Enhancing IoT with remote GPU virtualization: the rCUDA approach

Federico Silla

Universitat Politècnica de València

HPC ADMINTECH 2021

Mejorando la IoT con la virtualización remota de GPUs: el caso rCUDA

Federico Silla

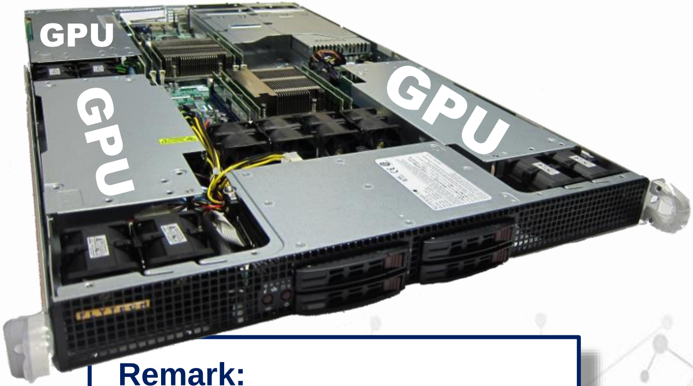
Universitat Politècnica de València

HPC ADMINTECH 2021

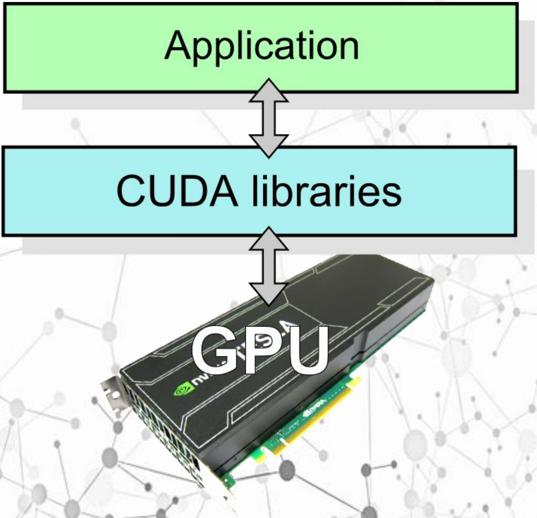
Some motivation …

Basics of GPU computing

Basic behavior of CUDA

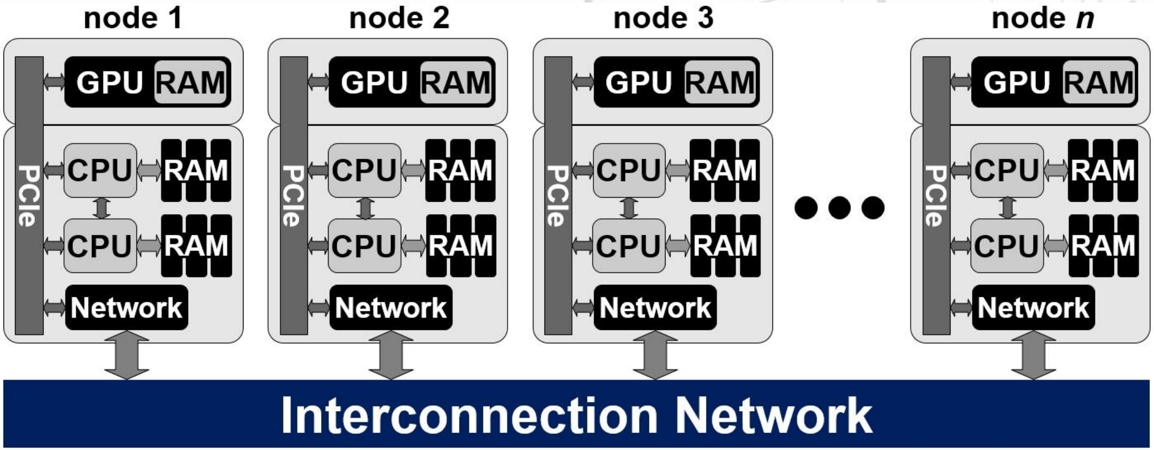


Remark:
GPUs can only be used within the node they are attached to



Using GPUs in a cluster

- A GPU-enabled cluster is a set of independent self-contained nodes. The cluster follows the **shared-nothing approach**:
    - Nothing is directly shared among nodes (MPI is typicaly required for aggregating computing resources within the cluster, **included GPUs**)
    - **GPUs can only be used within the node they are attached to**



Problems of using GPUs

We notice 3 main problems when using GPUs:

1. **GPUs can only be used locally in the node where they are installed**

2. **GPU utilization is, in general, low**

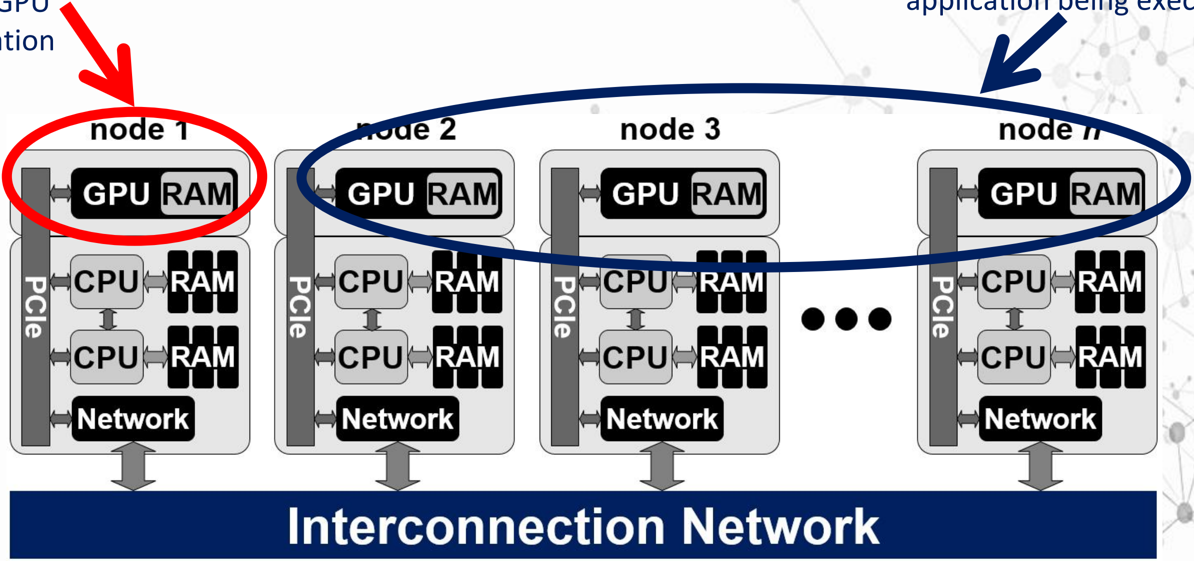
3. **GPUs keep consuming energy even when idle**

1. GPUs can only be used locally

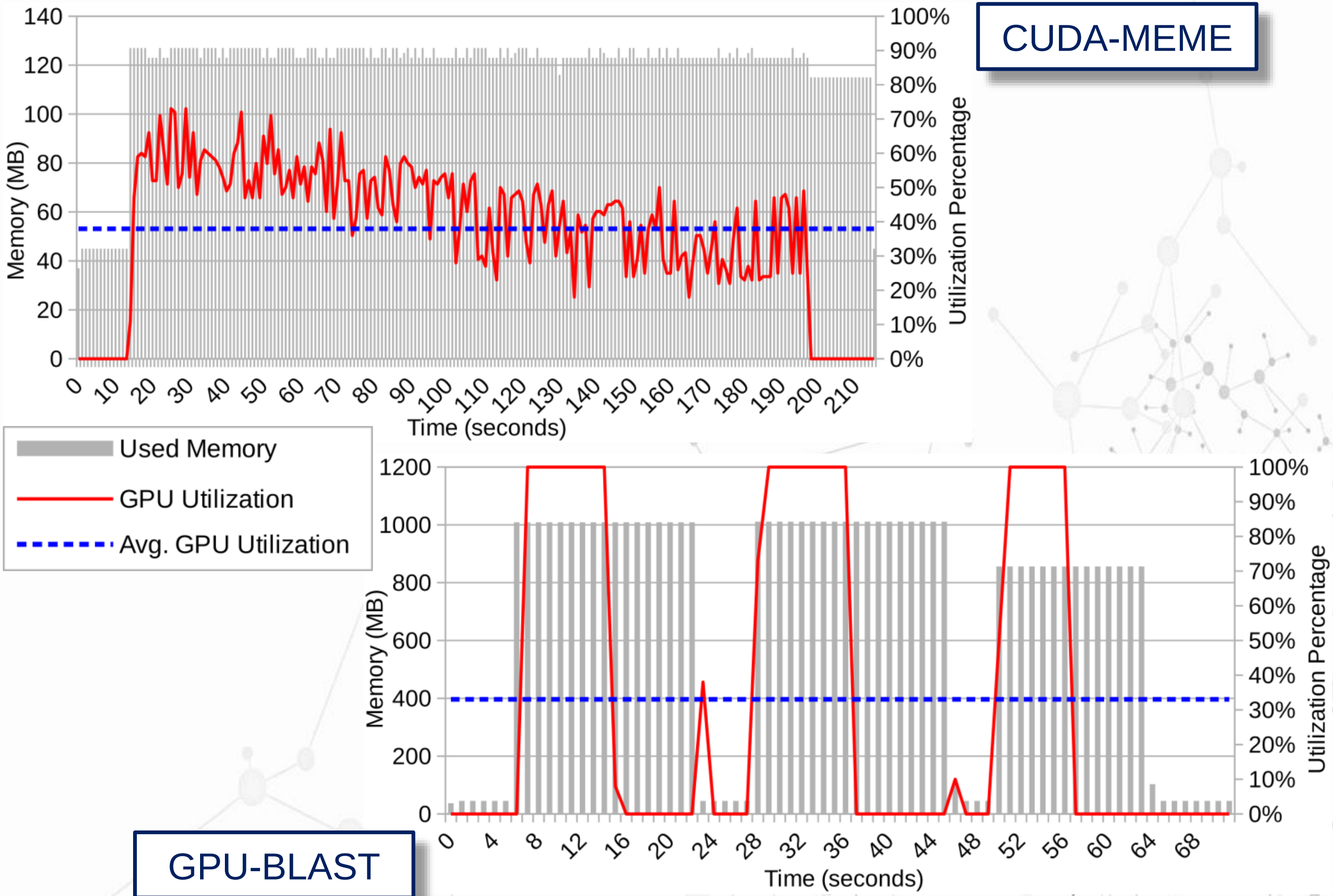
- Non-MPI multi-GPU applications cannot make use of the tremendous GPU resources available across the cluster (even if those GPU resources are idle)

