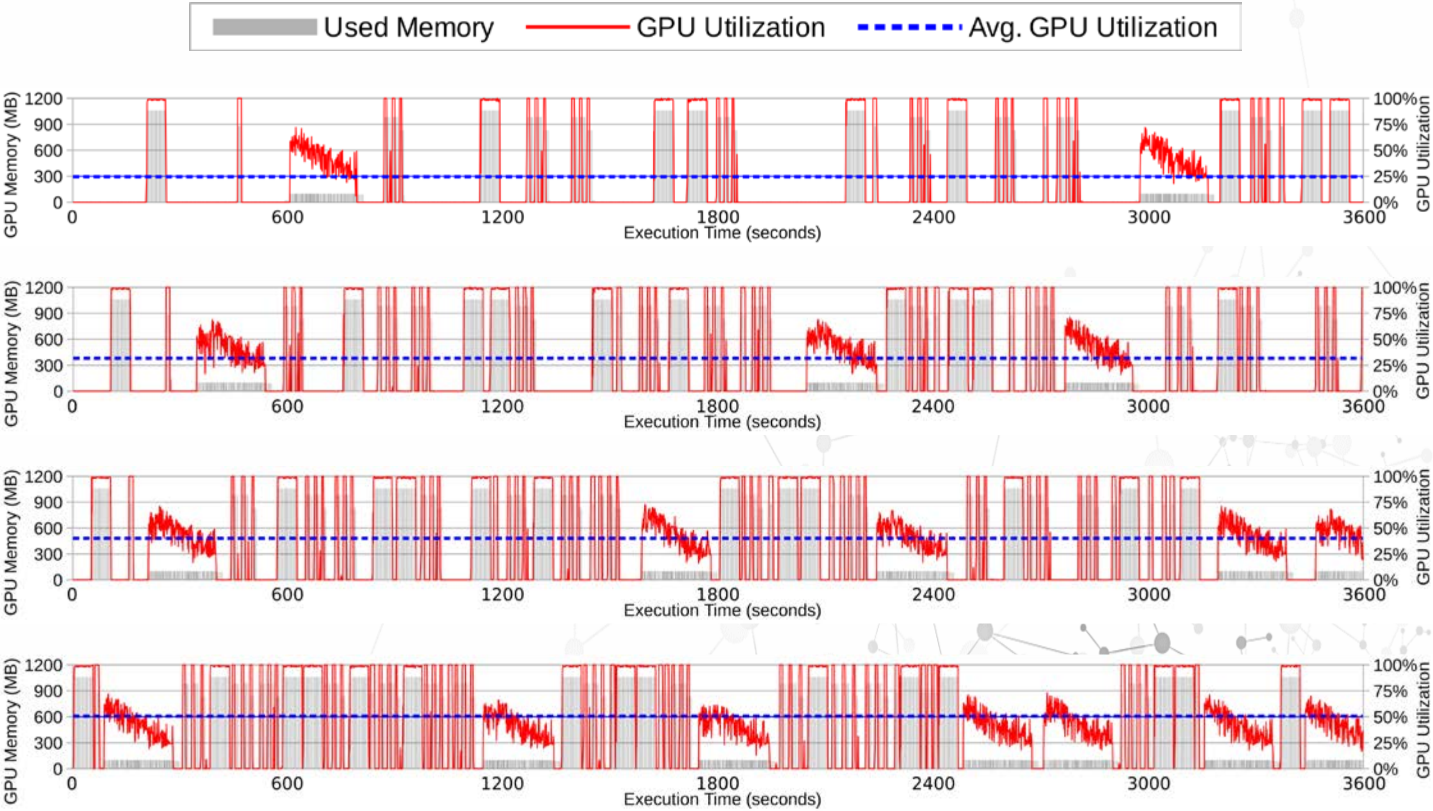


CUDASW++



LAMMPS

Are we making good use of GPUs?



Are we making good use of GPUs?

GPU utilization can be increased by **virtualizing** the GPU **and** concurrently **sharing** it among several applications

Are we making good use of GPUs?

virtualizing

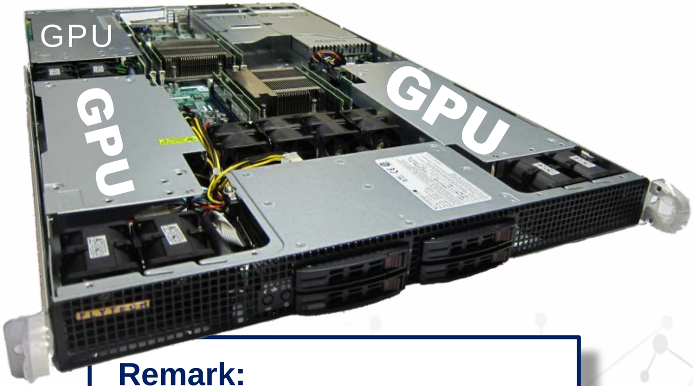
and **sharing**

Outline

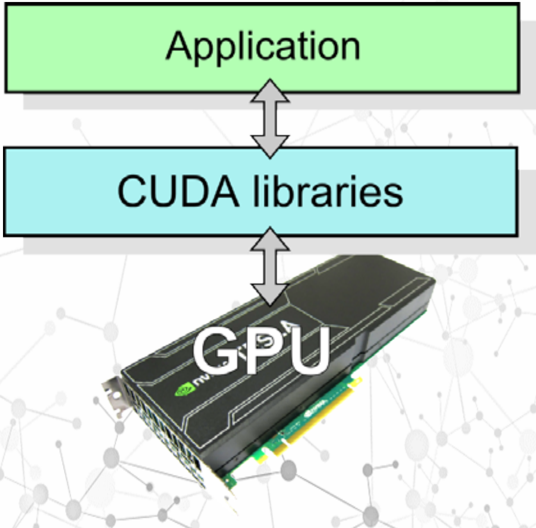
What is rCUDA?

