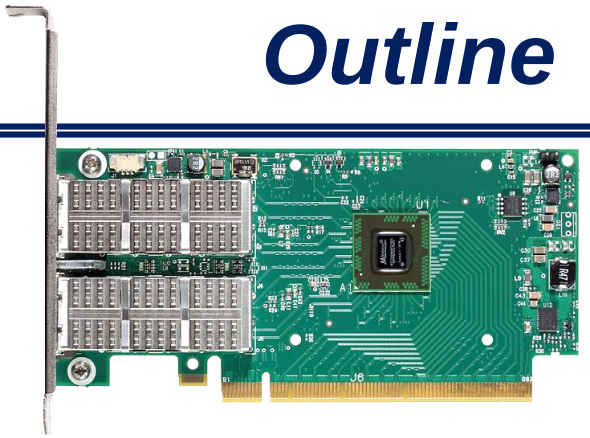
Integración de tecnologías de virtualización de GPUs en el planificador de recursos SLURM

Hands-on Session

Carlos Reaño
Queen's University Belfast, UK
http://go.qub.ac.uk/carlosreano

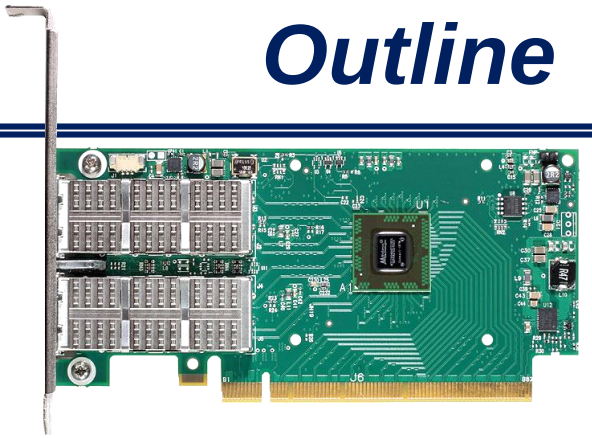
HPC ADMINTECH 2019 Conference
May 22-24, 2019, Valencia, Spain

Outline



- What is rCUDA?
- Installing and using rCUDA
- rCUDA over HPC networks
  - InfiniBand
- How taking benefit from rCUDA
  - Sample scenarios
- SLURM integration
- Questions & Answers

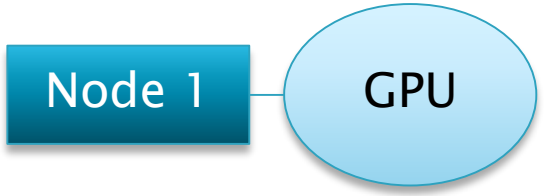
Outline



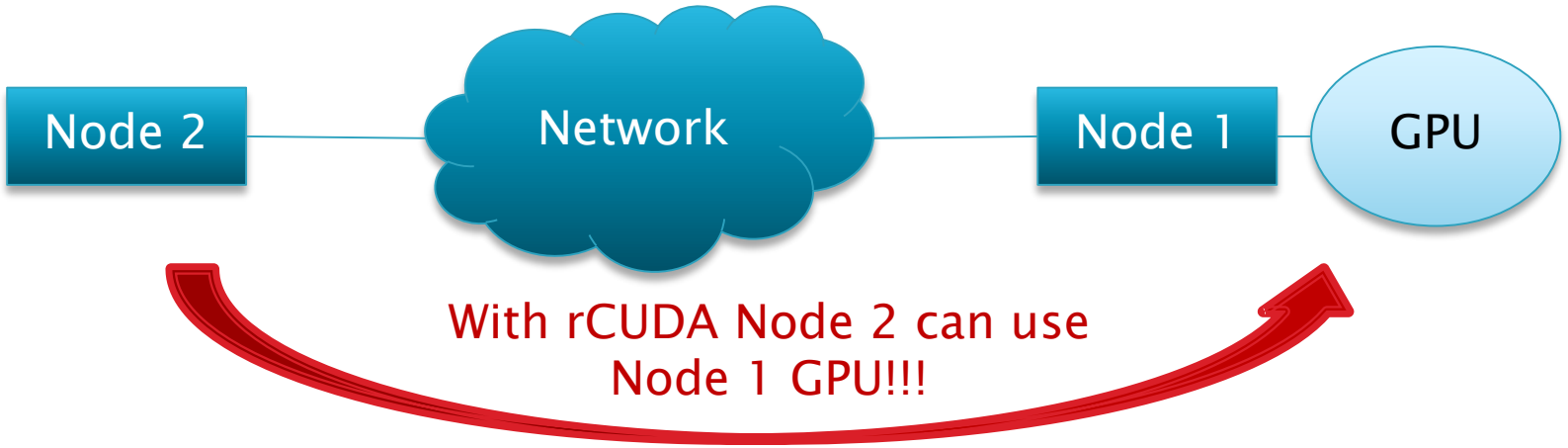
- **What is rCUDA?**
- Installing and using rCUDA
- rCUDA over HPC networks
  - InfiniBand
- How taking benefit from rCUDA
  - Sample scenarios
- SLURM integration
- Questions & Answers

What is rCUDA?