Non-MPI multi-GPU application

All these GPUs cannot be used by the multi-GPU application being executed



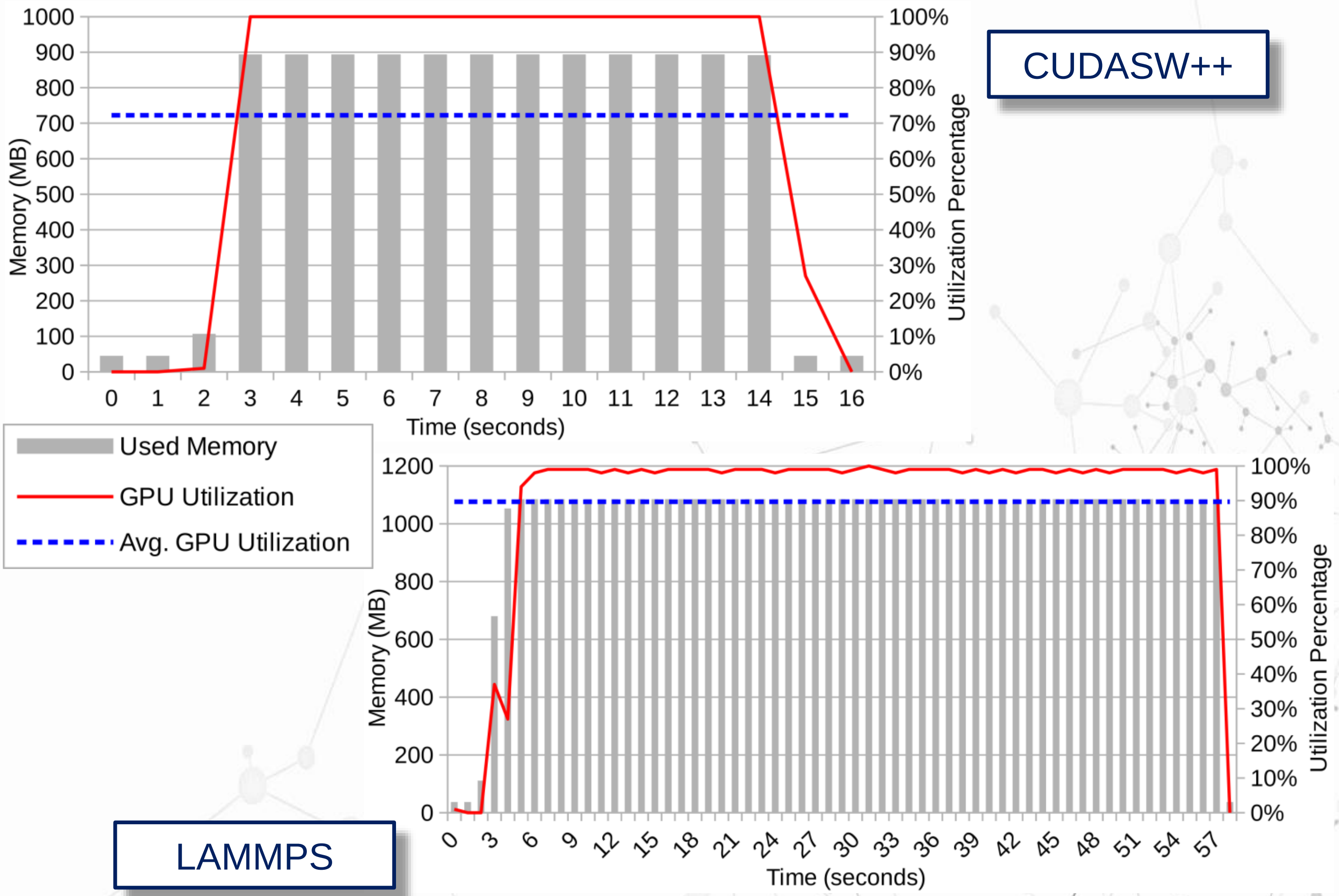
GPU utilization is, in general, low



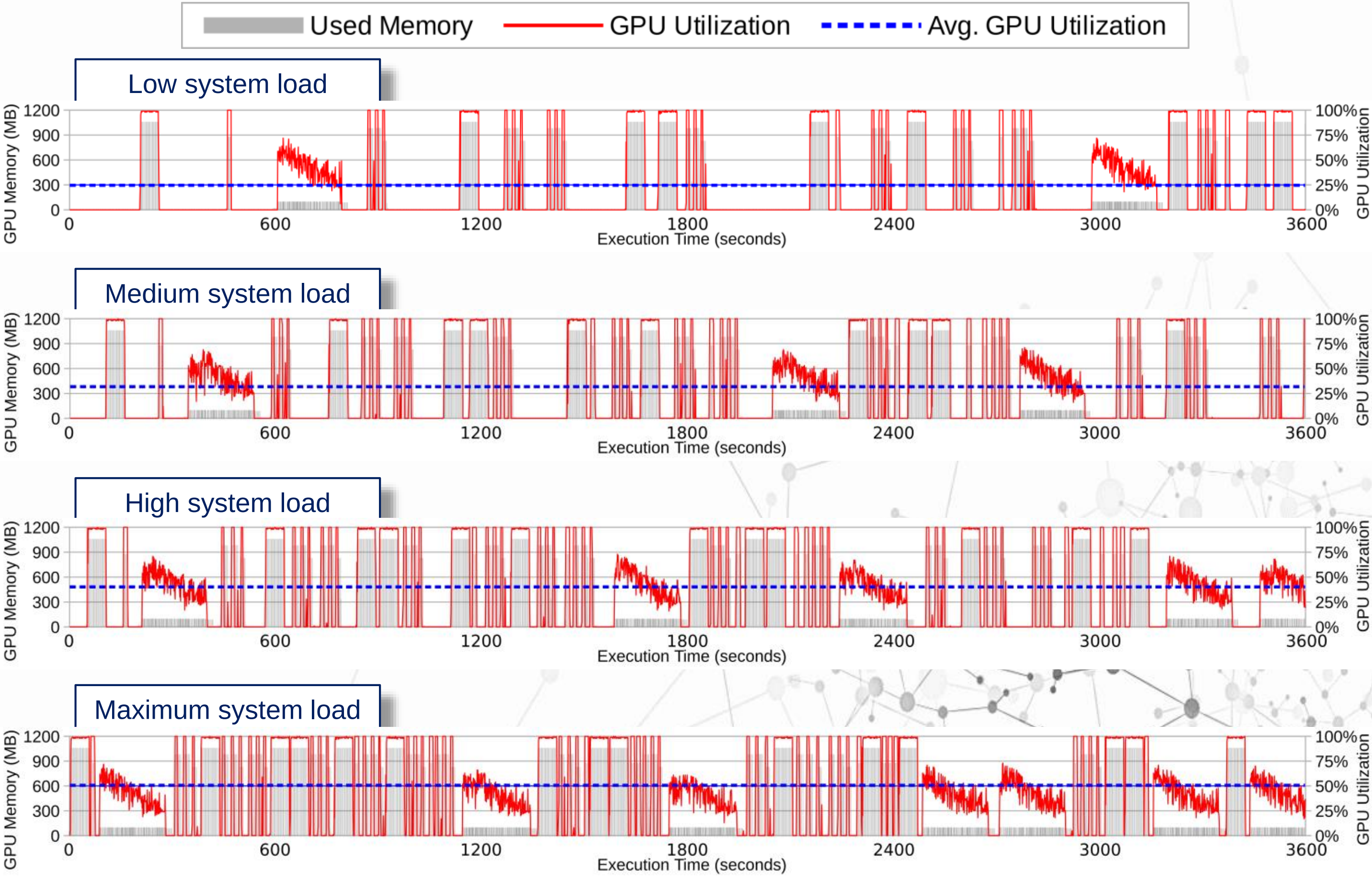
CUDA-MEME

GPU-BLAST

GPU utilization is, in general, low



GPU utilization is, in general, low



Low system load

Medium system load

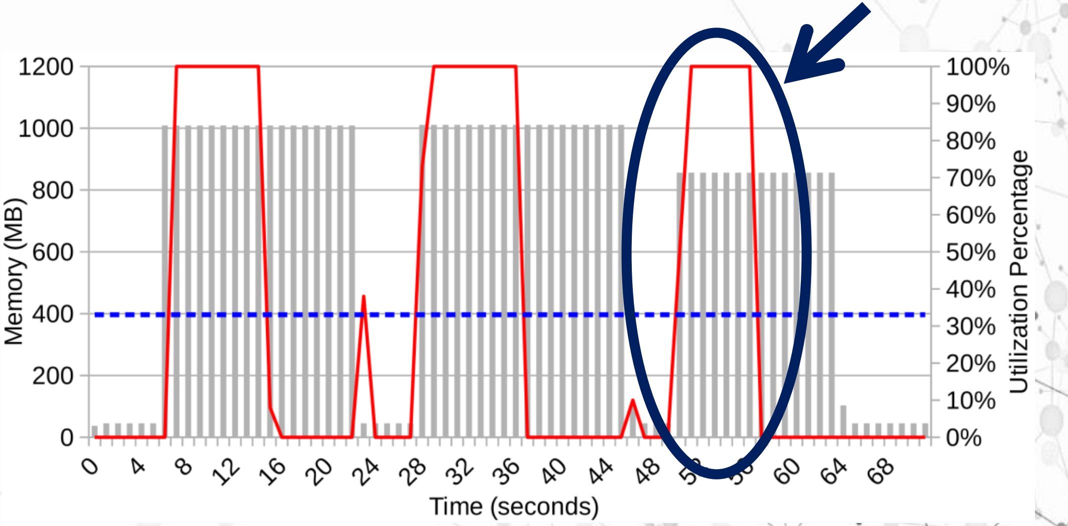
High system load

Maximum system load

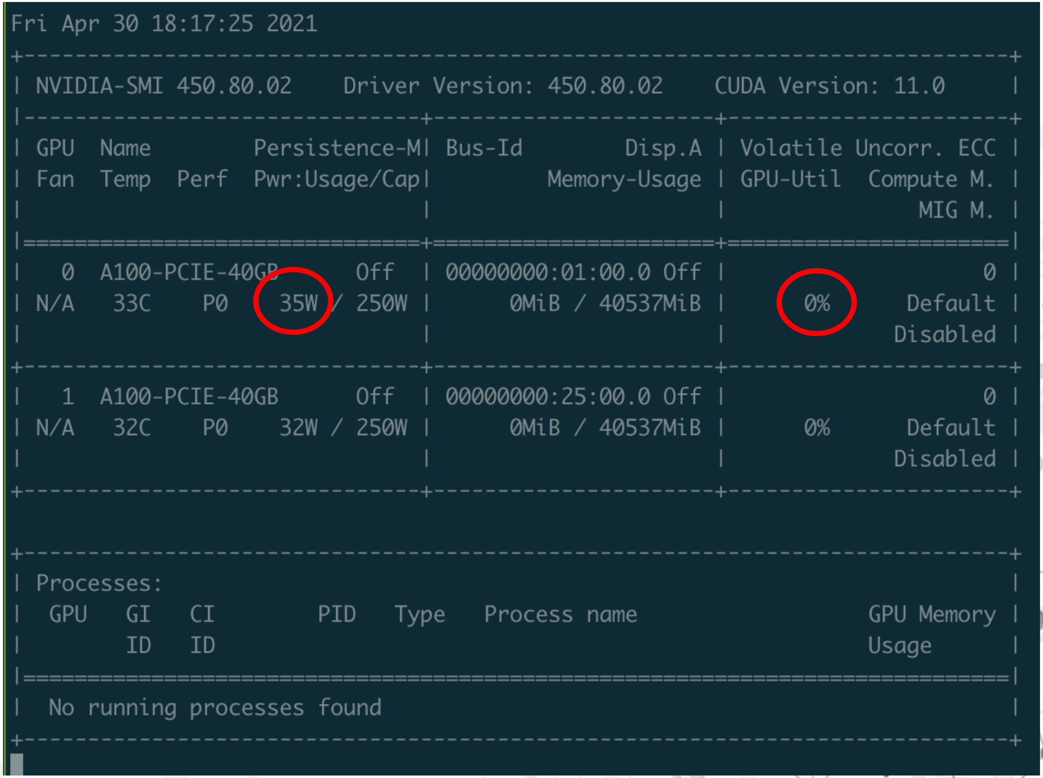
2. GPU utilization is, in general, low

- As GPUs become more powerful, it is expected that current applications will keep them busy for smaller amounts of time