Basics of GPU computing

Basic behavior of CUDA



Remark:
GPUs can only be used within the node they are attached to

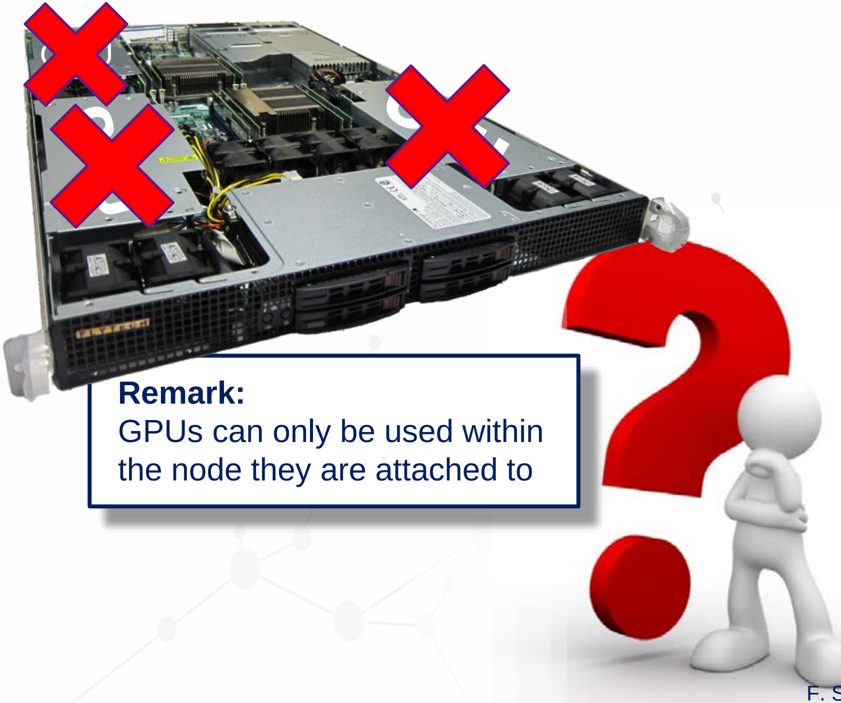


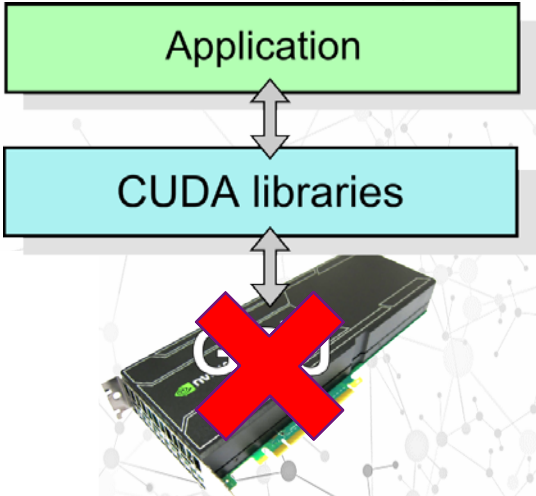
Application

CUDA libraries

Basics of GPU computing

Basic behavior of CUDA



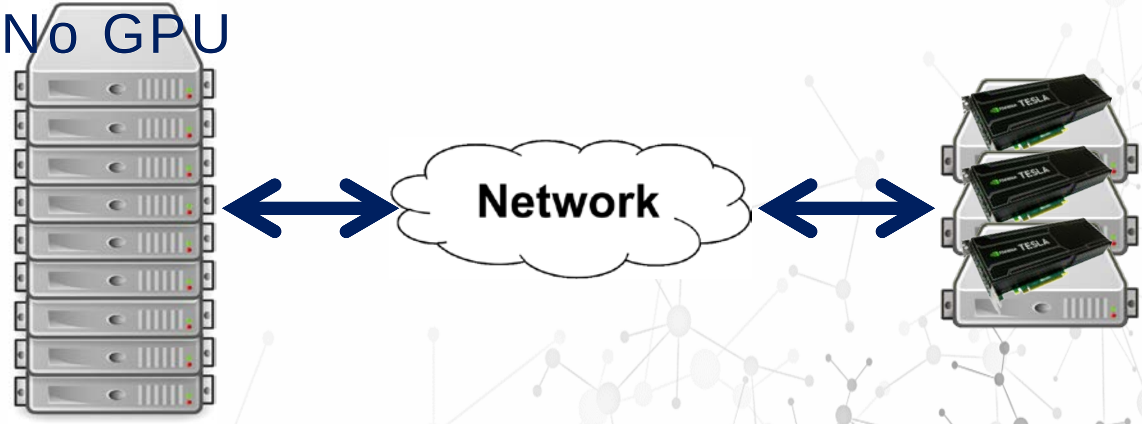


Application

CUDA libraries

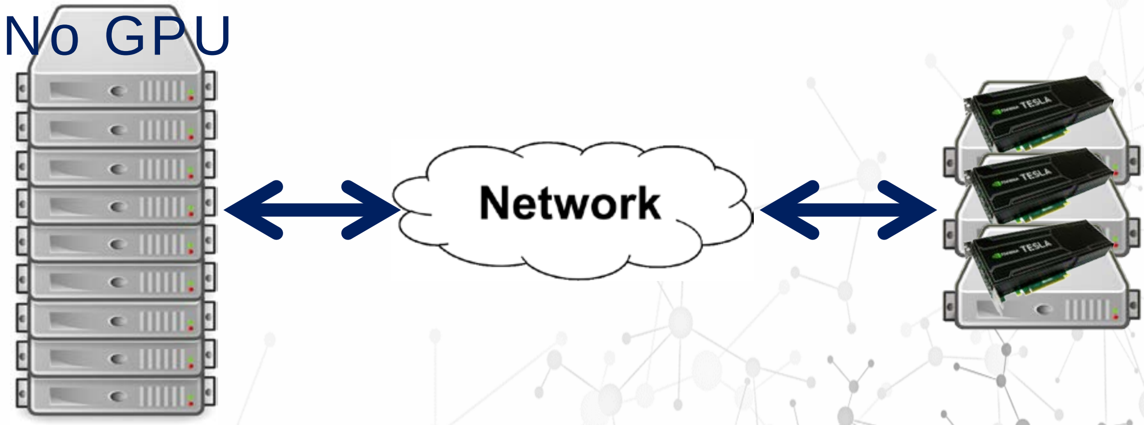
Remark:
GPUs can only be used within the node they are attached to

A different approach: remote GPU virtualization



A different approach: remote GPU virtualization

A software technology that enables a more flexible use of GPUs in computing facilities

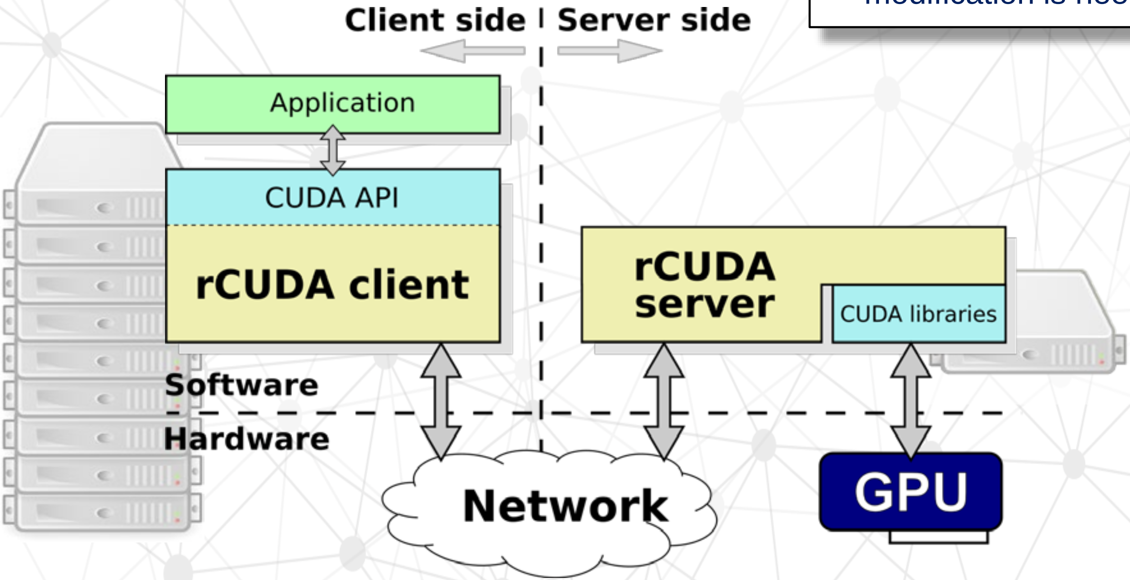


rCUDA … remote CUDA

rCUDA is a development by Universitat Politècnica de València

Basics or rCUDA

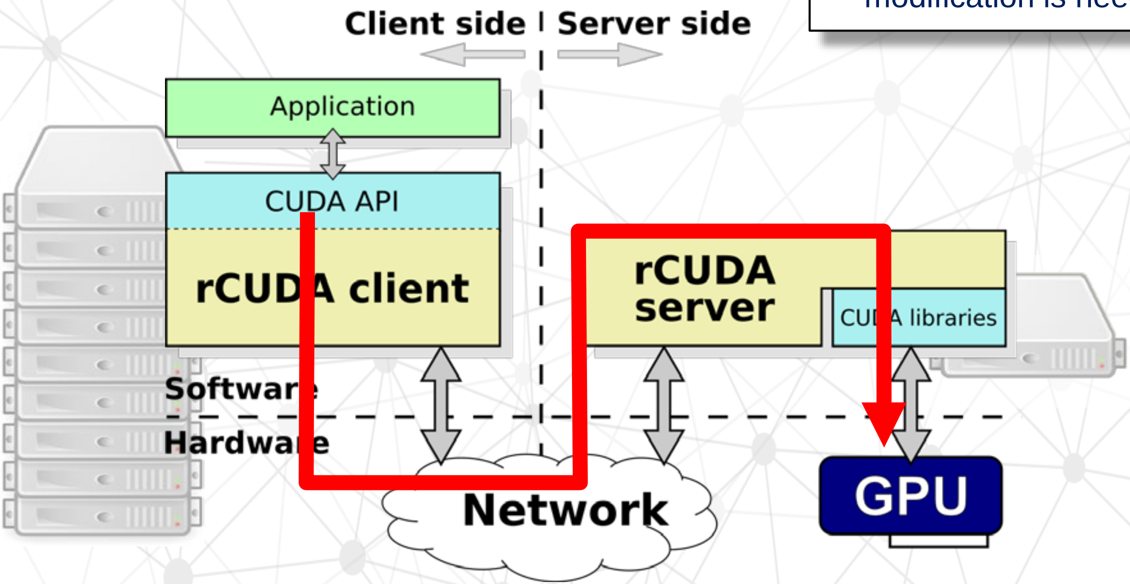
Access to remote GPU is transparent to applications: no source code modification is needed



rCUDA is a development by Universitat Politècnica de València

Basics or rCUDA

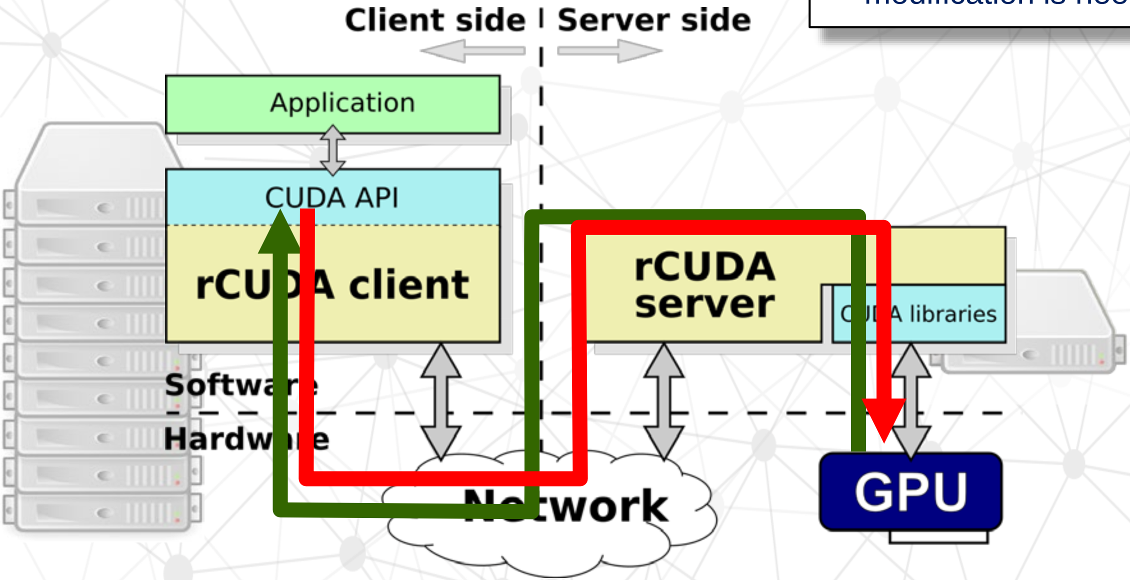
Access to remote GPU is transparent to applications: no source code modification is needed