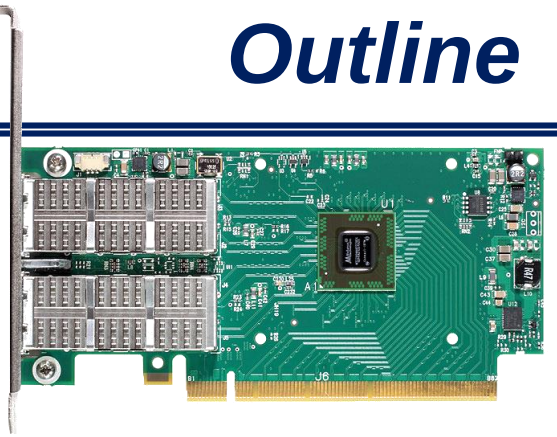
- CUDA:



- rCUDA (remote CUDA):



Outline



- What is rCUDA?
- **Installing and using rCUDA**
- rCUDA over HPC networks
  - InfiniBand
- How taking benefit from rCUDA
  - Sample scenarios
- SLURM integration
- Questions & Answers

Installing and using rCUDA

- Where obtain rCUDA?
  - www.rCUDA.net: Software Request Form
- Package contents. Important folders:
  -  doc: rCUDA user's guide & quick start guide
  -  bin: rCUDA server daemon
  -  lib: rCUDA library
- Installing rCUDA
  - Just untar the tarball in both the server(s) and the client(s) node(s)

Installing and using rCUDA

- Starting rCUDA server:
  - Set env. vars as if you were going to run a CUDA program:

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
```

  - Start rCUDA server:

```
cd $HOME/rCUDA/bin
./rCUDAd
```

Installing and using rCUDA

- Starting rCUDA server:
  - Set env. vars as if you were going to run a CUDA program:


```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
```

  - Start rCUDA server:

```
cd $HOME/rCUDA/bin
./rCUDAd
```

Installing and using rCUDA

- Starting rCUDA server:
  - Set env. vars as if you were going to run a CUDA program:

Path to CUDA libraries

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
```

  - Start rCUDA server:

```
cd $HOME/rCUDA/bin
./rCUDAd
```

Installing and using rCUDA

- Starting rCUDA server:
  - Set env. vars as if you were going to run a CUDA program:

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
```

  - Start rCUDA server:

Path to rCUDA server

```
cd $HOME/rCUDA/bin
./rCUDAd
```

Installing and using rCUDA

- Starting rCUDA server:
  - Set env. vars as if you were going to run a CUDA program:

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
```

  - Start rCUDA server:

```
cd $HOME/rCUDA/bin
./rCUDAd
```

Start rCUDA server in background

Installing and using rCUDA

- Running a CUDA program with rCUDA:
  - Set env. vars as follows:

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/lib:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=1
export RCUDA_DEVICE_0=<server_name_or_ip_address>:0
```

  - Compile CUDA program using dynamic libraries:

```
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/deviceQuery
make EXTRA_NVCCFLAGS=--cudart=shared
```

  - Run the CUDA program as usual:

```
./deviceQuery
...
```

Installing and using rCUDA

- Running a CUDA program with rCUDA:
  - Set env. vars as follows:

Path to CUDA binaries

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/lib:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=1
export RCUDA_DEVICE_0=<server_name_or_ip_address>:0
```

  - Compile CUDA program using dynamic libraries:

```
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/deviceQuery
make EXTRA_NVCCFLAGS=--cudart=shared
```

  - Run the CUDA program as usual:

```
./deviceQuery
...
```

Installing and using rCUDA

- Running a CUDA program with rCUDA:
  - Set env. vars as follows:

Path to rCUDA library

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/lib:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=1
export RCUDA_DEVICE_0=<server_name_or_ip_address>:0
```

  - Compile CUDA program using dynamic libraries:

```
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/deviceQuery
make EXTRA_NVCCFLAGS=--cudart=shared
```

  - Run the CUDA program as usual:

```
./deviceQuery
...
```

Installing and using rCUDA

- Running a CUDA program with rCUDA:
  - Set env. vars as follows:

Number of remote GPUs: 1, 2, 3...

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/lib:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=1
export RCUDA_DEVICE_0=<server_name_or_ip_address>:0
```