Faster GPUs will complete kernels earlier, but CPUs do not improve that fast



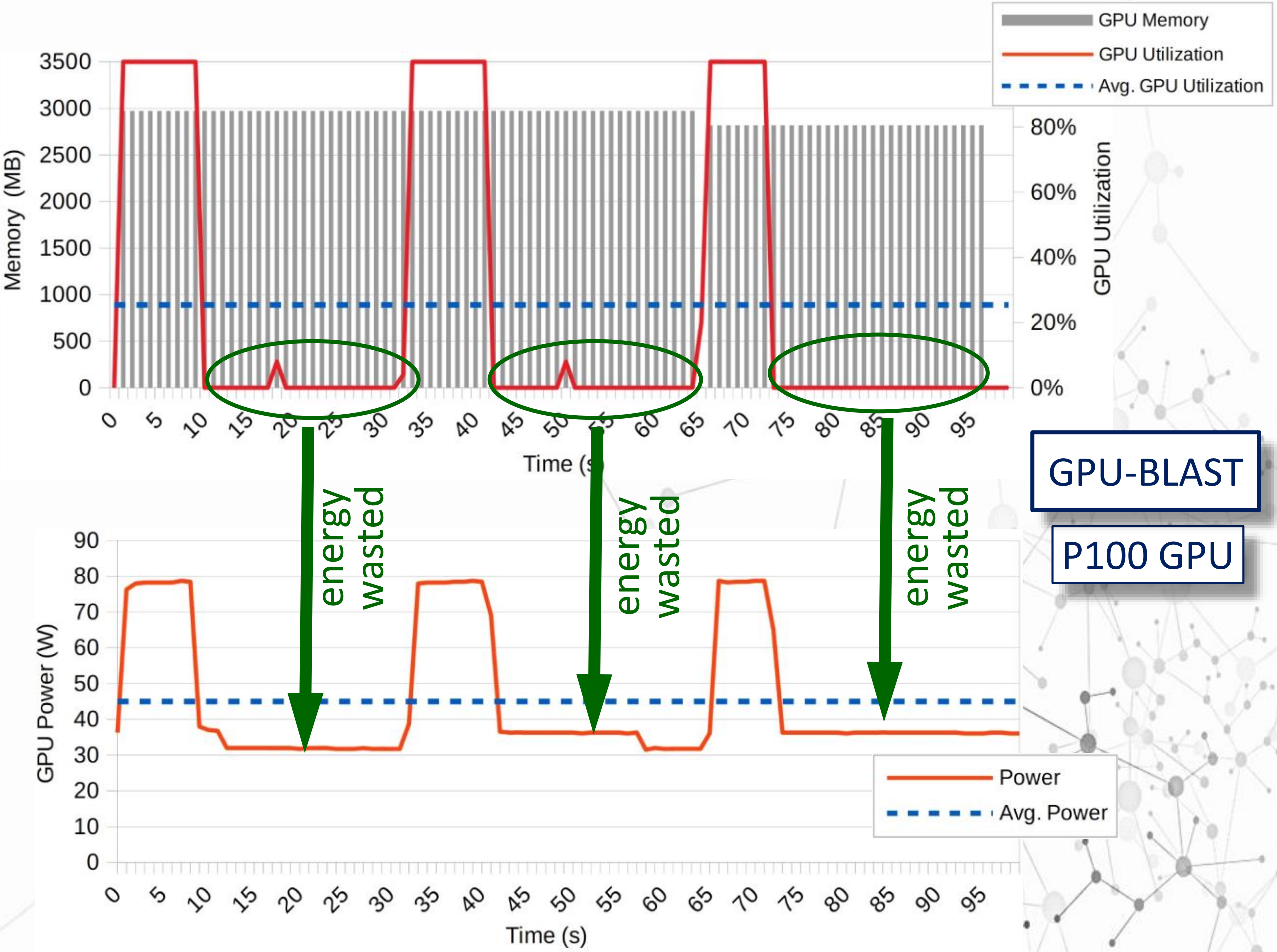
3. GPUs keep consuming energy even when idle



```
Fri Apr 30 18:17:25 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.80.02    Driver Version: 450.80.02    CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100-PCIE-40GB      Off  | 00000000:01:00.0 Off |                    0 |
| N/A   33C    P0    35W / 250W |      0MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  A100-PCIE-40GB      Off  | 00000000:25:00.0 Off |                    0 |
| N/A   32C    P0    32W / 250W |      0MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

GPUs keep consuming energy even when idle



GPU-BLAST

P100 GPU

How to address GPU concerns?

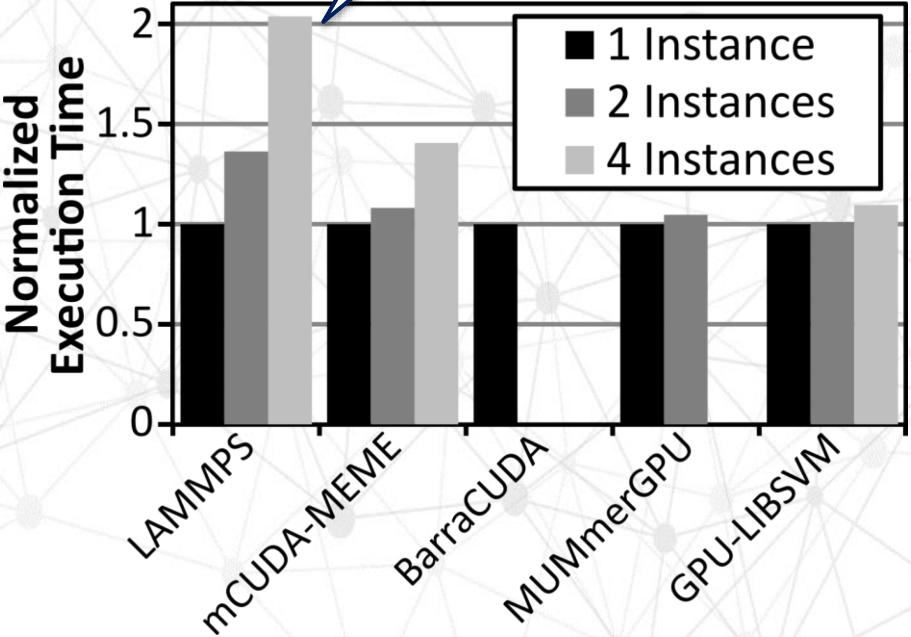
GPU utilization can be increased by **virtualizing** the GPU **and** concurrently **sharing** it among several applications

Overhead of GPU sharing

Time for 4 instances is **only 2x time** for one instance

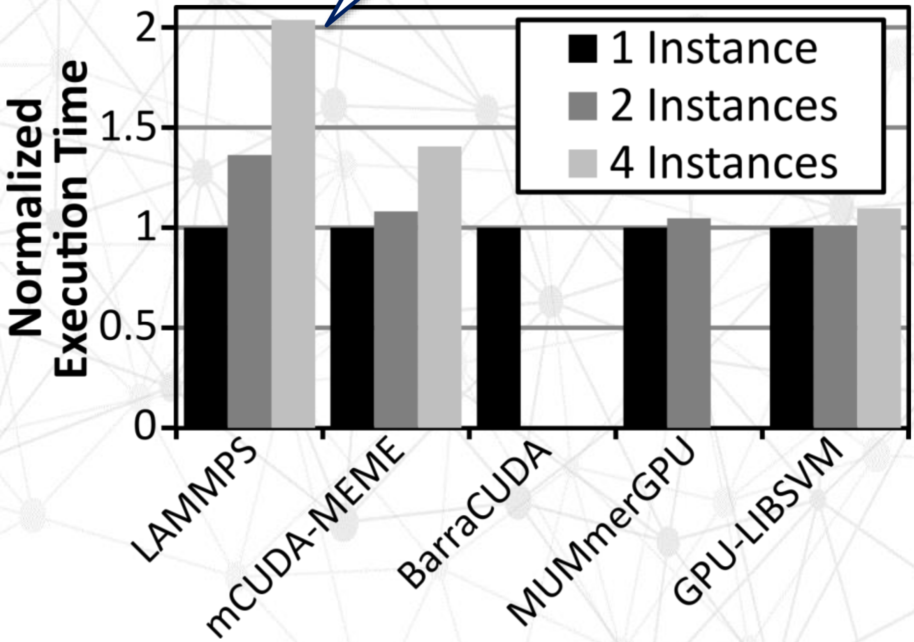
K20 GPU

(5GB memory)



- LAMMPS: 876 MB
- mCUDA-MEME: 151 MB
- BarraCUDA: 3319 MB
- MUMmerGPU: 2104 MB
- GPU-LIBSVM: 145 MB

Overhead of GPU sharing



Time for 4 instances is **only 2x time** for one instance

K20 GPU

(5GB memory)

- LAMMPS: 876 MB
- mCUDA-MEME: 151 MB
- BarraCUDA: 3319 MB
- MUMmerGPU: 2104 MB
- GPU-LIBSVM: 145 MB

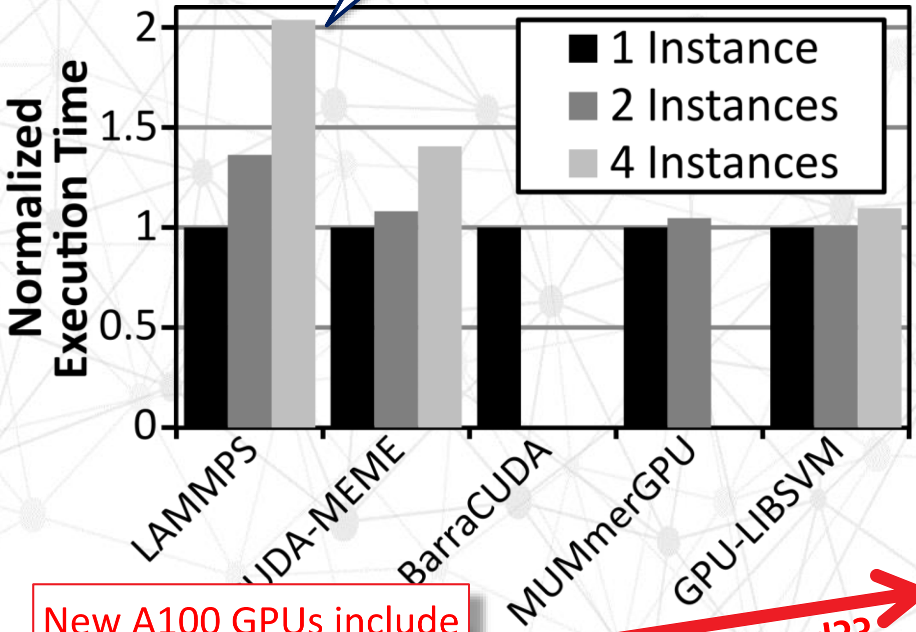
The main concern for sharing a GPU is the memory limitation

Overhead of GPU sharing

Time for 4 instances is **only 2x time** for one instance

K20 GPU

(5GB memory)



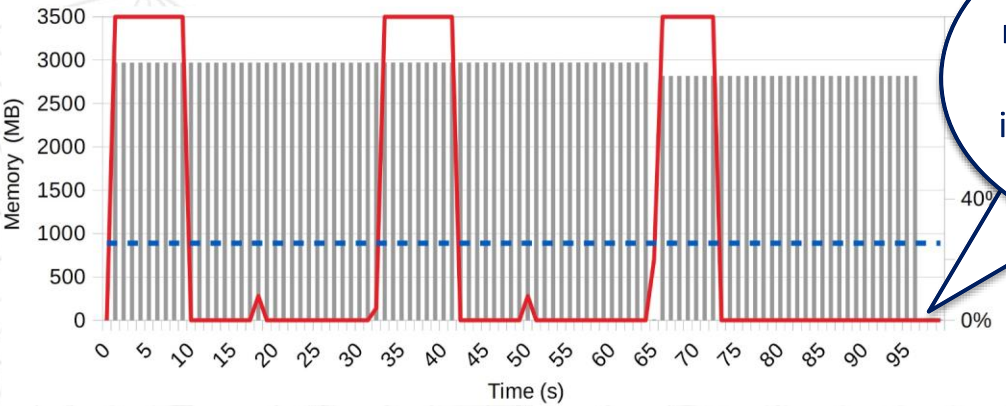
- LAMMPS: 876 MB
- mCUDA-MEME: 151 MB
- BarraCUDA: 3319 MB
- MUMmerGPU: 2104 MB
- GPU-LIBSVM: 145 MB

The main concern for sharing a GPU is the memory limitation

New A100 GPUs include 40 GB of memory

Limitation removed??

Overhead of GPU sharing

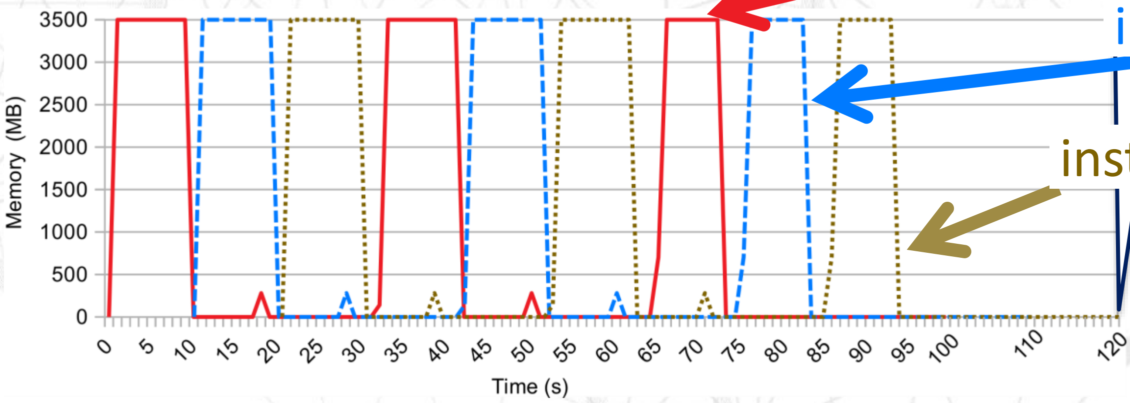


One instance requires 100 seconds while 3 concurrent instances require 120 seconds!!

instance 1

instance 2

instance 3

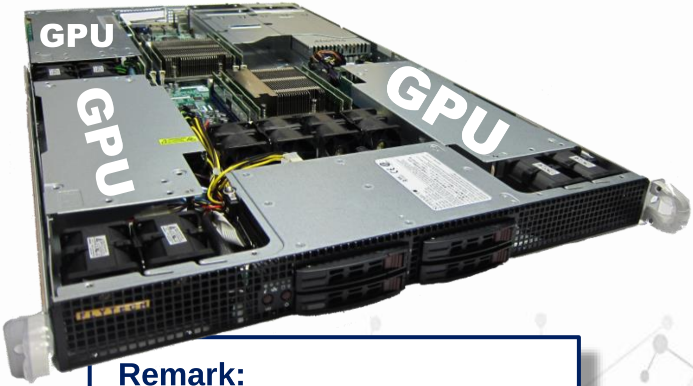


This plot is an estimation! It is not accurate

What is rCUDA?

Basics of GPU computing

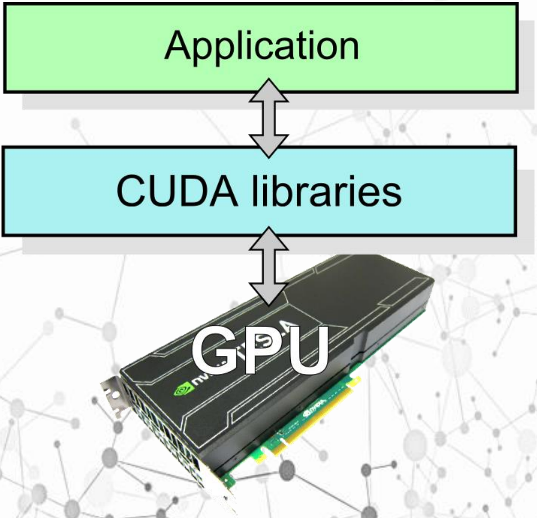
Basic behavior of CUDA



Application

CUDA libraries

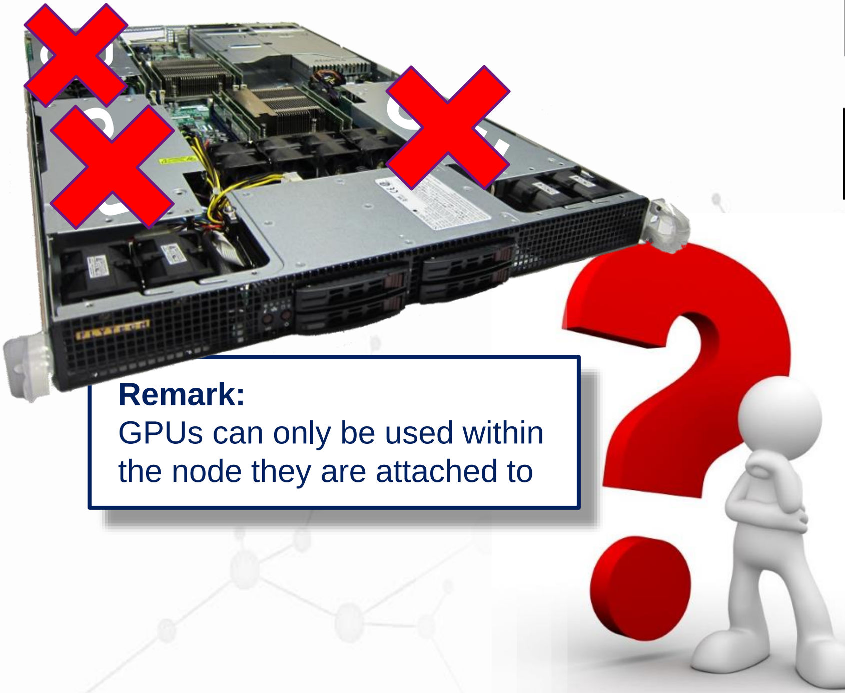
GPU



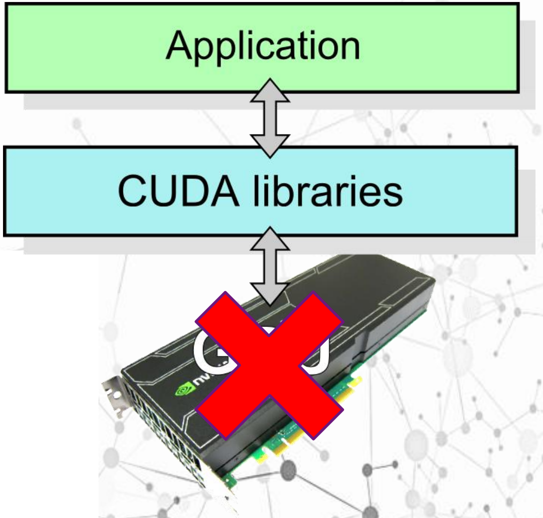
Remark:
GPUs can only be used within the node they are attached to

Basics of GPU computing

Basic behavior of CUDA



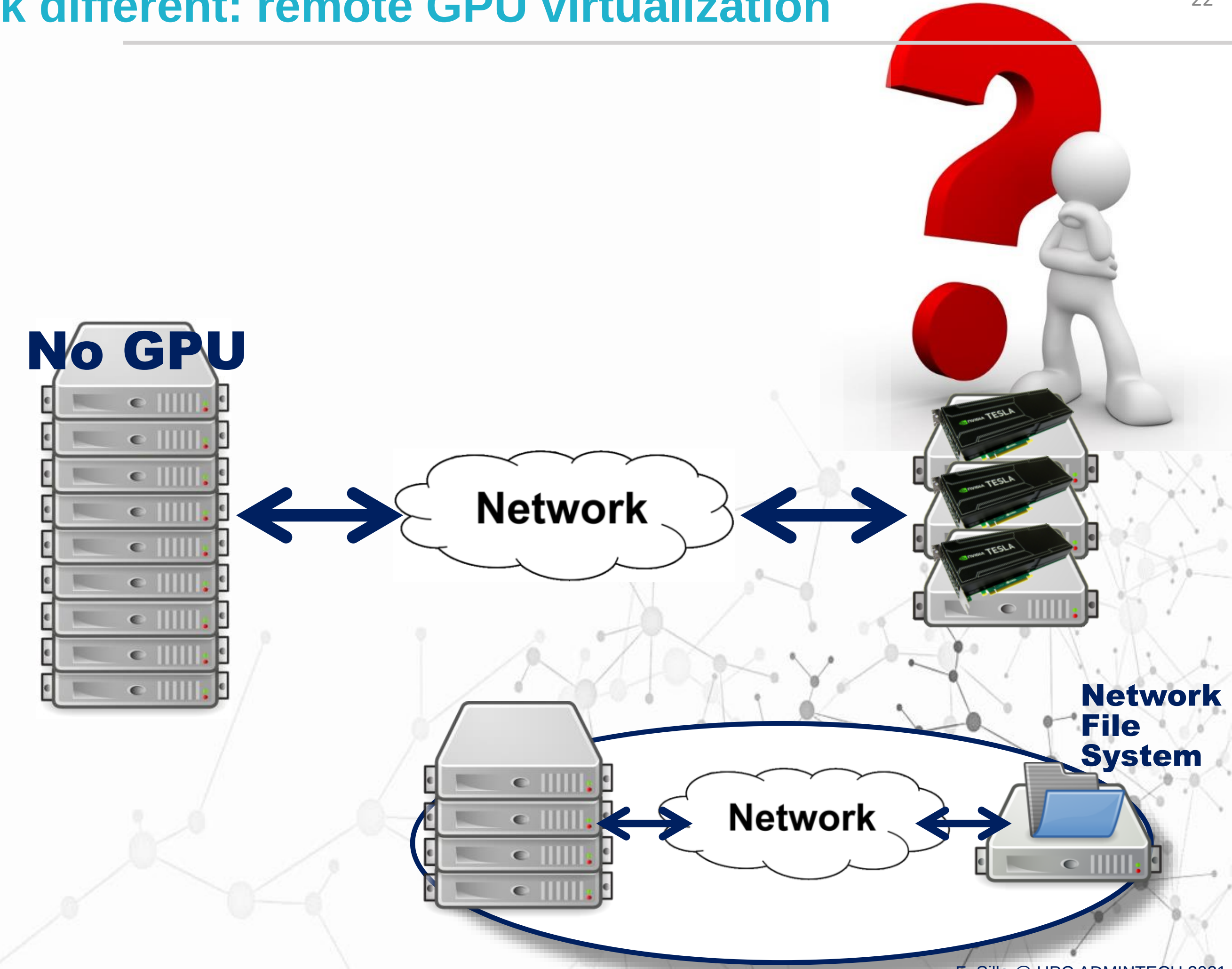
Remark:
GPUs can only be used within the node they are attached to



Application

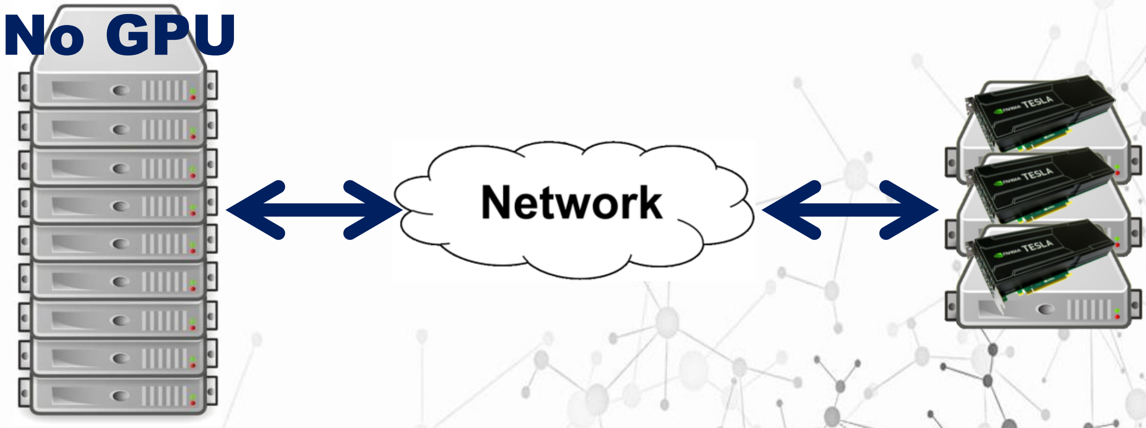
CUDA libraries

Think different: remote GPU virtualization



Think different: remote GPU virtualization

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



rCUDA … remote CUDA

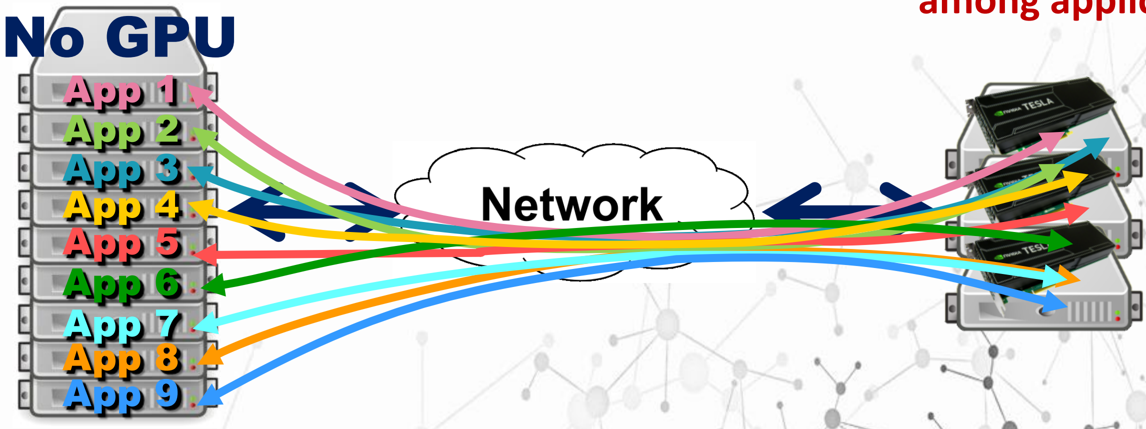
rCUDA is a development by Universitat Politècnica de València

rCUDA remote CUDA

Think different: remote GPU virtualization

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

GPUs are shared among applications



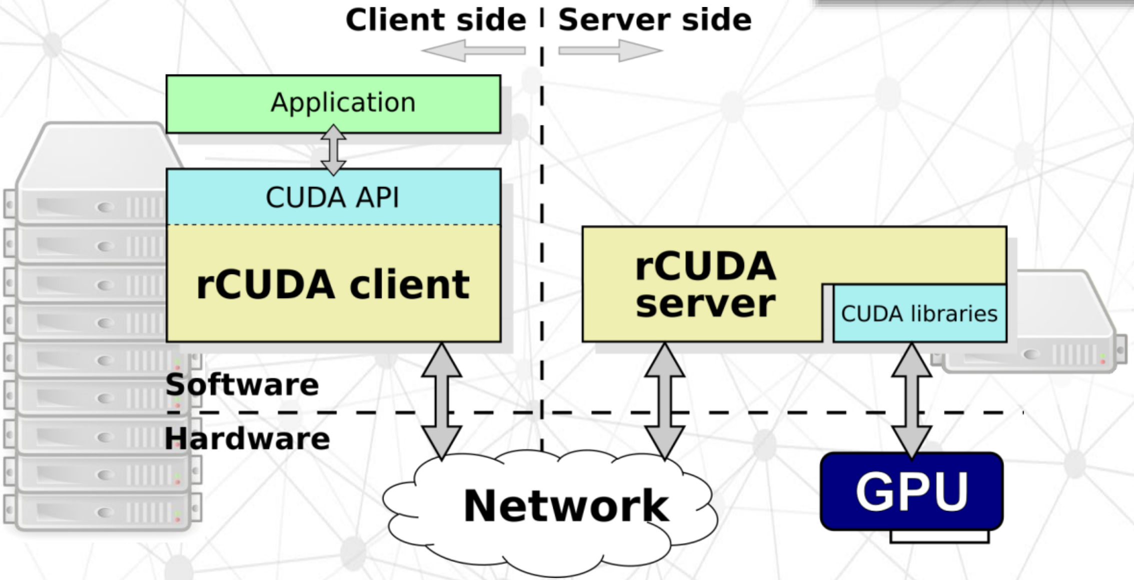
rCUDA … remote CUDA

rCUDA is a development by Universitat Politècnica de València

rCUDA remote CUDA

Basics or rCUDA

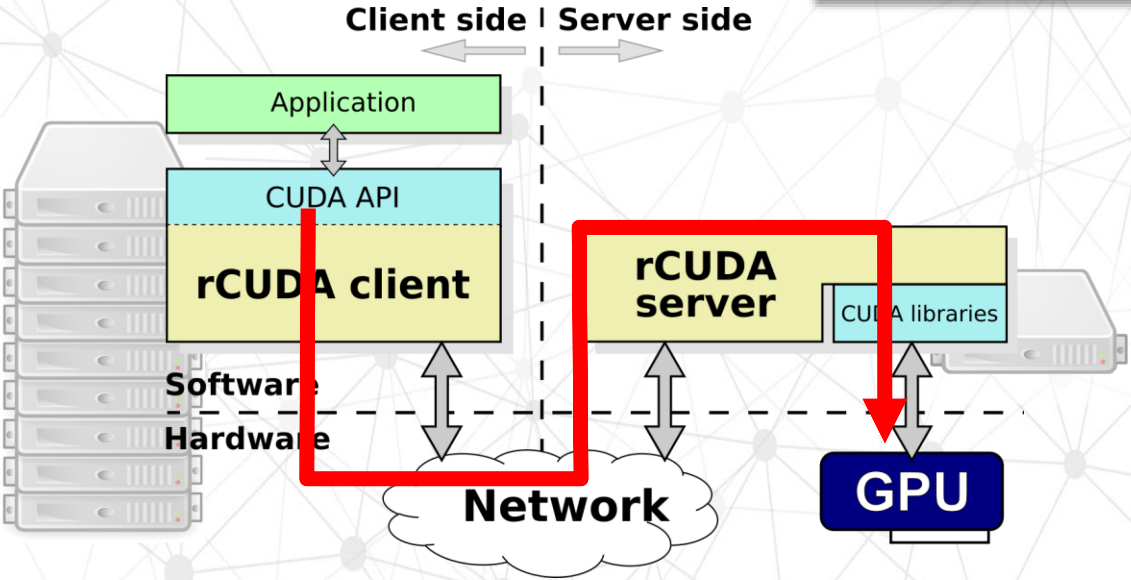
Access to remote GPU is **transparent** to applications: no source code modification is needed



rCUDA is a development by Universitat Politècnica de València

Basics or rCUDA

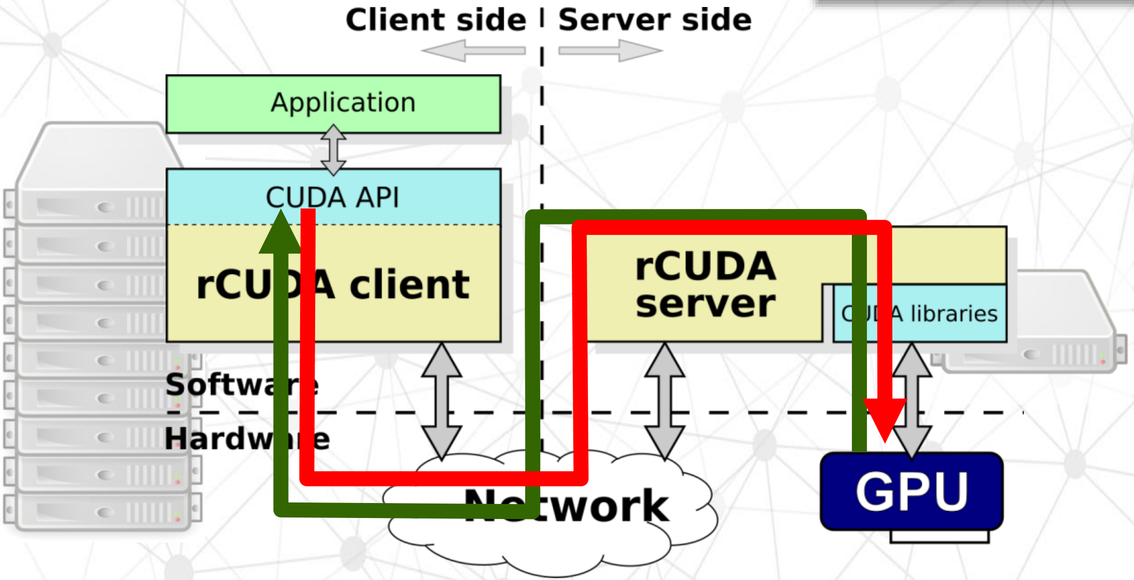
Access to remote GPU is **transparent** to applications: no source code modification is needed



rCUDA is a development by Universitat Politècnica de València

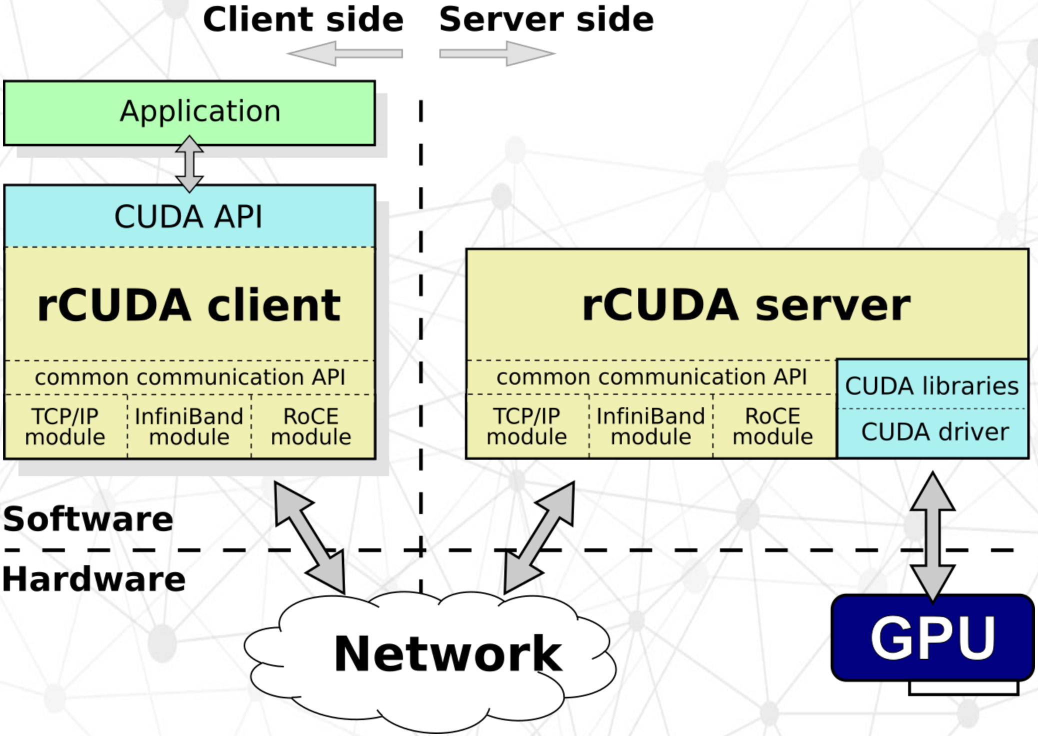
Basics or rCUDA

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



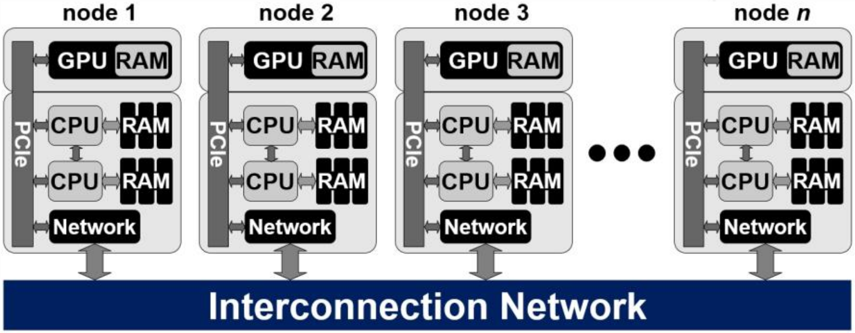
rCUDA is a development by Universitat Politècnica de València

rCUDA supports RDMA transfers



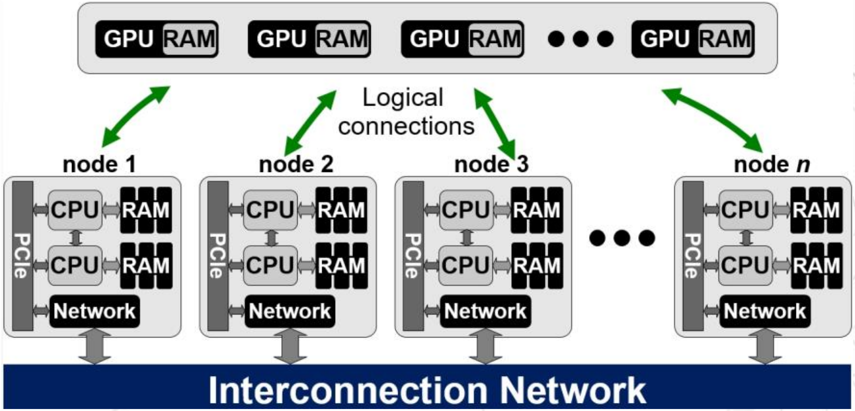
rCUDA envision

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



Physical configuration

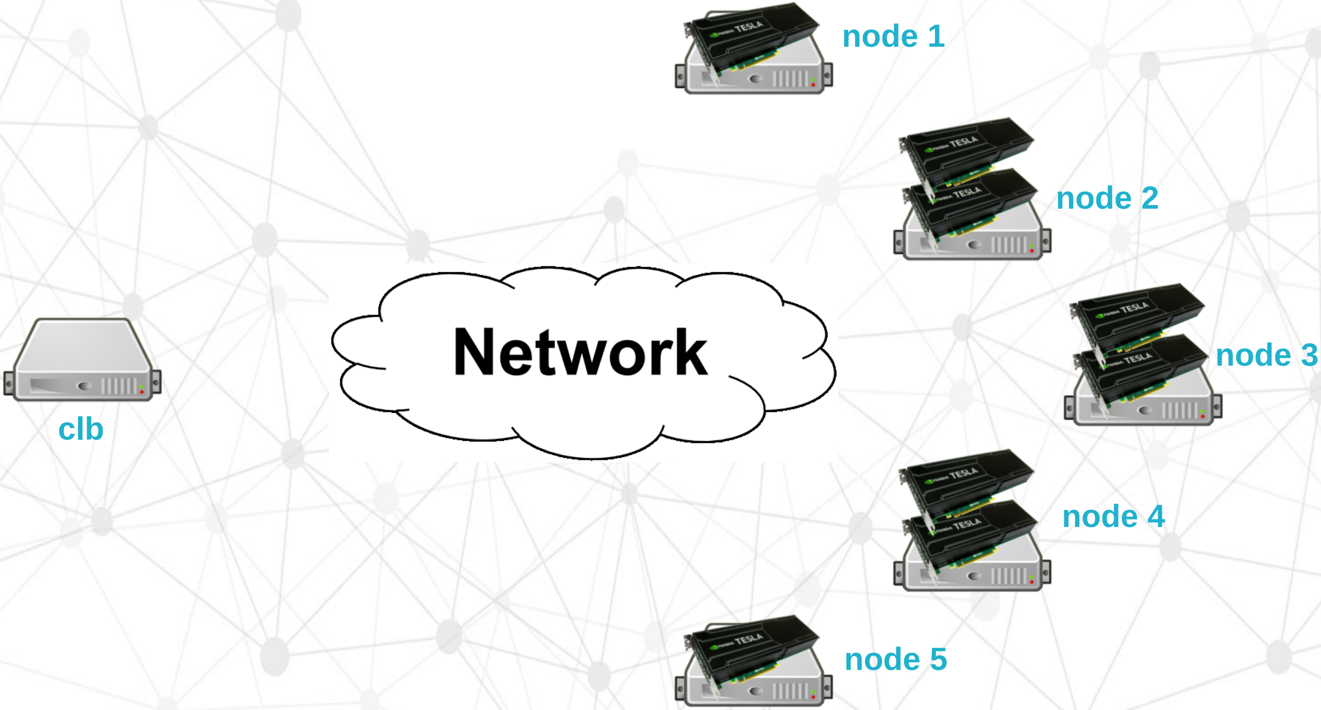
… to the following one:



Logical configuration

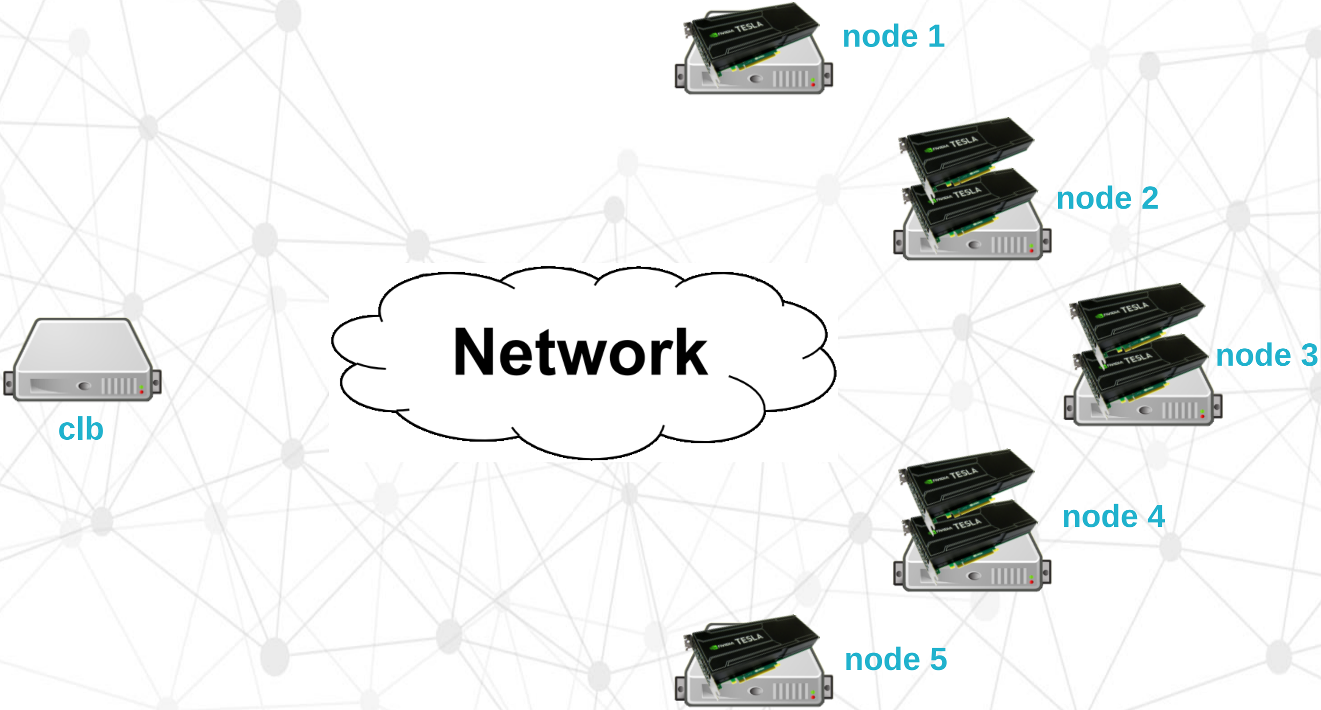
How is rCUDA deployed in a cluster?

Deploying rCUDA in a cluster



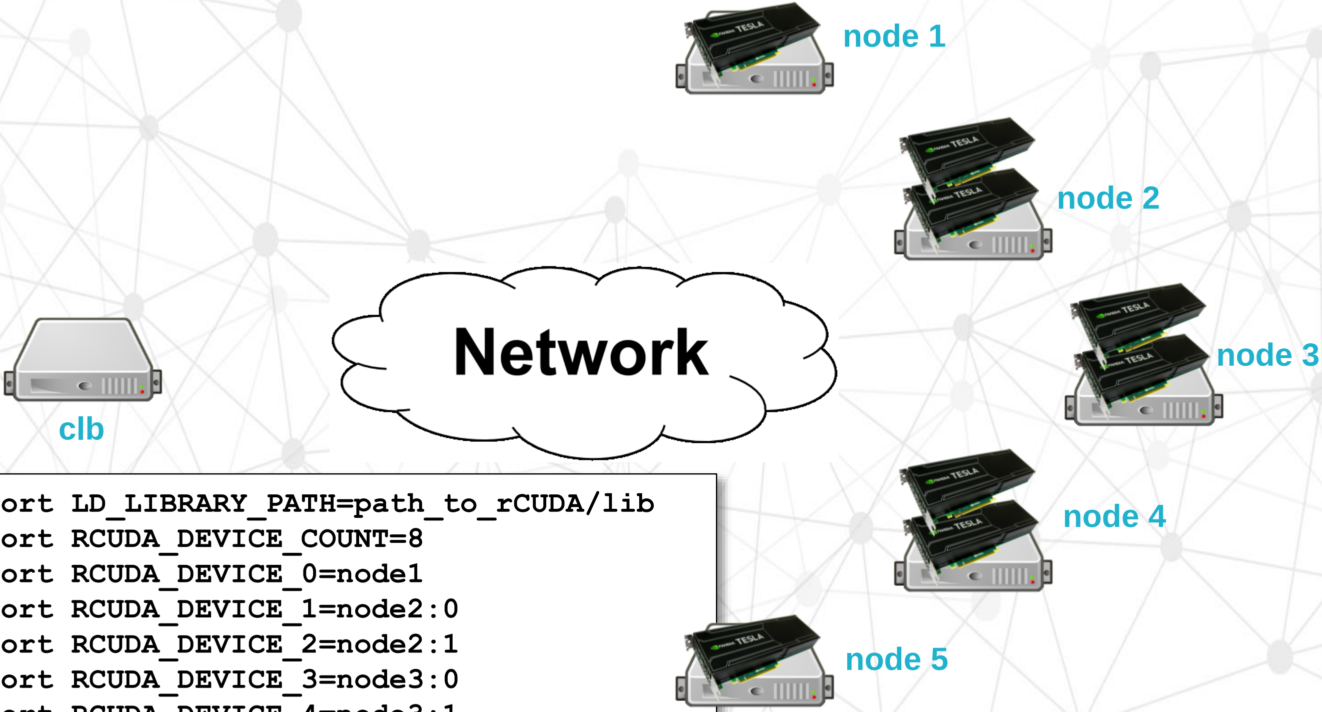
Starting the server side