rCUDA is a development by Universitat Politècnica de València

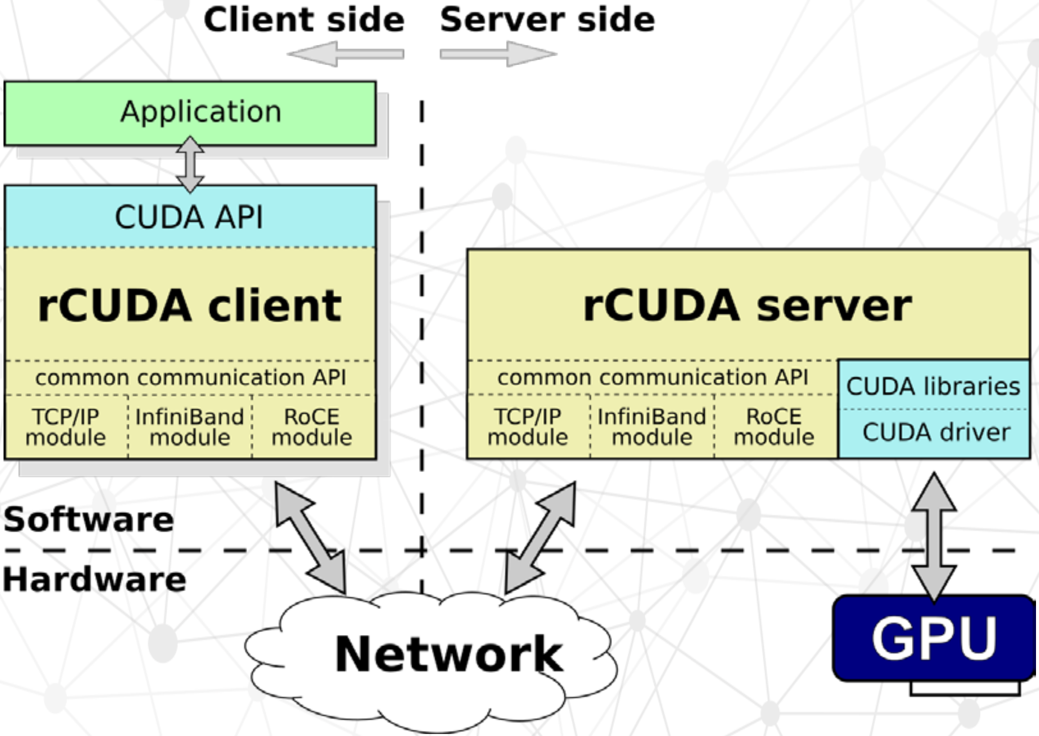
Basics or rCUDA

Access to remote GPU is transparent to applications: no source code modification is needed



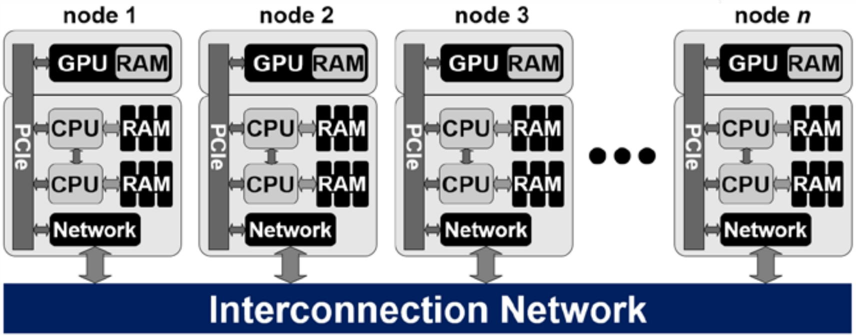
rCUDA is a development by Universitat Politècnica de València

rCUDA supports RDMA transfers



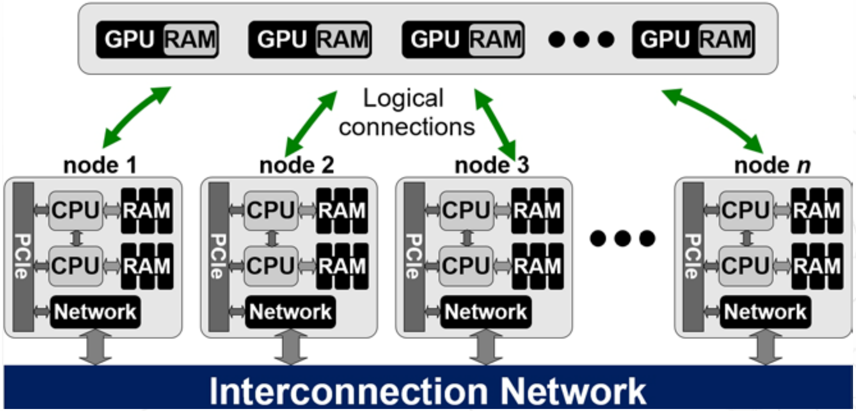
rCUDA envision

- **rCUDA allows a new vision** of a GPU deployment, moving from the usual cluster configuration …



Physical configuration

… to the following one:

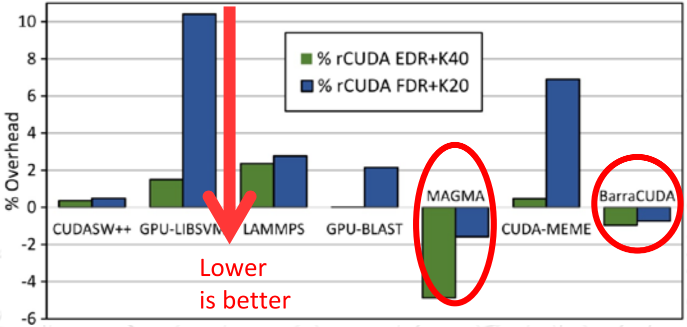


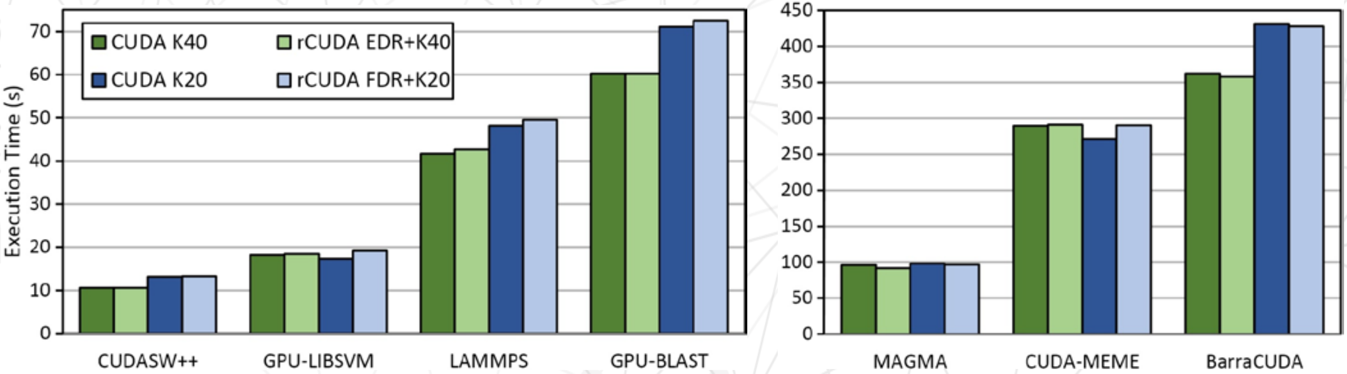
Logical configuration

Perfomance of rCUDA?