```
export LD_LIBRARY_PATH=path_to_CUDA
./rCUDAd
```



Starting the client side

```
export LD_LIBRARY_PATH=path_to_CUDA
./rCUDAd
```

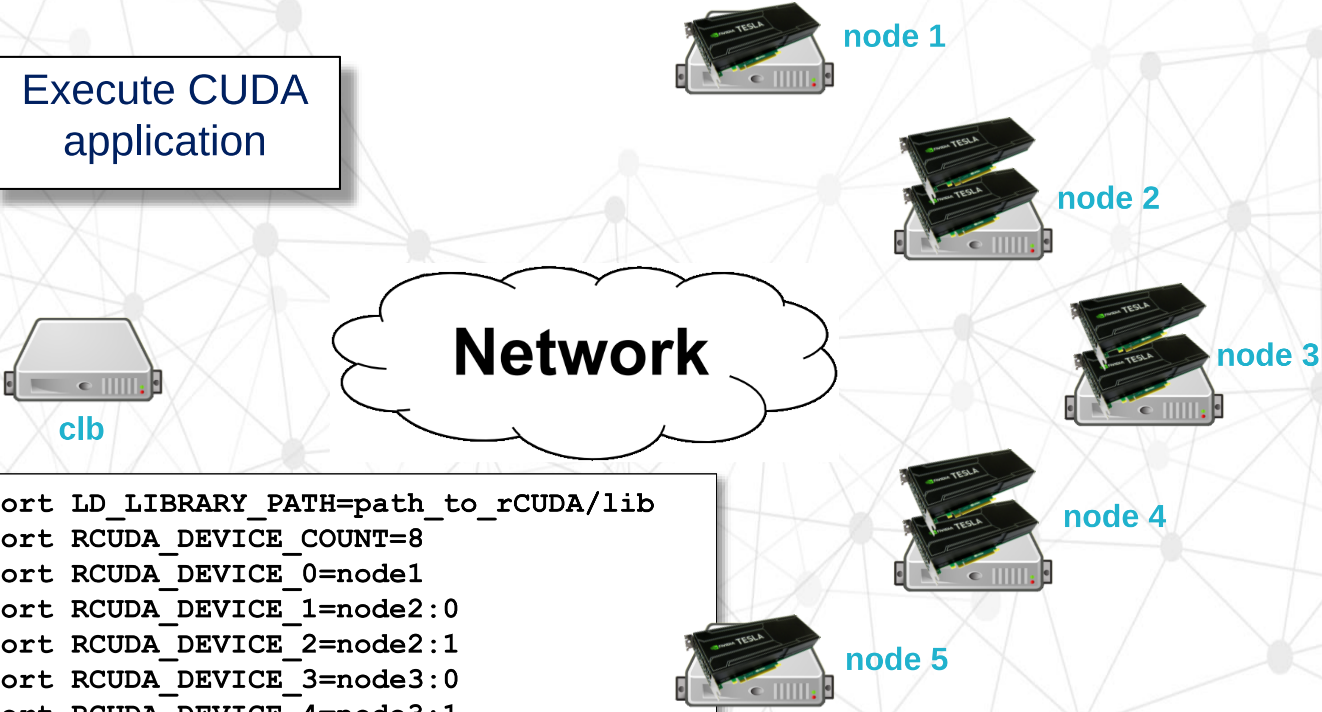