Performance of rCUDA

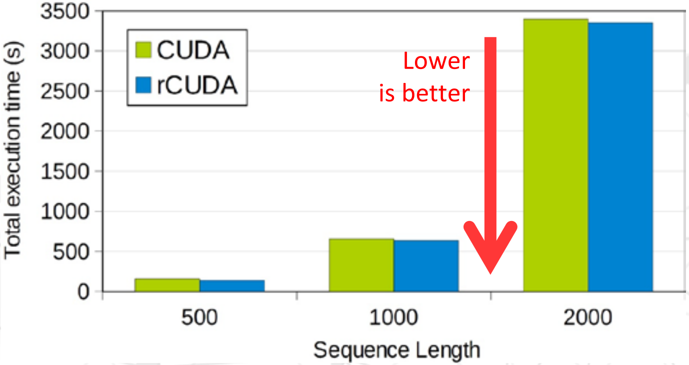
- K20 GPU and FDR InfiniBand
- K40 GPU and EDR InfiniBand





Performance of rCUDA

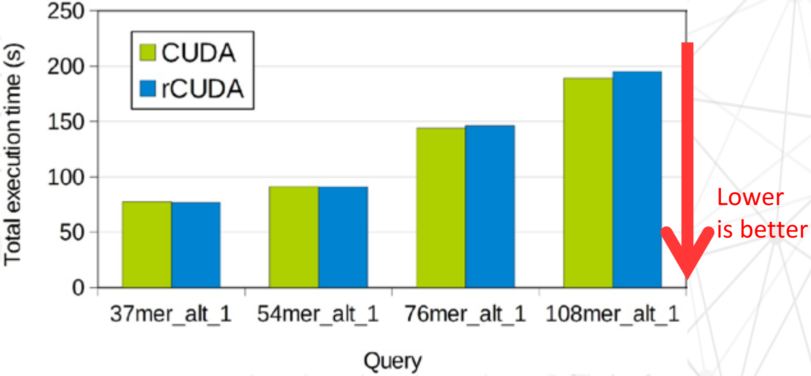
P100 GPU and EDR InfiniBand



Lower is better

CUDA-MEME

BarraCUDA



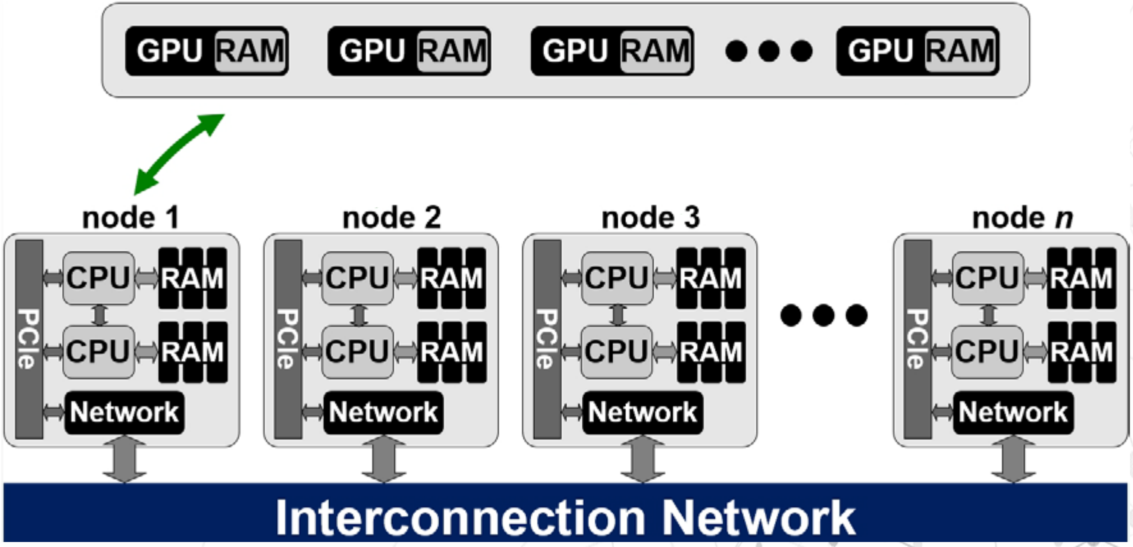
Lower is better

Benefits of rCUDA?

Benefits of rCUDA:

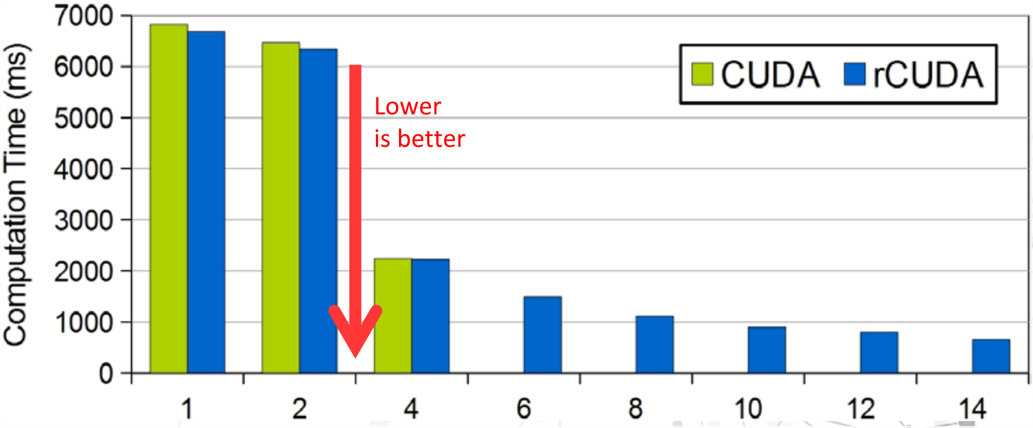
1. Many GPUs for an application
2. Server consolidation
3. Increased cluster throughput

Providing many GPUs to an application with rCUDA



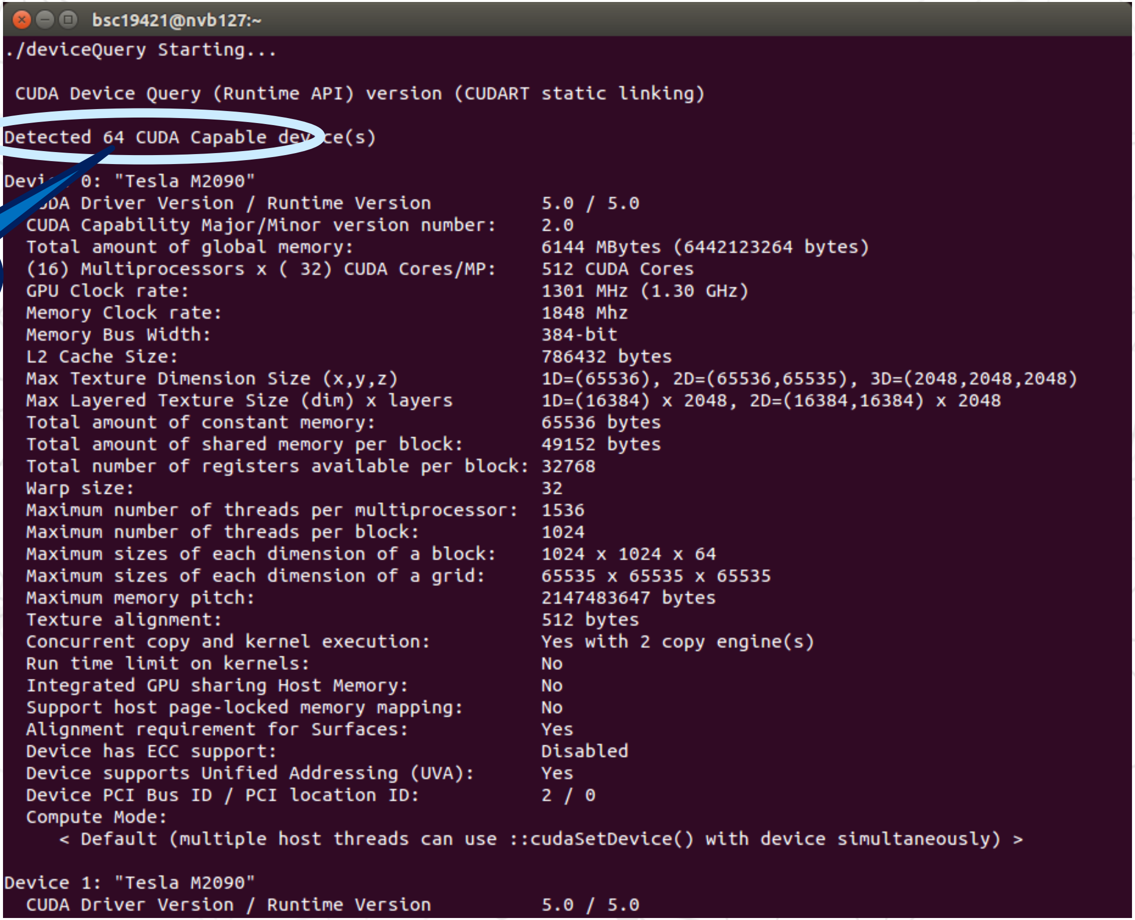
Providing many GPUs to an application with rCUDA

K20 GPUs and FDR InfiniBand



MonteCarlo multi-GPU program running in 14 NVIDIA Tesla K20 GPUs

Providing many GPUs to an application with rCUDA



bsc19421@nvb127:~

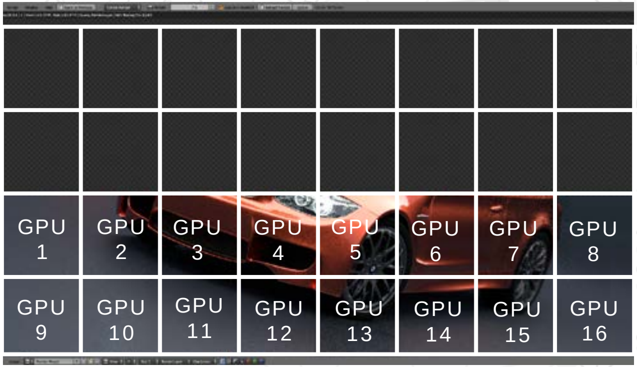
```
./deviceQuery Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 64 CUDA Capable device(s)

Device 0: "Tesla M2090"
  CUDA Driver Version / Runtime Version          5.0 / 5.0
  CUDA Capability Major/Minor version number:    2.0
  Total amount of global memory:                 6144 MBytes (6442123264 bytes)
  (16) Multiprocessors x ( 32) CUDA Cores/MP:    512 CUDA Cores
  GPU Clock rate:                                1301 MHz (1.30 GHz)
  Memory Clock rate:                             1848 Mhz
  Memory Bus Width:                              384-bit
  L2 Cache Size:                                 786432 bytes
  Max Texture Dimension Size (x,y,z)             1D=(65536), 2D=(65536,65535), 3D=(2048,2048,2048)
  Max Layered Texture Size (dim) x layers        1D=(16384) x 2048, 2D=(16384,16384) x 2048
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 32768
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1536
  Maximum number of threads per block:           1024
  Maximum sizes of each dimension of a block:    1024 x 1024 x 64
  Maximum sizes of each dimension of a grid:     65535 x 65535 x 65535
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 2 copy engine(s)
  Run time limit on kernels:                     No
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       No
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  Device supports Unified Addressing (UVA):      Yes
  Device PCI Bus ID / PCI location ID:           2 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

Device 1: "Tesla M2090"
  CUDA Driver Version / Runtime Version          5.0 / 5.0
```

64
GPUs !!

Providing many GPUs to an application with rCUDA

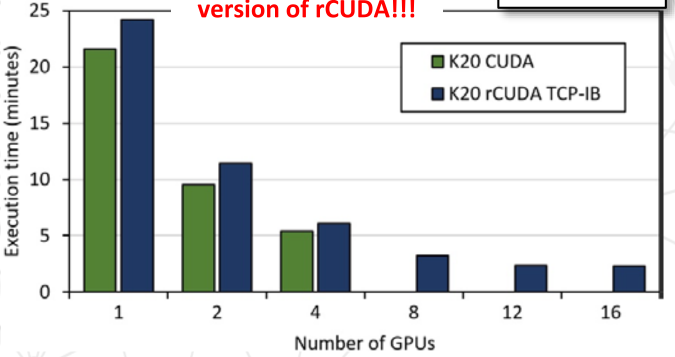
Work in progress!!

blender™

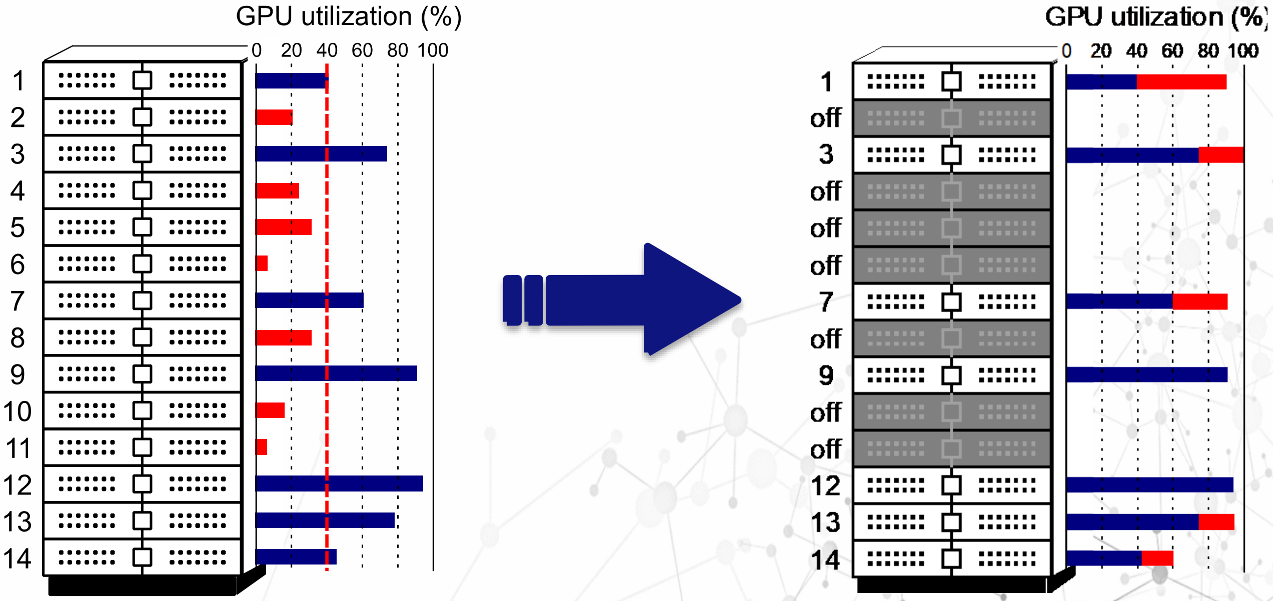


non-optimized (yet) version of rCUDA!!!

K20 GPUs

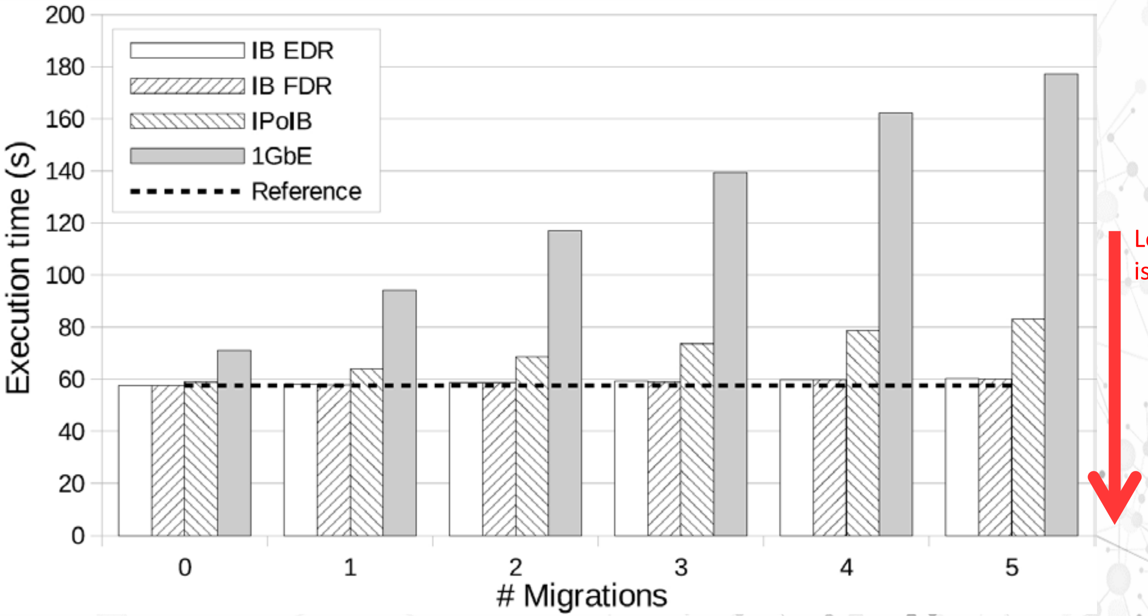


Server consolidation with rCUDA



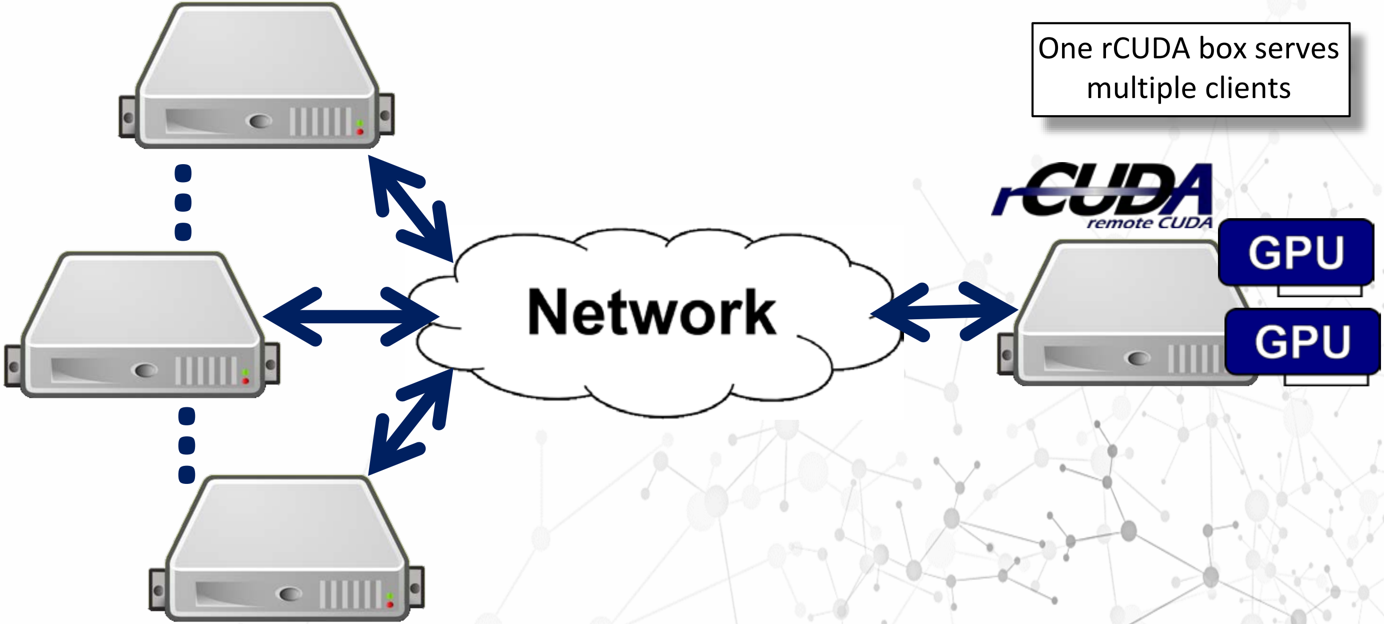
Server consolidation with rCUDA

- The GPU-Blast application is migrated up to 5 times among K40 GPUs
  - The aggregated volume of GPU data is 1300 MB (consisting of 9 memory regions)