```
export LD_LIBRARY_PATH=path_to_rCUDA/lib
export RCUDA_DEVICE_COUNT=8
export RCUDA_DEVICE_0=node1
export RCUDA_DEVICE_1=node2:0
export RCUDA_DEVICE_2=node2:1
export RCUDA_DEVICE_3=node3:0
export RCUDA_DEVICE_4=node3:1
export RCUDA_DEVICE_5=node4:0
export RCUDA_DEVICE_6=node4:1
export RCUDA_DEVICE_7=node5
```

Executing the application

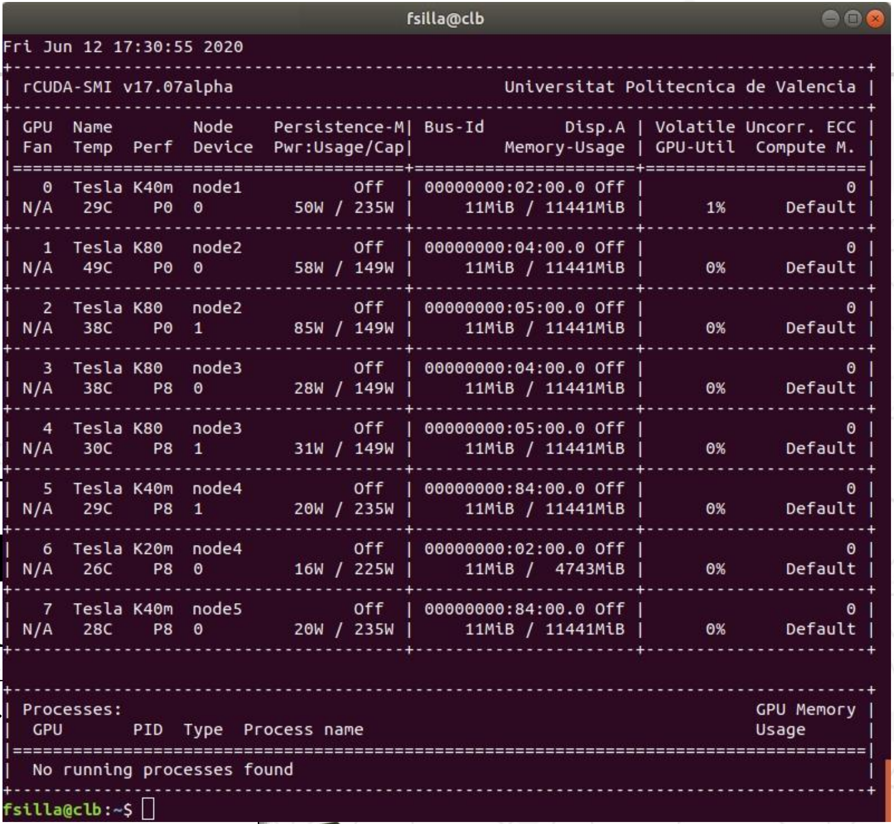
```
export LD_LIBRARY_PATH=path_to_CUDA
./rCUDAd
```

Execute CUDA application



```
export LD_LIBRARY_PATH=path_to_rCUDA/lib
export RCUDA_DEVICE_COUNT=8
export RCUDA_DEVICE_0=node1
export RCUDA_DEVICE_1=node2:0
export RCUDA_DEVICE_2=node2:1
export RCUDA_DEVICE_3=node3:0
export RCUDA_DEVICE_4=node3:1
export RCUDA_DEVICE_5=node4:0
export RCUDA_DEVICE_6=node4:1
export RCUDA_DEVICE_7=node5
```

rCUDA-smi tool