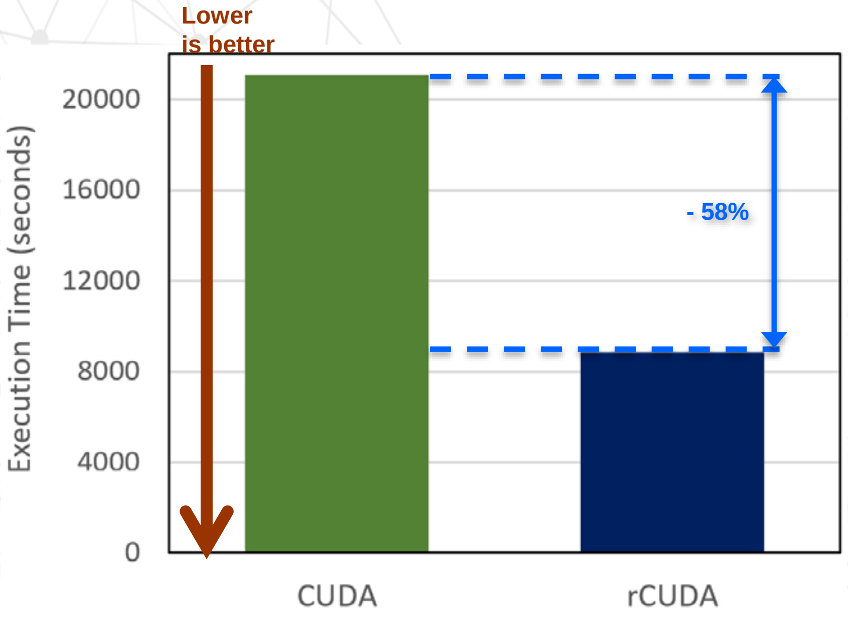
The "Reference" line is the execution time of the application when using CUDA with a local GPU and without any migration

Increased cluster throughput



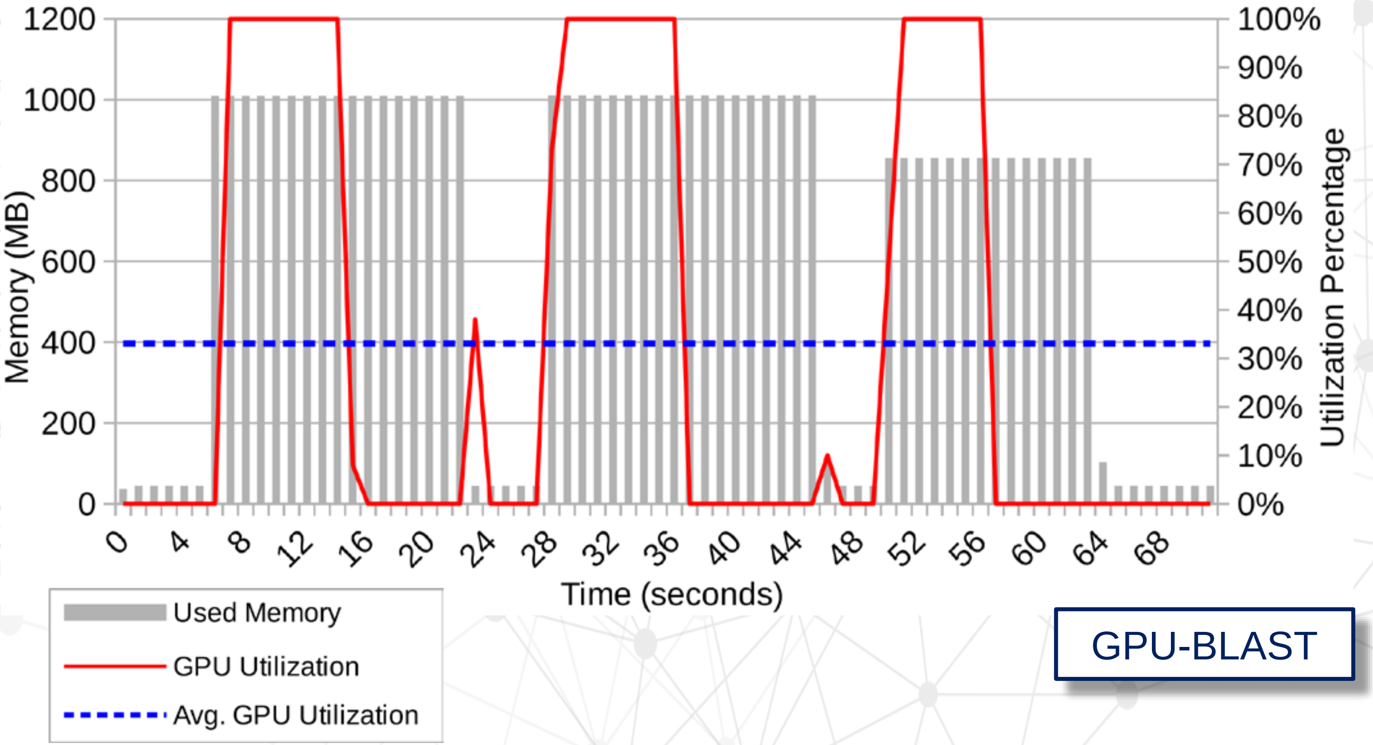
One rCUDA box serves multiple clients

Increased cluster throughput



1. BarraCUDA
2. CUDA-MEME
3. CUDASW++
4. GPU-Blast
5. Gromacs
6. Magma

Increased cluster throughput



GPU-BLAST

Outline

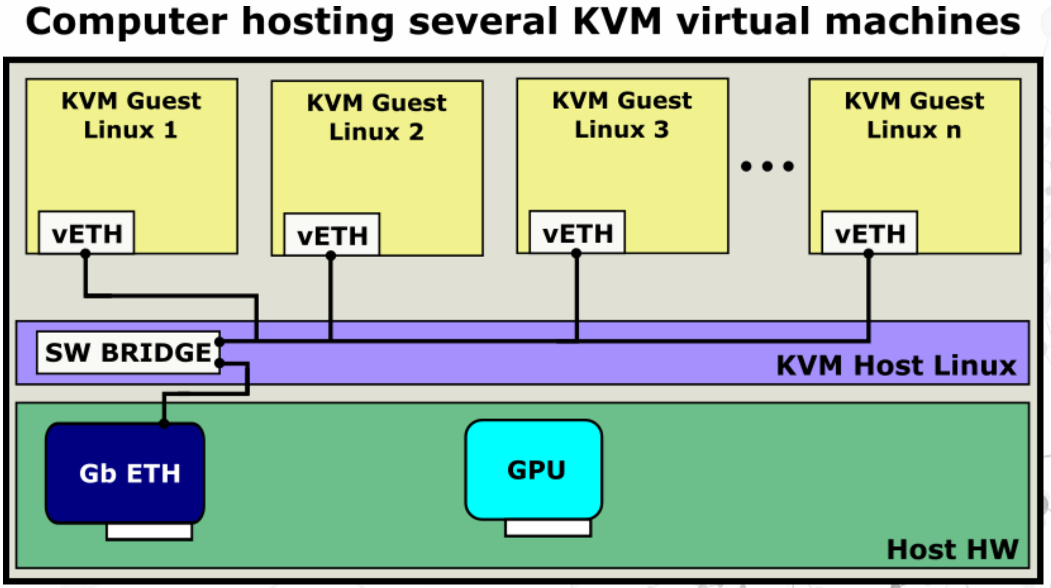
rCUDA and virtual machines

Outline

rCUDA and containers

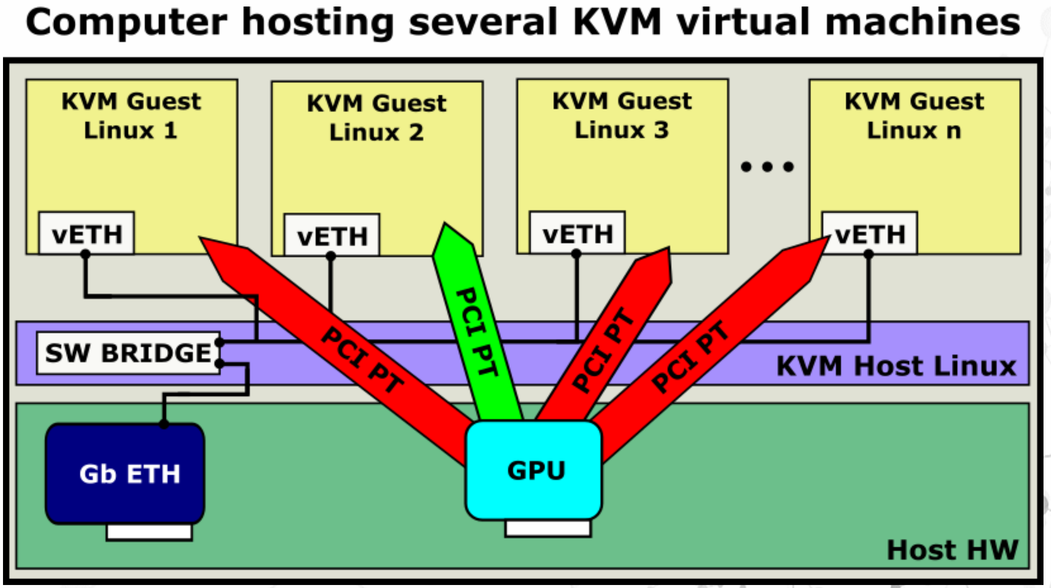
Virtual machines may need access to GPUs

- How to access the GPU in the native domain from inside of virtual machines?



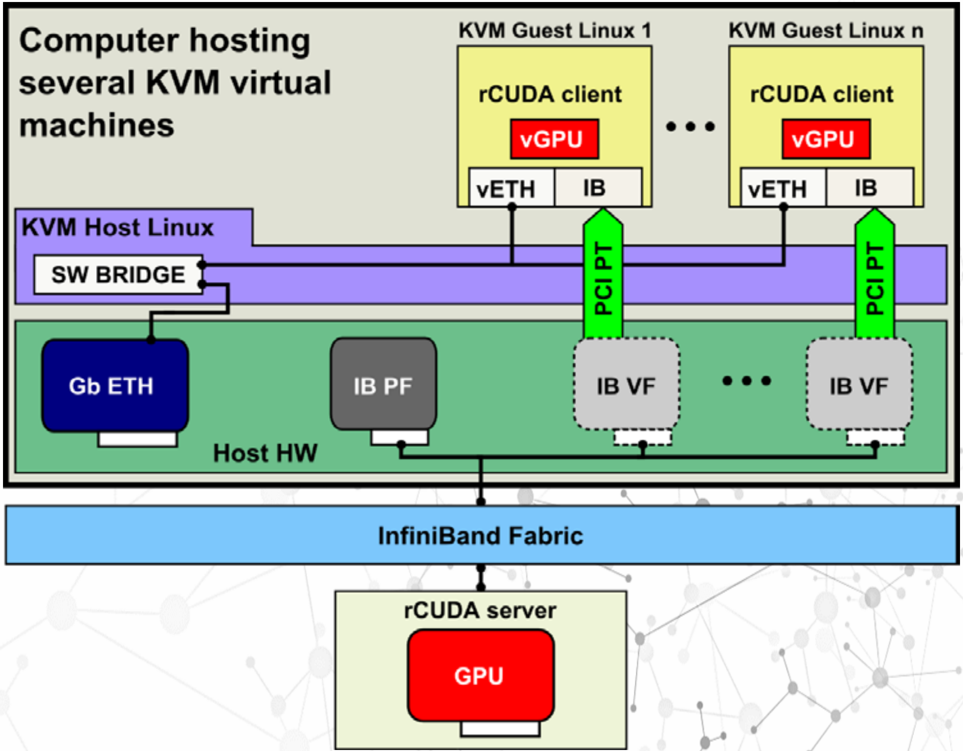
Virtual machines may need access to GPUs

- The GPU is assigned by using PCI passthrough exclusively to a single virtual machine
- Concurrent usage of the GPU is not possible

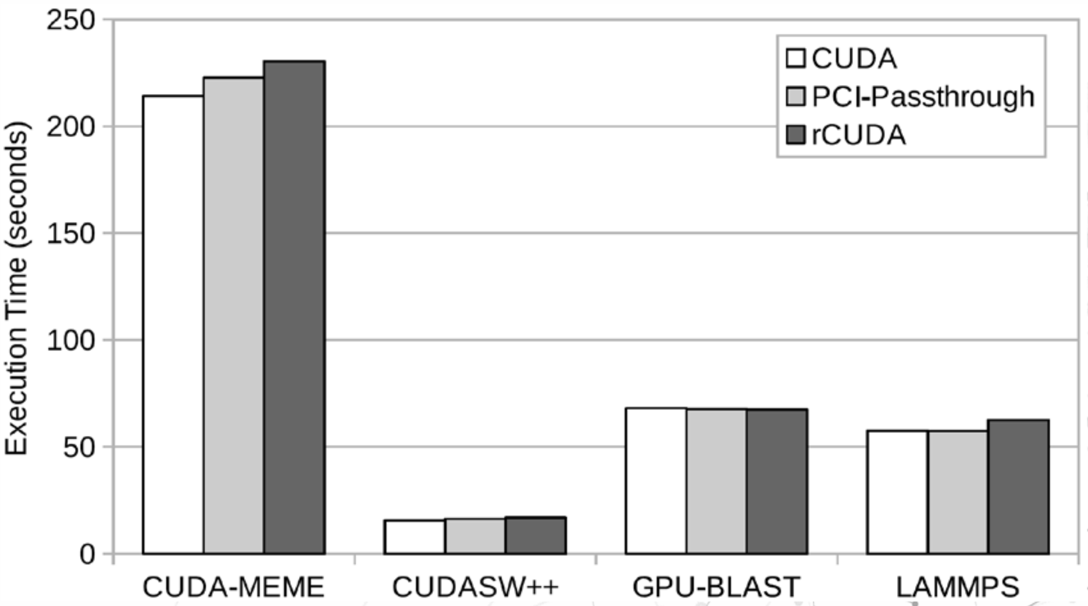


Using rCUDA to access the GPU

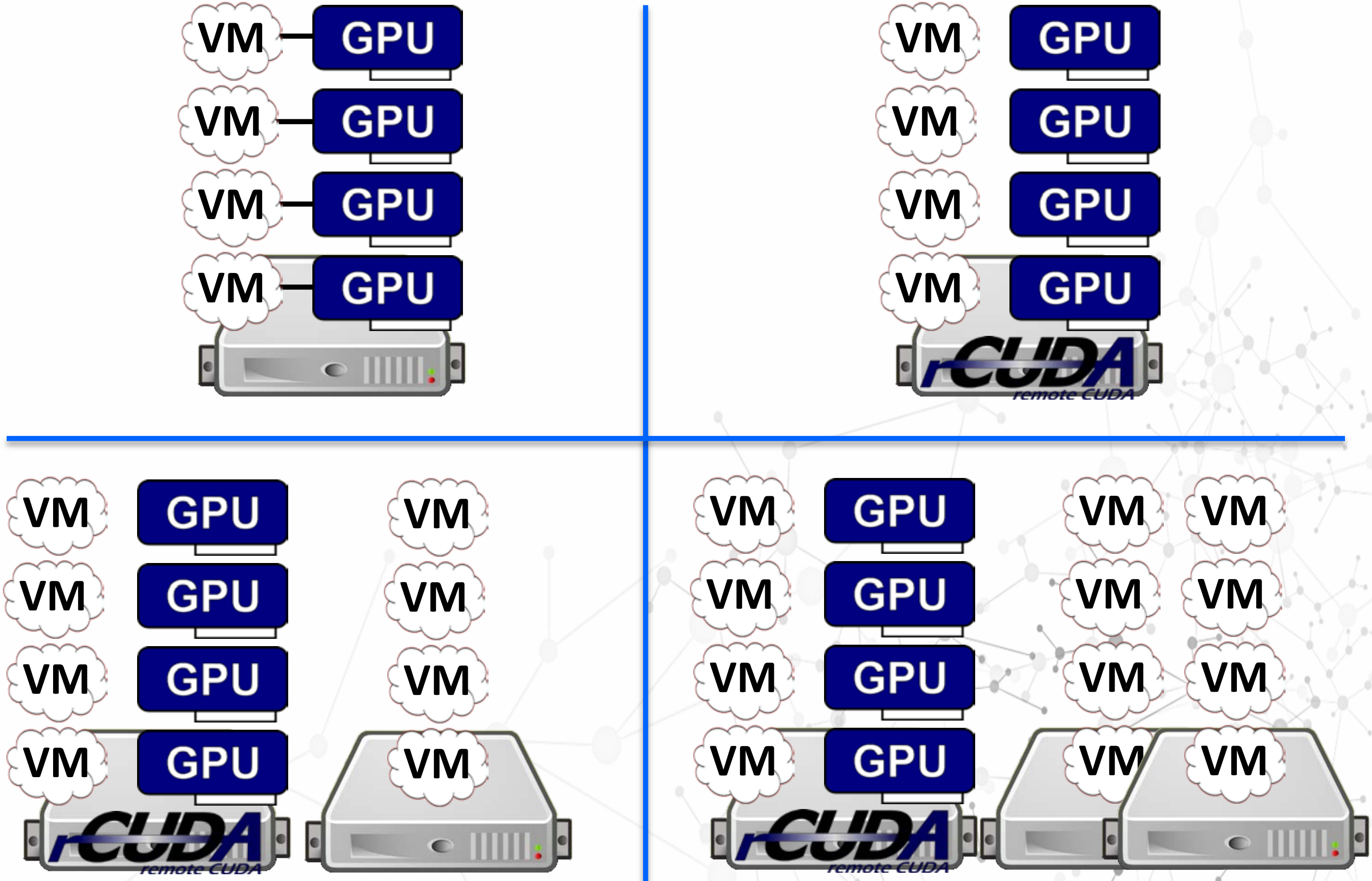
- If InfiniBand is available, the rCUDA server can be placed in another node
- Several GPUs can be provided to the VMs, either in a single remote node or in several remote nodes