```
Fri Jun 12 17:30:55 2020
+-----------------------------------------------------------------------------+
| rCUDA-SMI v17.07alpha                      Universitat Politecnica de Valencia |
+-----------------------------------------------------------------------------+
| GPU  Name        Node    Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Device  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|=======================================+======================+======================|
|   0  Tesla K40m  node1           Off  | 00000000:02:00.0 Off |                    0 |
| N/A   29C    P0  0          50W / 235W |     11MiB / 11441MiB |      1%      Default |
+---------------------------------------+----------------------+----------------------+
|   1  Tesla K80   node2           Off  | 00000000:04:00.0 Off |                    0 |
| N/A   49C    P0  0          58W / 149W |     11MiB / 11441MiB |      0%      Default |
+---------------------------------------+----------------------+----------------------+
|   2  Tesla K80   node2           Off  | 00000000:05:00.0 Off |                    0 |
| N/A   38C    P0  1          85W / 149W |     11MiB / 11441MiB |      0%      Default |
+---------------------------------------+----------------------+----------------------+
|   3  Tesla K80   node3           Off  | 00000000:04:00.0 Off |                    0 |
| N/A   38C    P8  0          28W / 149W |     11MiB / 11441MiB |      0%      Default |
+---------------------------------------+----------------------+----------------------+
|   4  Tesla K80   node3           Off  | 00000000:05:00.0 Off |                    0 |
| N/A   30C    P8  1          31W / 149W |     11MiB / 11441MiB |      0%      Default |
+---------------------------------------+----------------------+----------------------+
|   5  Tesla K40m  node4           Off  | 00000000:84:00.0 Off |                    0 |
| N/A   29C    P8  1          20W / 235W |     11MiB / 11441MiB |      0%      Default |
+---------------------------------------+----------------------+----------------------+
|   6  Tesla K20m  node4           Off  | 00000000:02:00.0 Off |                    0 |
| N/A   26C    P8  0          16W / 225W |     11MiB /  4743MiB |      0%      Default |
+---------------------------------------+----------------------+----------------------+
|   7  Tesla K40m  node5           Off  | 00000000:84:00.0 Off |                    0 |
| N/A   28C    P8  0          20W / 235W |     11MiB / 11441MiB |      0%      Default |
+---------------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
fsilla@clb:~$ 
```

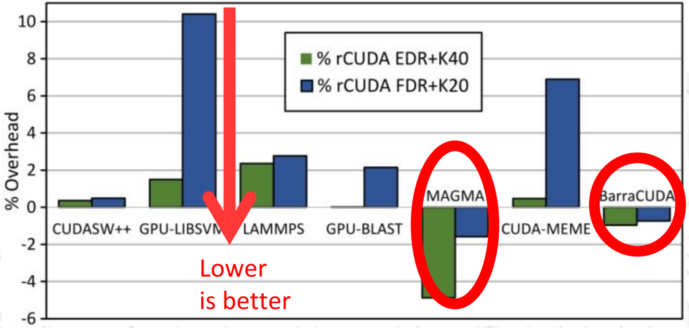
```
export LD_LIBRARY_PATH=path_t
export RCUDA_DEVICE_COUNT=8
export RCUDA_DEVICE_0=node1
export RCUDA_DEVICE_1=node2:0
export RCUDA_DEVICE_2=node2:1
export RCUDA_DEVICE_3=node3:0
export RCUDA_DEVICE_4=node3:1
export RCUDA_DEVICE_5=node4:0
export RCUDA_DEVICE_6=node4:1
export RCUDA_DEVICE_7=node5
```

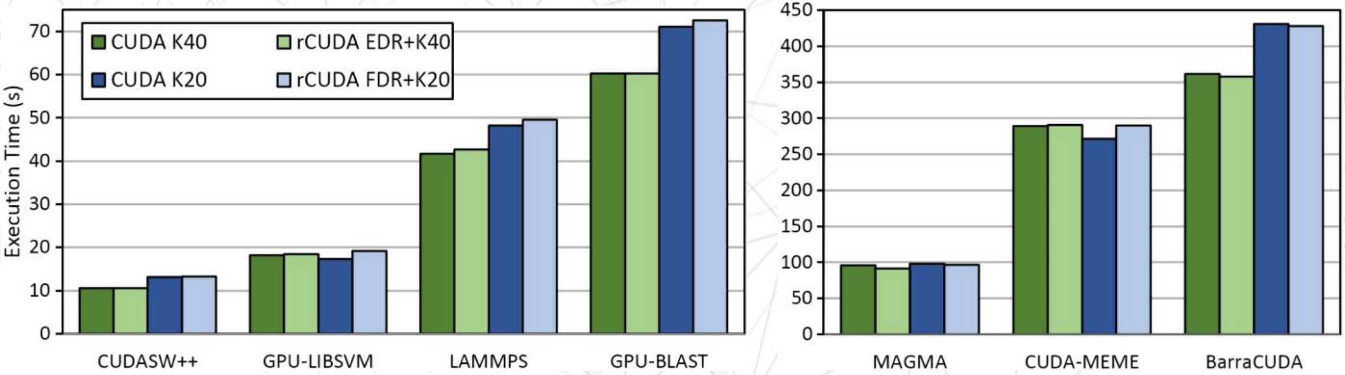
Performance of rCUDA?

(local PCIe link replaced by a network fabric)

Performance of rCUDA

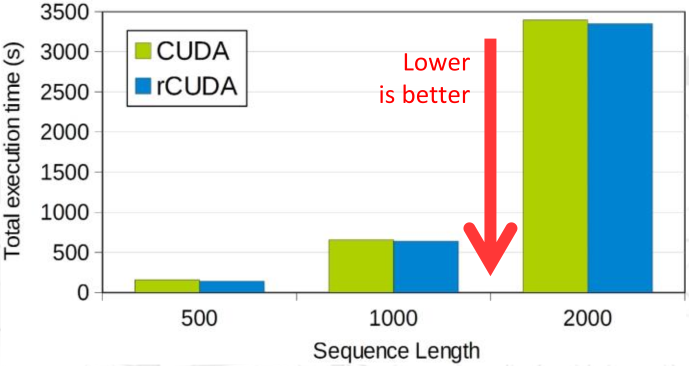
- K20 GPU and FDR InfiniBand
- K40 GPU and EDR InfiniBand





Performance of rCUDA

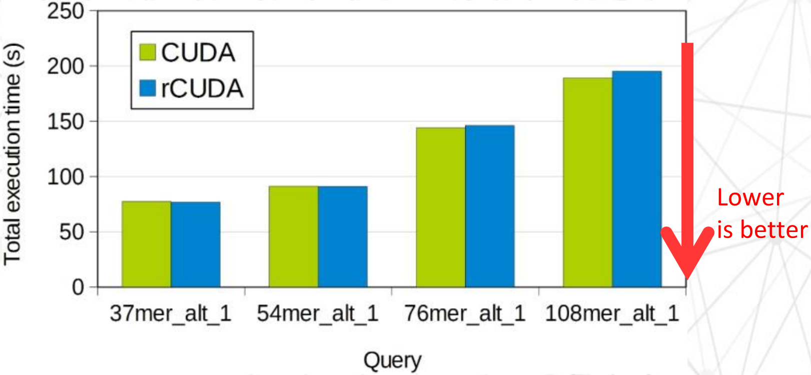
P100 GPU and EDR InfiniBand



Lower
is better

CUDA-MEME

BarraCUDA

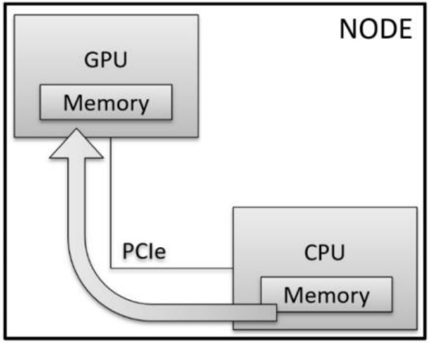


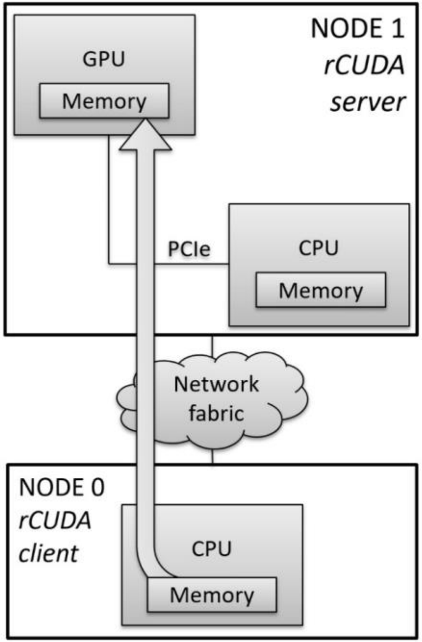
Lower
is better

Performance of data movements to/from GPUs

CUDA

rCUDA

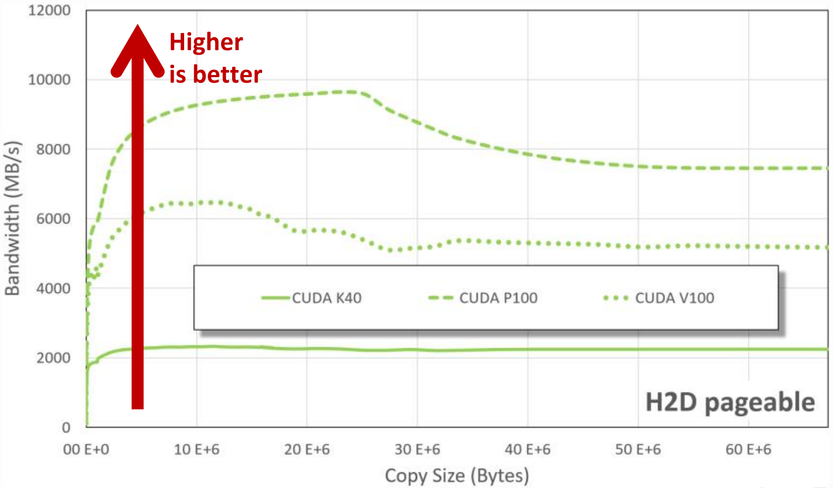




- CPU to GPU (Host to Device)
  - H2D pinned
  - **H2D pageable**
- GPU to CPU (Device to Host)
  - D2H pinned
  - **D2H pageable**

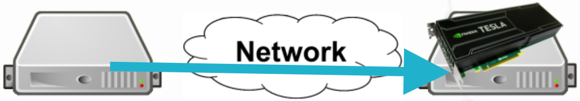
Used by applications

Performance of data movements to/from GPUs

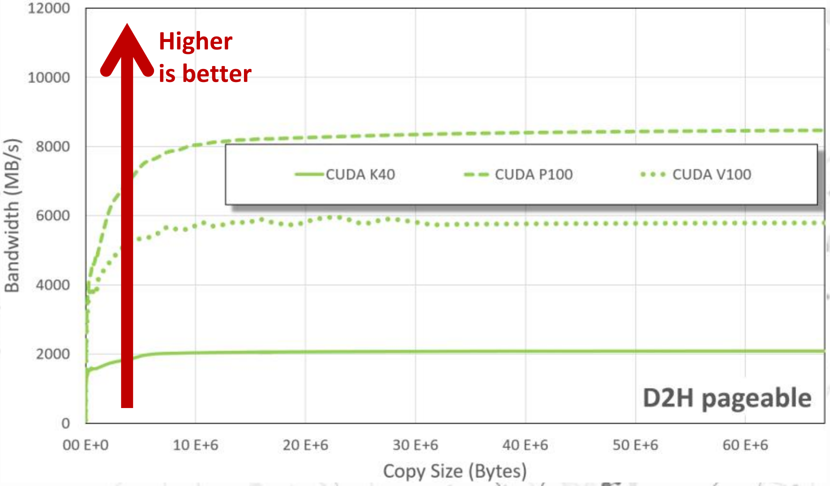


Higher is better

H2D pageable

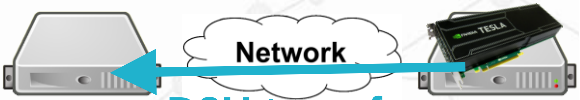


H2D transfer



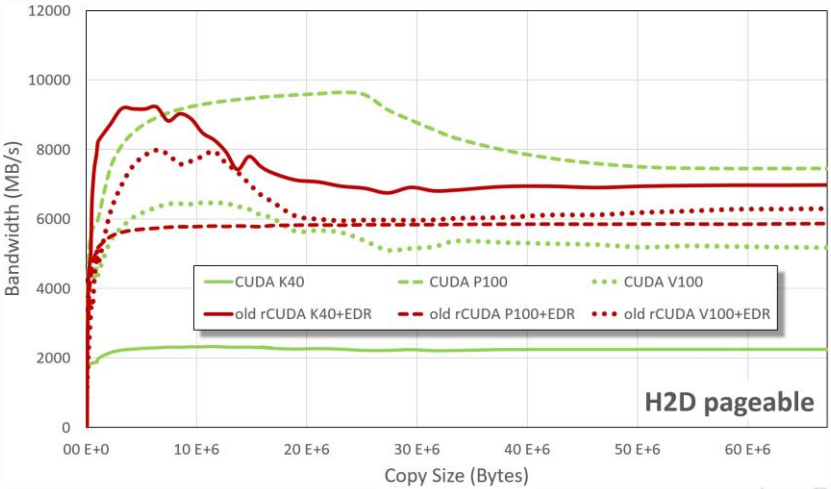
Higher is better

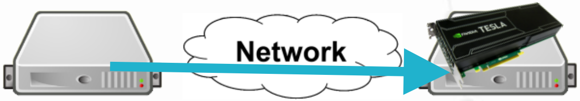
D2H pageable



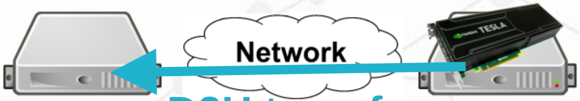
D2H transfer

Performance of data movements to/from GPUs

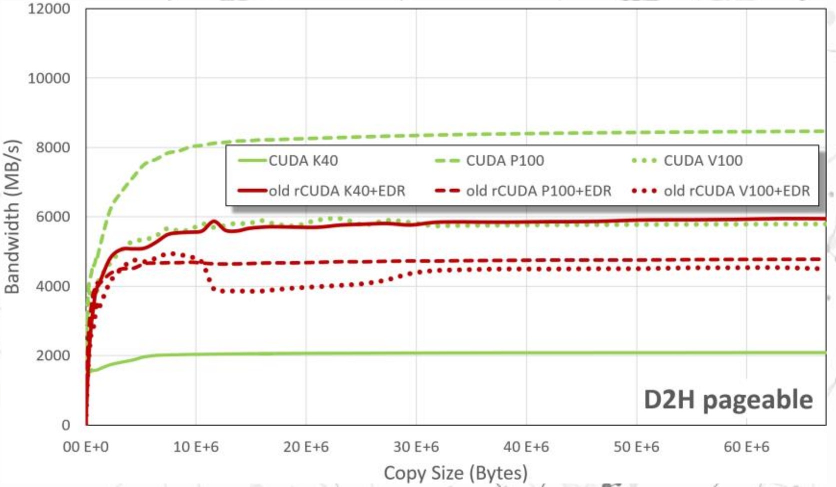




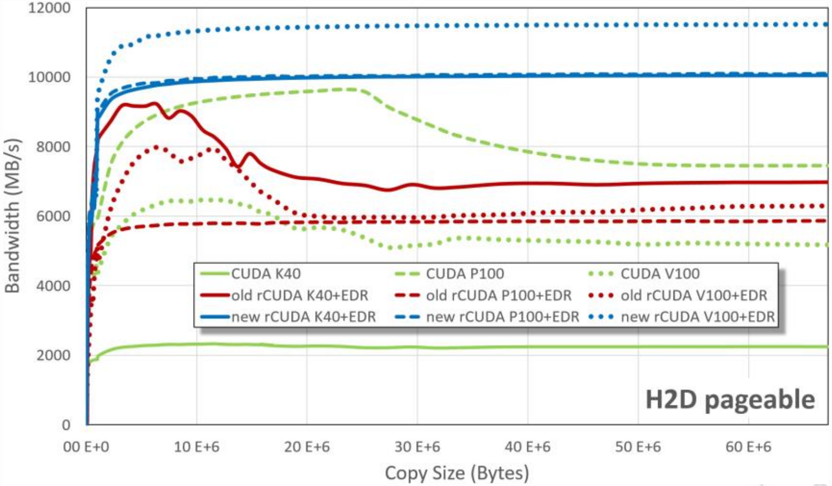
H2D transfer

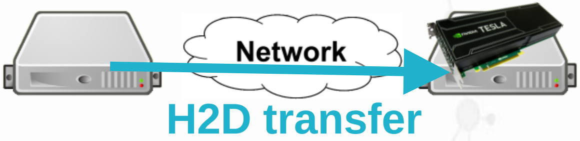


D2H transfer



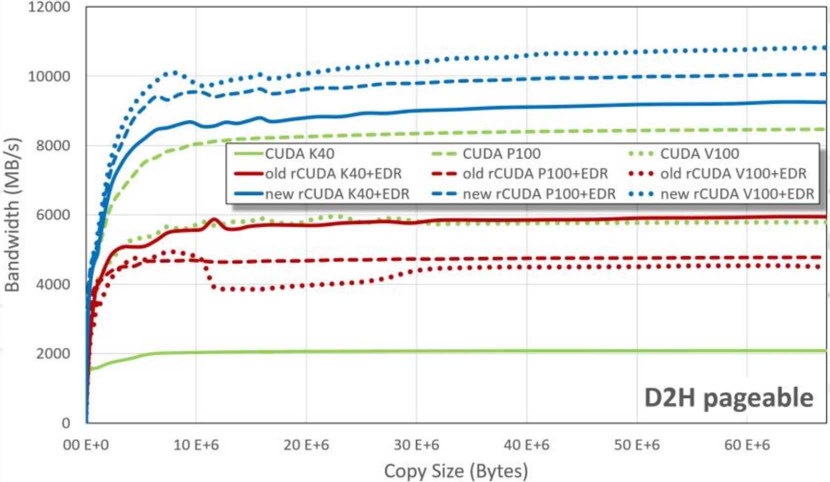
Performance of data movements to/from GPUs

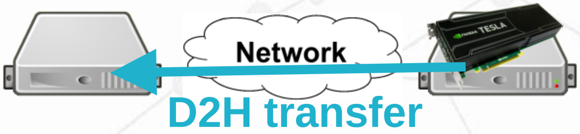




Network

H2D transfer

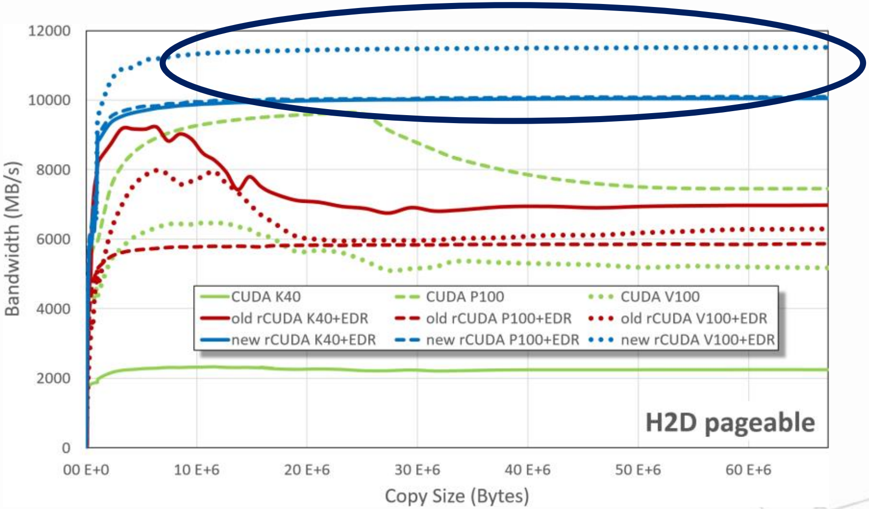




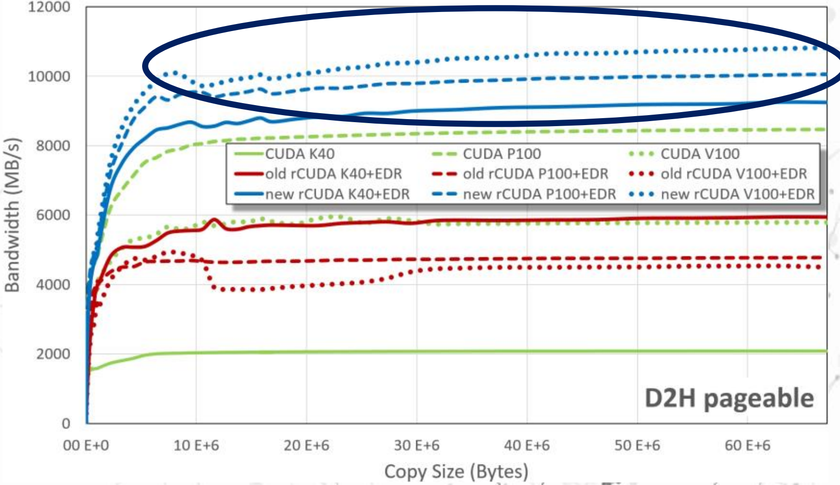
Network

D2H transfer

Performance of data movements to/from GPUs



This performance is possible thanks to the use of InfiniBand

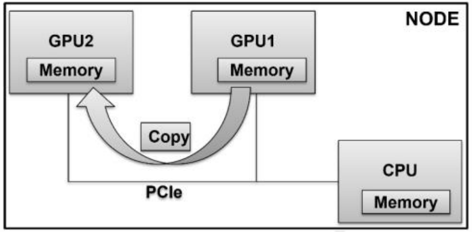


New communication layer:

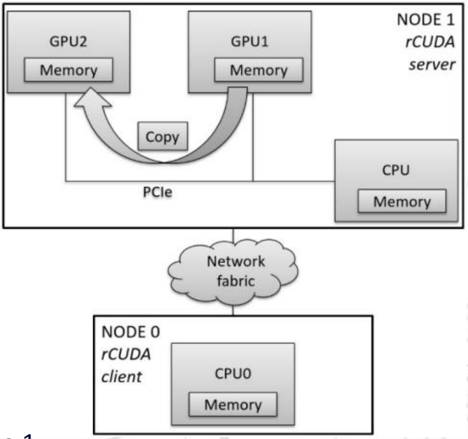
- **<u>2x</u> performance over old communication layer**
- **better performance than CUDA in all cases**

Performance of data movements among GPUs

CUDA

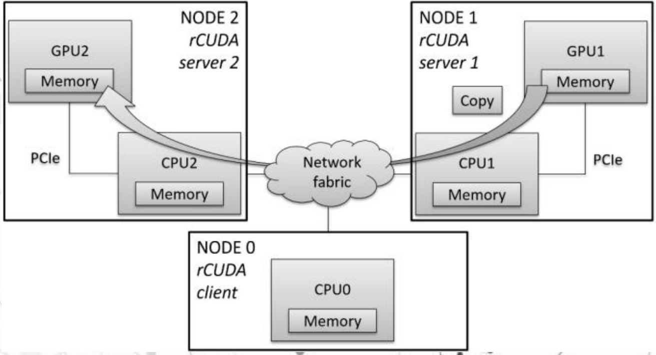


rCUDA

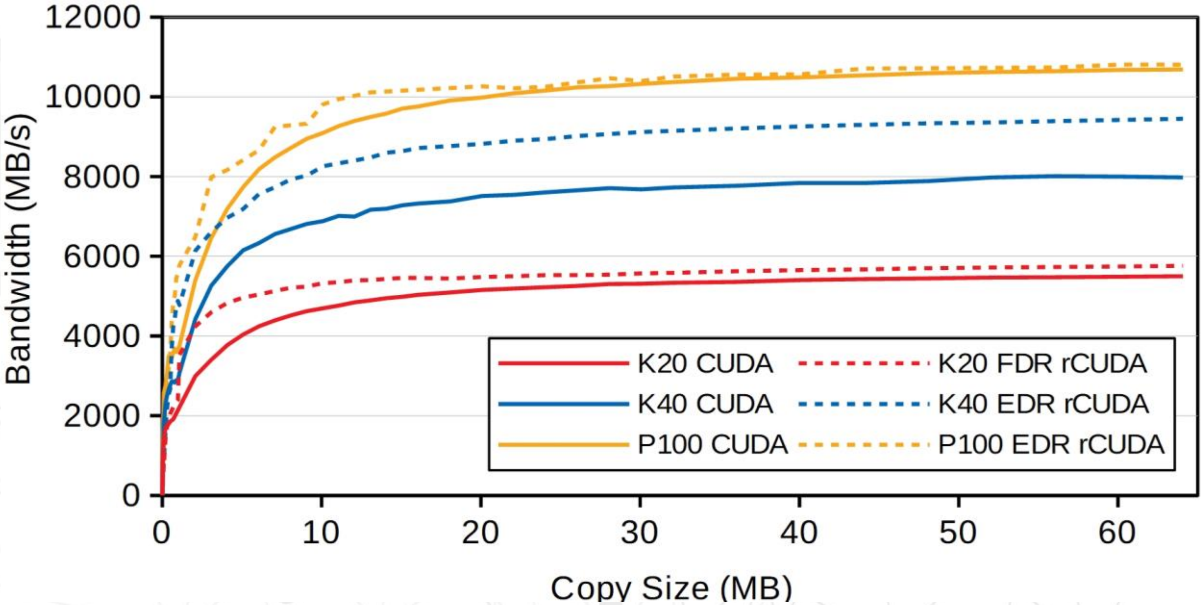


rCUDA scenario 1

rCUDA scenario 2



Performance of data movements among GPUs



Higher
is better

Benefits of rCUDA?

Benefits of rCUDA?

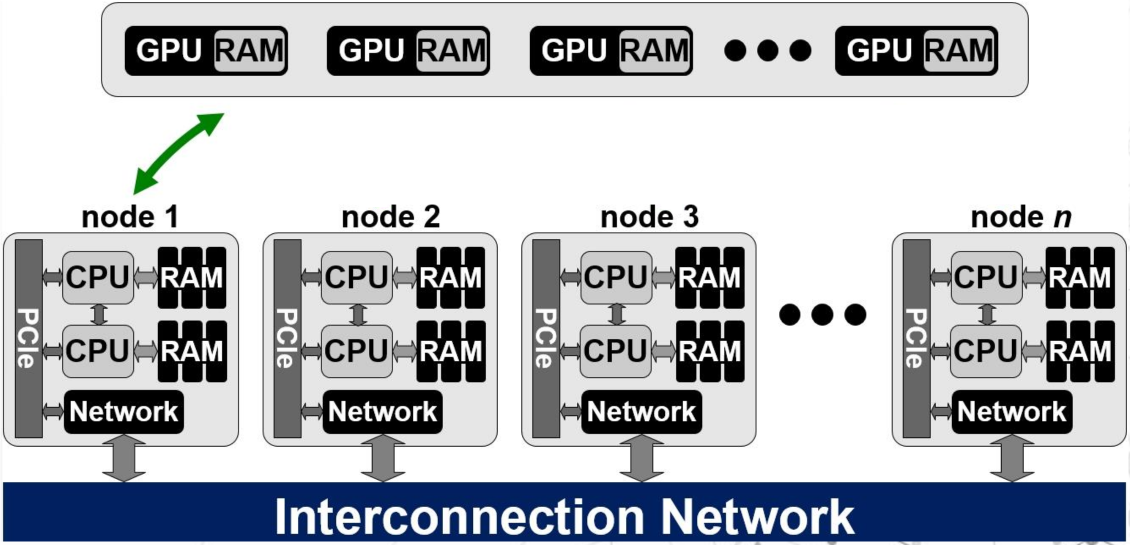
1. Many GPUs for an application
2. Increased cluster throughput

Outline

Benefits of rCUDA?

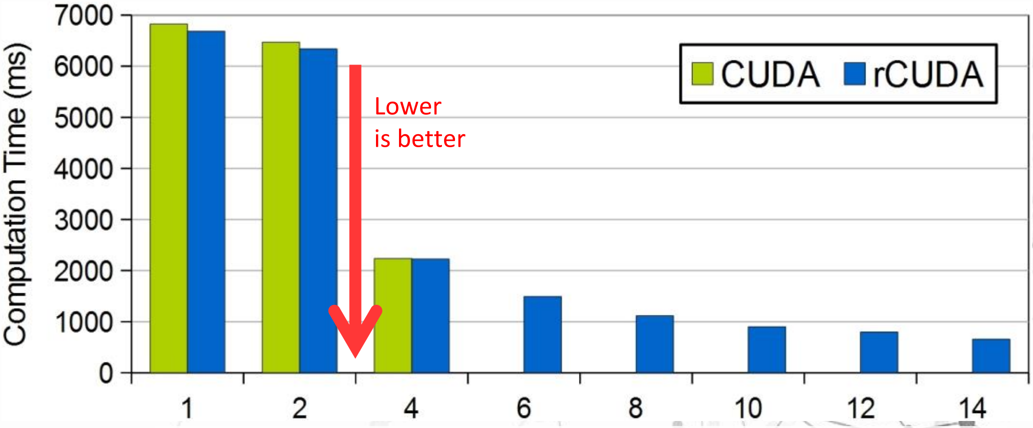
1. Many GPUs for an application
2. Increased cluster throughput

Providing many GPUs to an application with rCUDA



Providing many GPUs to an application with rCUDA

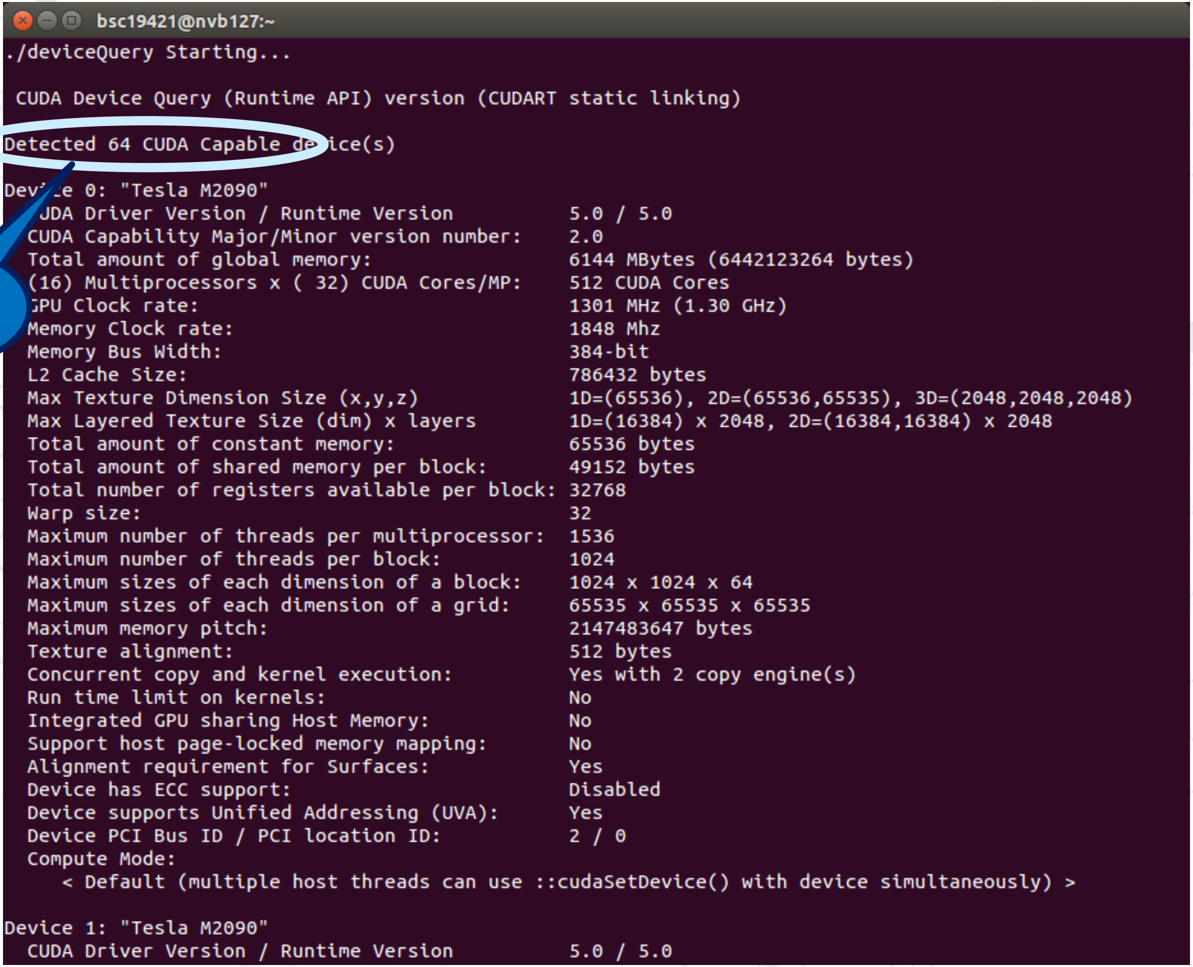
K20 GPUs and FDR InfiniBand



Lower is better

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
```

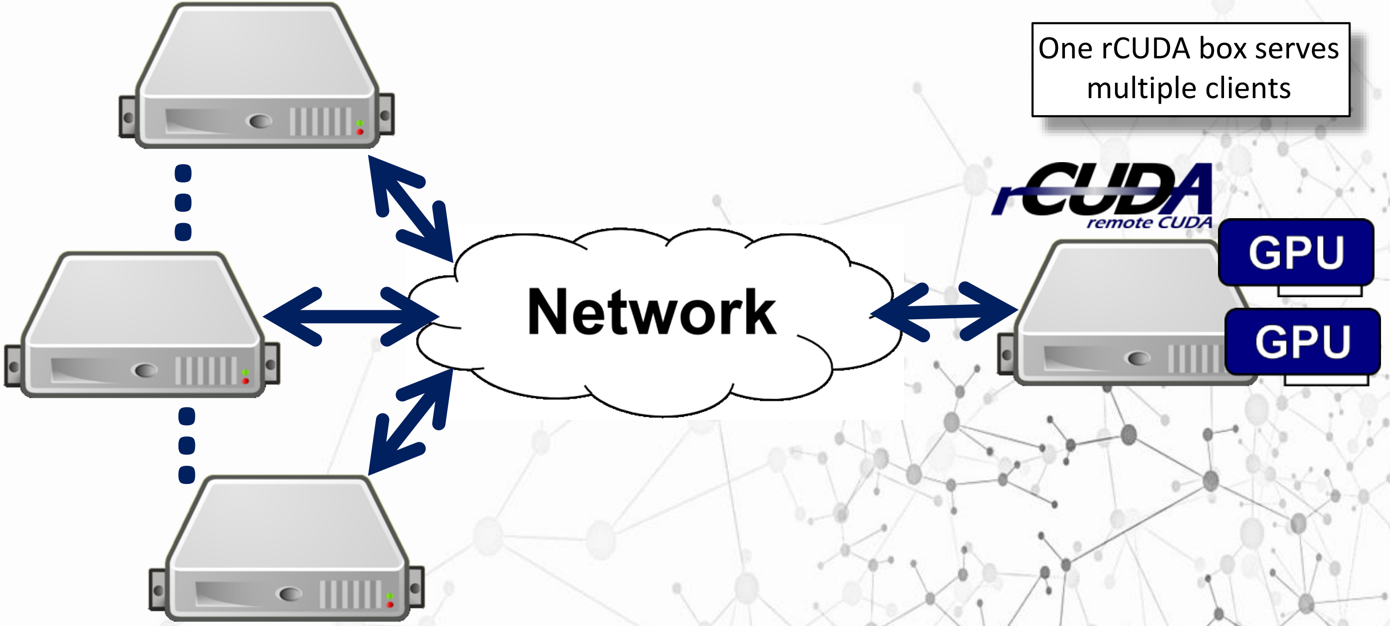
64
GPUs !!

Benefits of rCUDA?

1. Many GPUs for an application
2. Increased cluster throughput

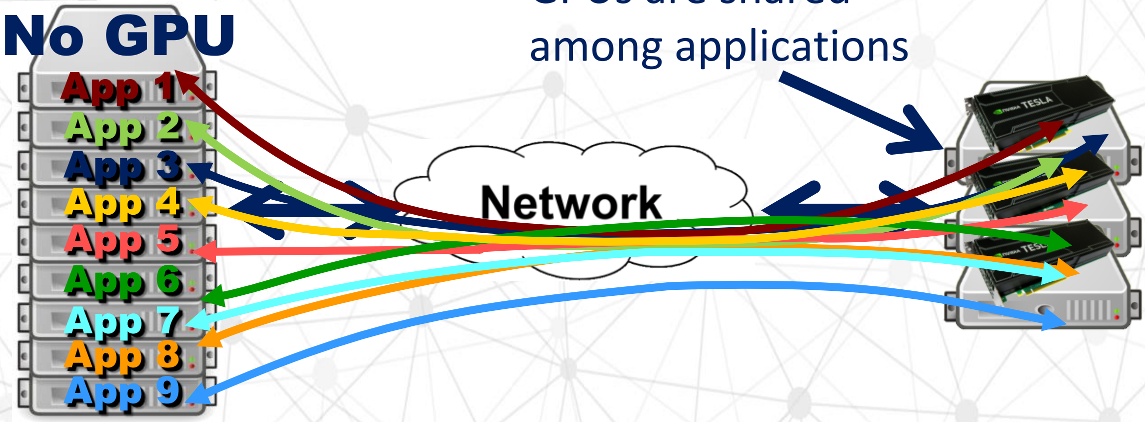
Increased cluster throughput

- **rCUDA servers can concurrently provide service to multiple clients**
  - Cluster throughput is increased as a consequence of serving multiple clients



One rCUDA box serves multiple clients

Need of a resource scheduler



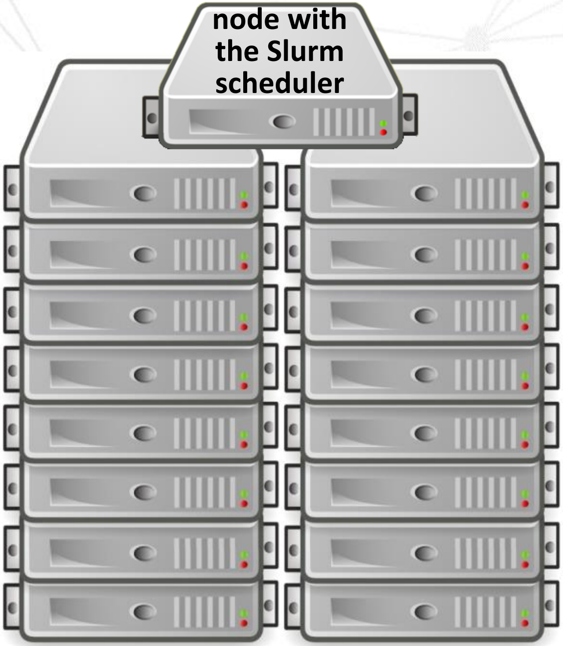
GPUs are shared among applications

Which is the limit of GPU sharing?

Cluster test bench

- Dual socket E5-2620v2 Intel Xeon + 32GB RAM + K20 GPU
- FDR InfiniBand based cluster

16 GPU nodes + 1 node



Workload used in the experiment

- Applications used in the tests:
  - GPU-Blast
  - LAMMPS
  - MCUDA-MEME
  - GROMACS
  - NAMD
  - BarraCUDA
  - GPU-LIBSVM
  - MUMmerGPU

Non-GPU

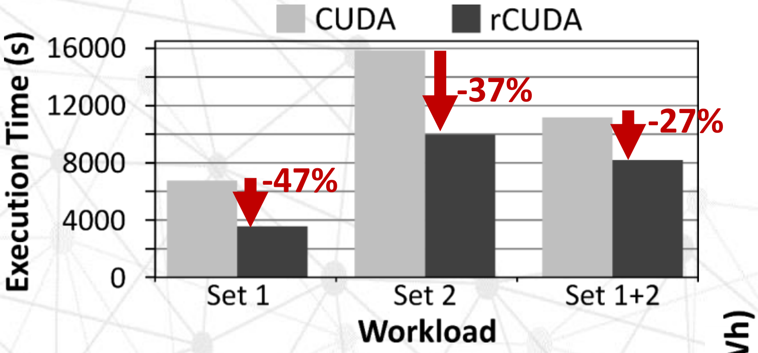
Set 1 Short execution time

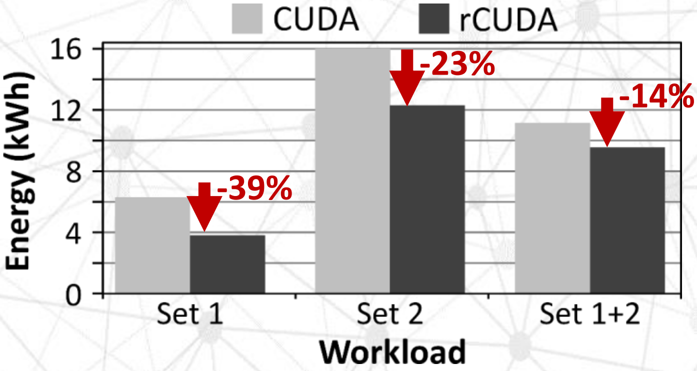
Set 2 Long execution time

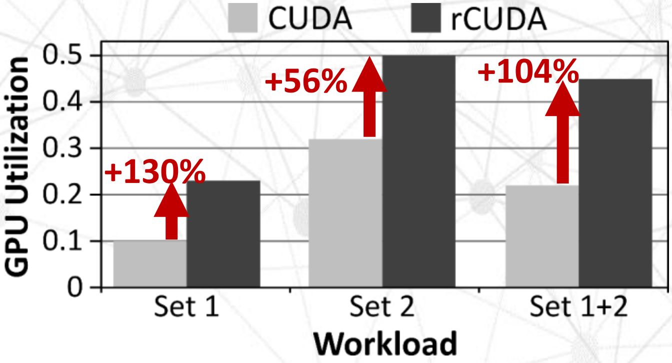
- Three workloads:
  - Set 1
  - Set 2
  - Set 1 + 2