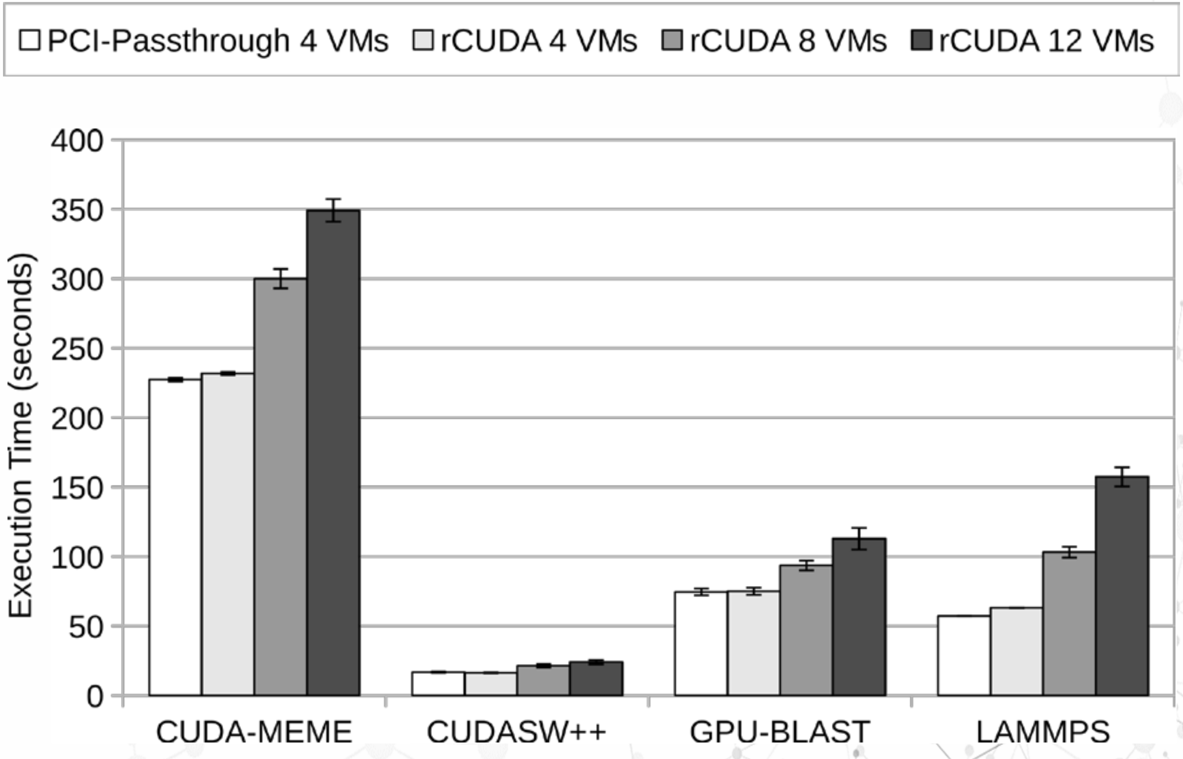
Using rCUDA to access the GPU



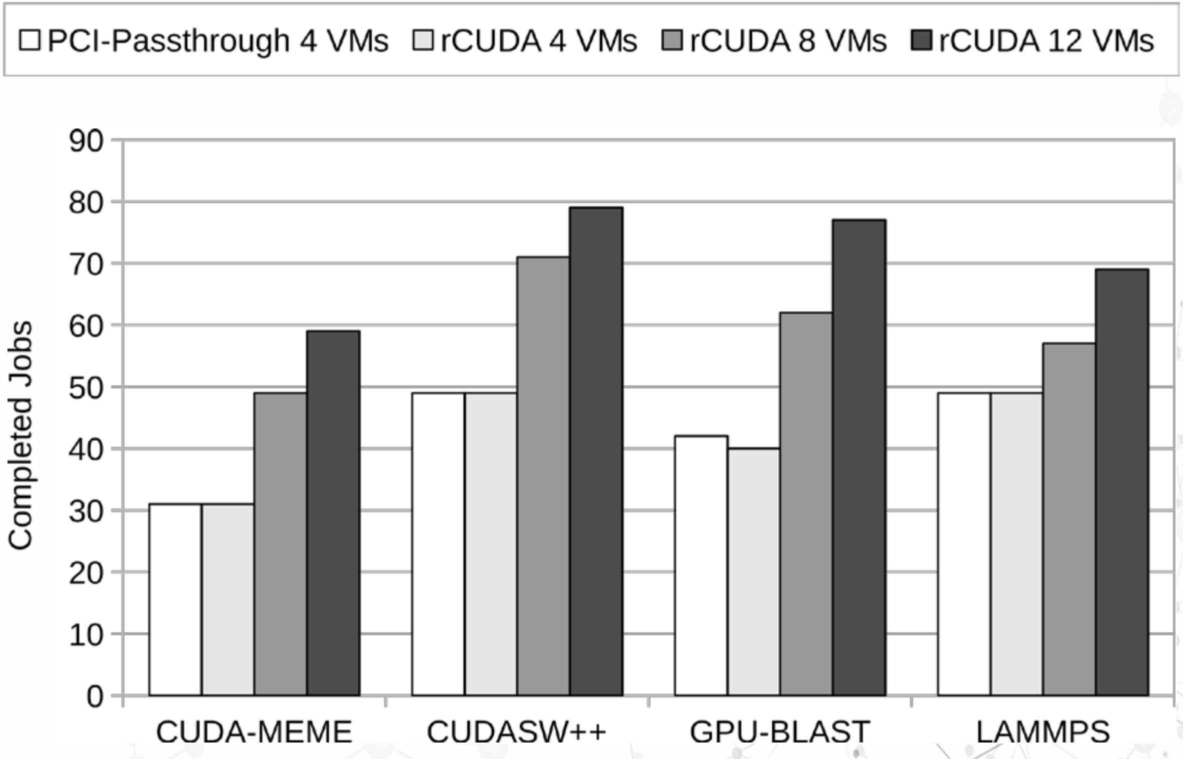
Using (sharing) the available GPUs



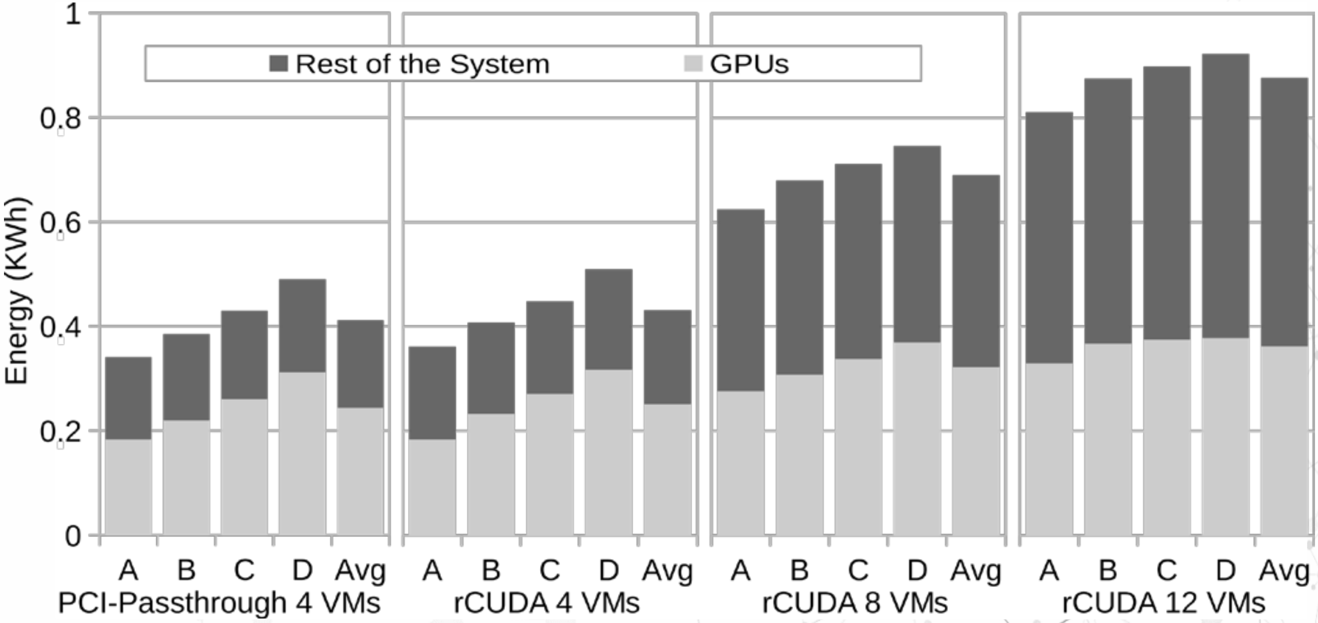
Using (sharing) the available GPUs



Using (sharing) the available GPUs



Using (sharing) the available GPUs



rCUDA
remote CUDA

Get a free copy of rCUDA at

http://www.rcuda.net

More than 900 requests world wide

@rcuda_

rCUDA is a development by Universitat Politècnica de València, Spain

·Tony Díaz · Pablo Higueras · Javier Prades · Jaime Sierra

· Cristian Peñaranda · Federico Silla · Carlos Reaño

rCUDA is a development by Universitat Politècnica de València, Spain