400 jobs at each set

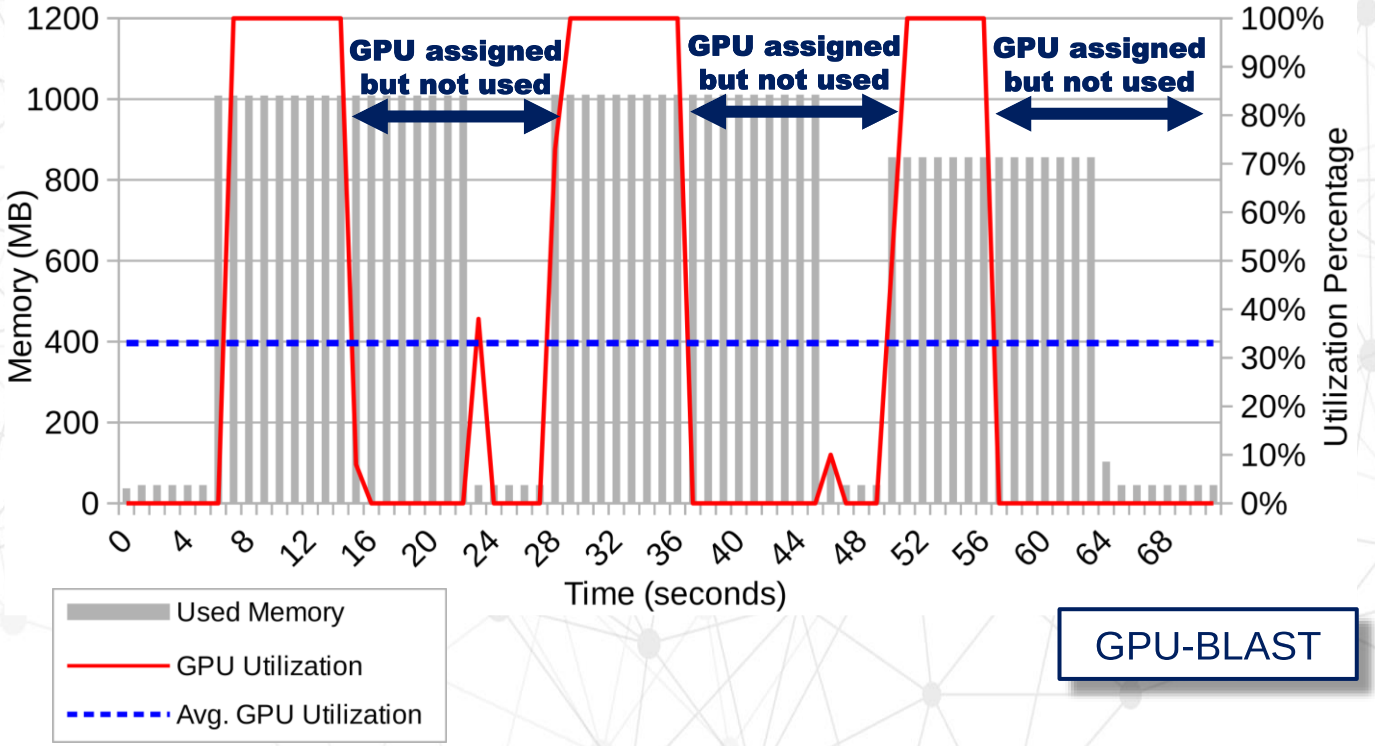
Cluster performance when using rCUDA







Increased cluster throughput



GPU assigned but not used

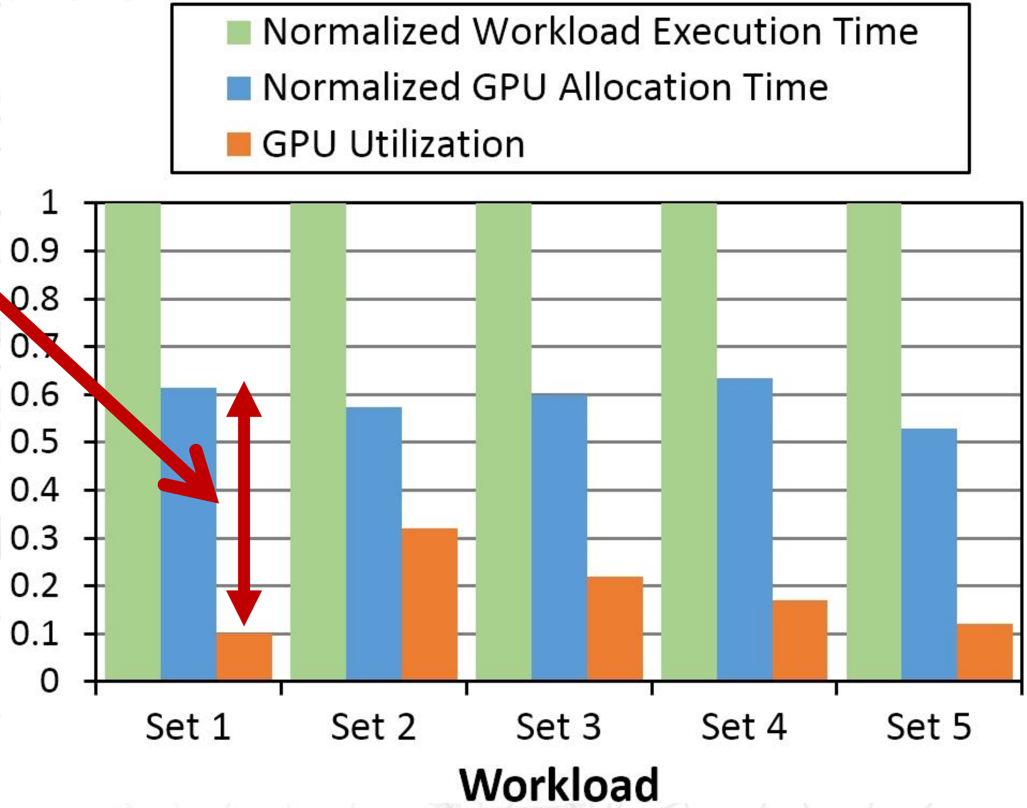
GPU assigned but not used

GPU assigned but not used

GPU-BLAST

GPU allocation vs GPU utilization

GPUs assigned but not used

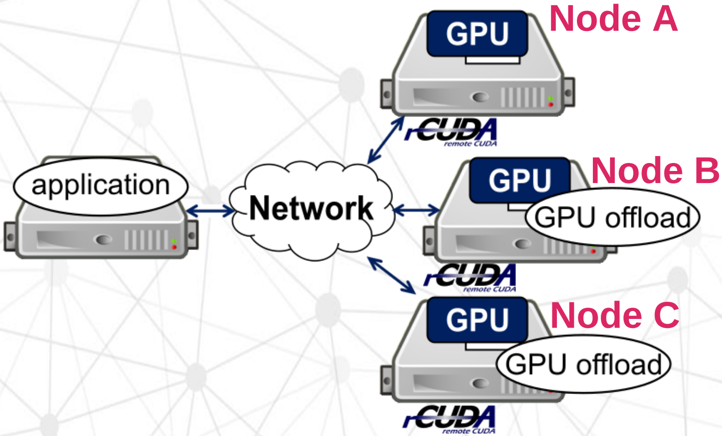


Outline

GPU-Job migration

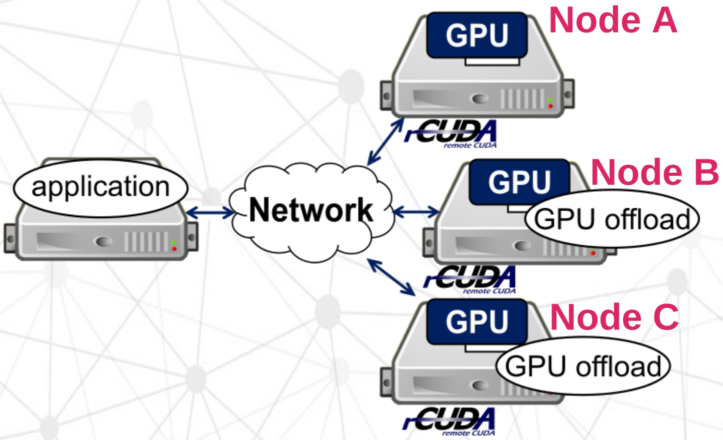
(increasing flexibility of using GPUs even more)

GPU-Job migration



1. The application is initially assigned GPUs in nodes B and C

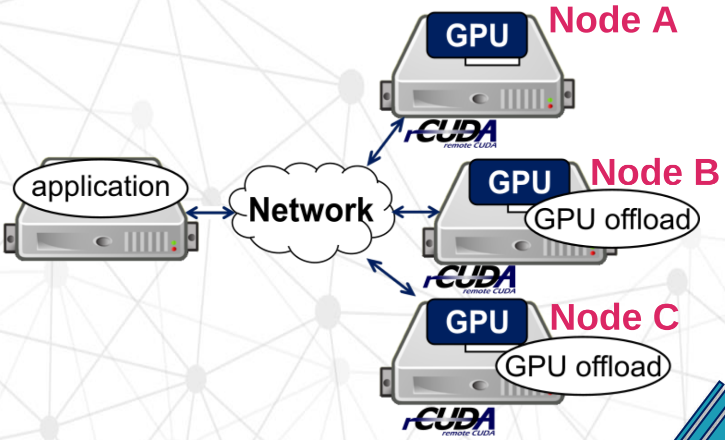
GPU-Job migration



1. The application is initially assigned GPUs in nodes B and C

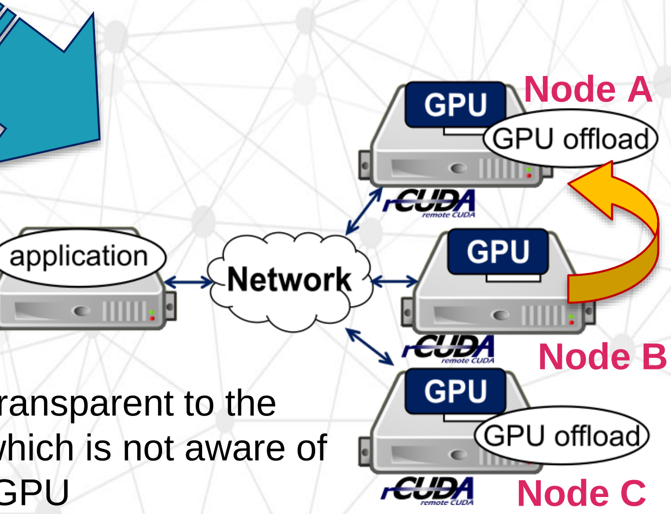
2. During the execution of the application, it is decided that **the GPU part** of the application in node B **must be moved** to node A

GPU-Job migration



1. The application is initially assigned GPUs in nodes B and C

2. During the execution of the application, it is decided that **the GPU part** of the application in node B **must be moved** to node A



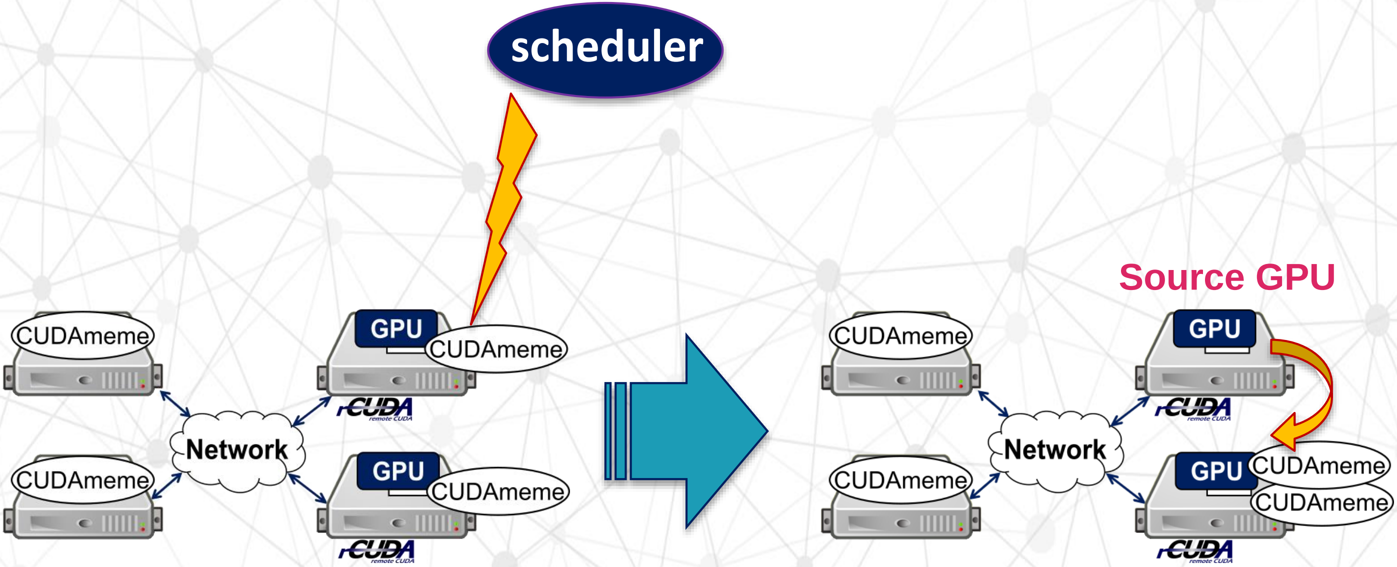
3. Migration is transparent to the application, which is not aware of using a new GPU

Outline

Use cases

- server consolidation
- load balancing

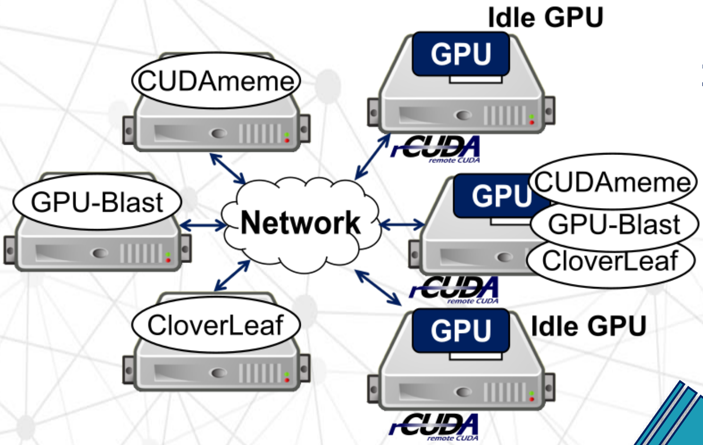
Consolidation process



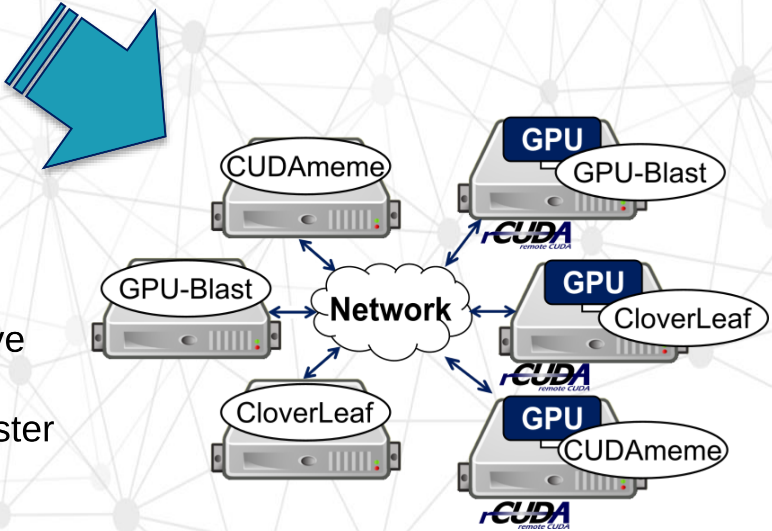
Source GPU

Target GPU

General idea for load balancing



1. At some point in time, one of the GPUs is heavily loaded while other GPUs remain idle

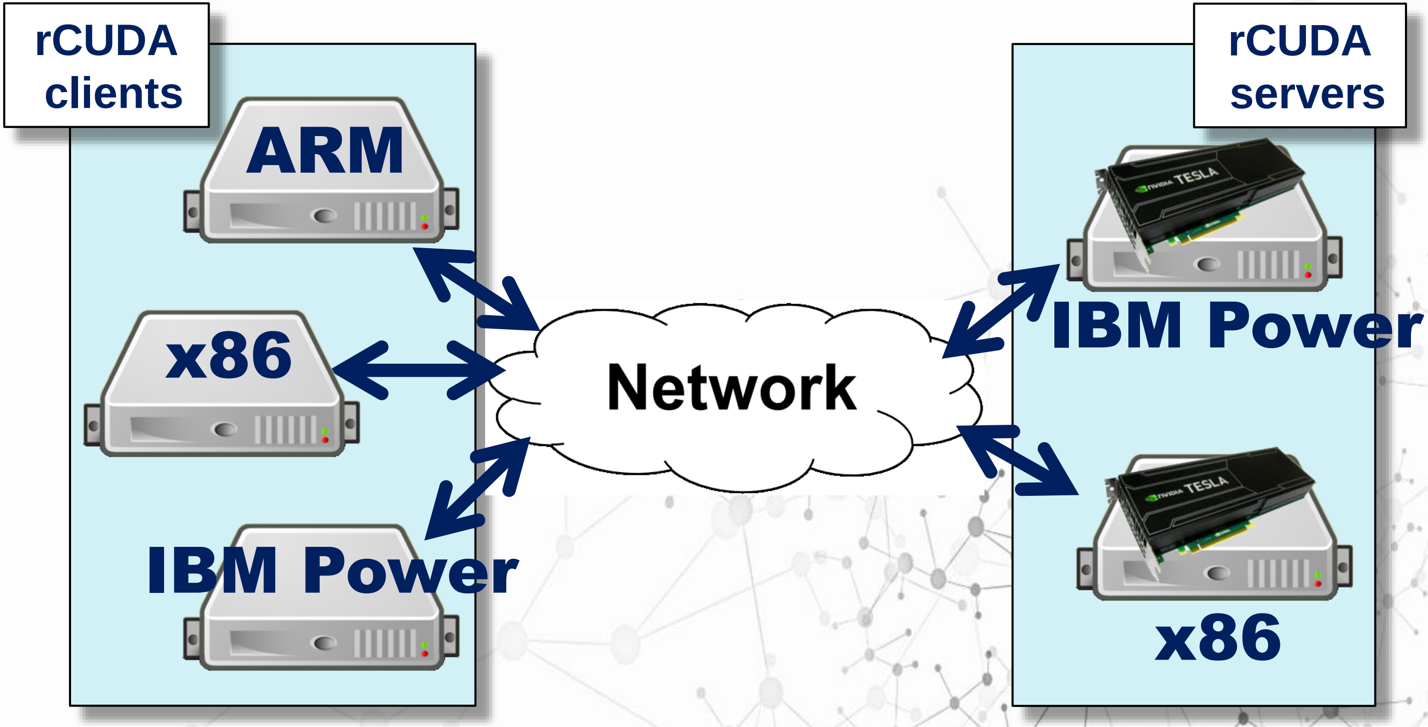


2. Migration can be used to move GPU-jobs to idle GPUs thus balancing load across the cluster

rCUDA increases heterogeneity in the cluster

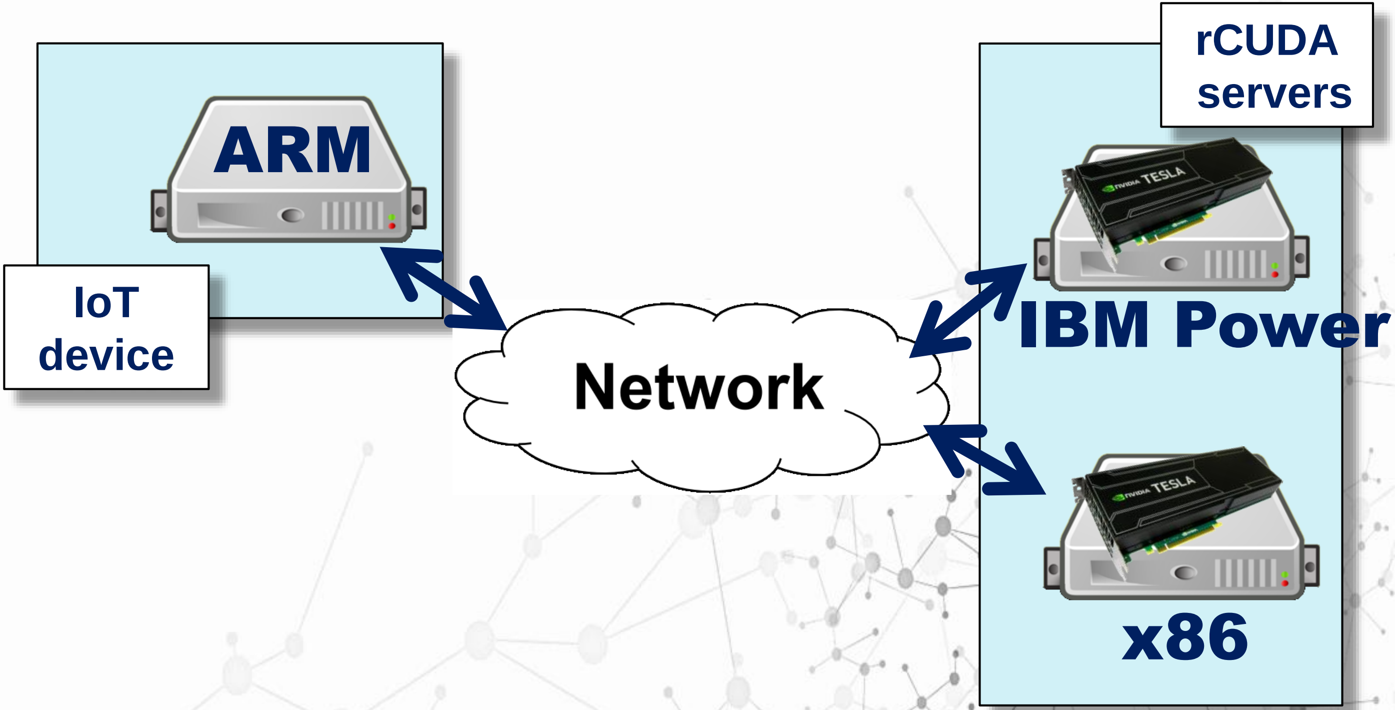
Heterogeneous clusters

rCUDA clients and servers can use different processor architectures



Heterogeneous clusters

rCUDA clients and servers can use different processor architectures



rCUDA

remote CUDA

Get a free copy of rCUDA at

http://www.rcuda.net

More than 1000 requests world wide

@rcuda_

rCUDA is a development by Universitat Politècnica de València, Spain