  - Compile CUDA program using dynamic libraries:

```
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/deviceQuery
make EXTRA_NVCCFLAGS=--cudart=shared
```

  - Run the CUDA program as usual:

```
./deviceQuery
...
```

Installing and using rCUDA

- Running a CUDA program with rCUDA:
  - Set env. vars as follows:

Name/IP of rCUDA server

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/li$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=1
export RCUDA_DEVICE_0=<server_name_or_ip_address>:0
```

  - Compile CUDA program using dynamic libraries:

```
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/deviceQuery
make EXTRA_NVCCFLAGS=--cudart=shared
```

  - Run the CUDA program as usual:

```
./deviceQuery
...
```

Installing and using rCUDA

- Running a CUDA program with rCUDA:
  - Set env. vars as follows:

GPU of remote server to use

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/lib:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=1
export RCUDA_DEVICE_0=<server_name_or_ip_address>:0
```

  - Compile CUDA program using dynamic libraries:

```
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/deviceQuery
make EXTRA_NVCCFLAGS=--cudart=shared
```

  - Run the CUDA program as usual:

```
./deviceQuery
...
```

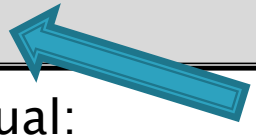
Installing and using rCUDA

- Running a CUDA program with rCUDA:
  - Set env. vars as follows:

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/lib:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=1
export RCUDA_DEVICE_0=<server_name_or_ip_address>:0
```

  - Compile CUDA program using dynamic libraries:

```
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/deviceQuery
make EXTRA_NVCCFLAGS=--cudart=shared
```



Very important!!!

  - Run the CUDA program as usual:

```
./deviceQuery
...
```

Installing and using rCUDA

- Running a CUDA program with rCUDA:
  - Set env. vars as follows:

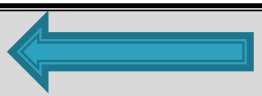
```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/lib:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=1
export RCUDA_DEVICE_0=<server_name_or_ip_address>:0
```

  - Compile CUDA program using dynamic libraries:

```
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/deviceQuery
make EXTRA_NVCCFLAGS=--cudart=shared
```

  - Run the CUDA program as usual:

```
./deviceQuery
...
```



Installing and using rCUDA

- Live demonstration:
  - deviceQuery
  - bandwidthTest

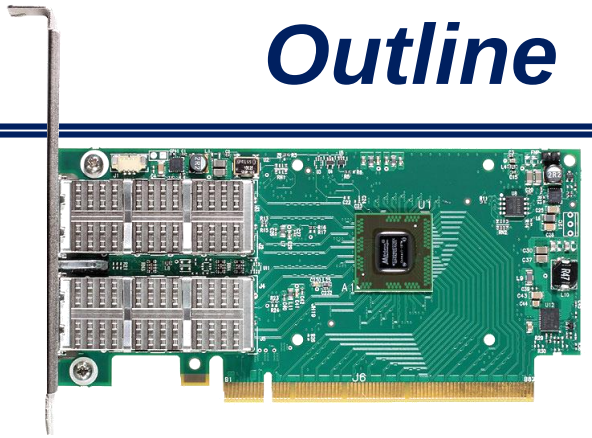
Installing and using rCUDA

- Live demonstration:
  - deviceQuery
  - bandwidthTest
- Problem: bandwidth with rCUDA is too low!!
  - Why? We are using TCP

Installing and using rCUDA

- Live demonstration:
  - deviceQuery
  - bandwidthTest
- Problem: bandwidth with rCUDA is too low!!
  - Why? We are using TCP
- Solution: HPC networks
  - InfiniBand (IB)

Outline



- What is rCUDA?
- Installing and using rCUDA
- **rCUDA over HPC networks**
  - **InfiniBand**
- How taking benefit from rCUDA
  - Sample scenarios
- SLURM integration
- Questions & Answers

rCUDA over HPC networks: InfiniBand

- Starting rCUDA server using IB:

```
export RCUDA_NETWORK=IB
cd $HOME/rCUDA/bin
./rCUDAd
```

- Run CUDA program using rCUDA over IB:

```
export RCUDA_NETWORK=IB
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/bandwidthTest
./bandwidthTest
```

rCUDA over HPC networks: InfiniBand

- Starting rCUDA server using IB:

Tell rCUDA we want to use IB

```
export RCUDA_NETWORK=IB
cd $HOME/rCUDA/bin
./rCUDAd
```

- Run CUDA program using rCUDA over IB:

Also in the client!!

```
export RCUDA_NETWORK=IB
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/bandwidthTest
./bandwidthTest
```

rCUDA over HPC networks: InfiniBand

- Starting rCUDA server using IB:

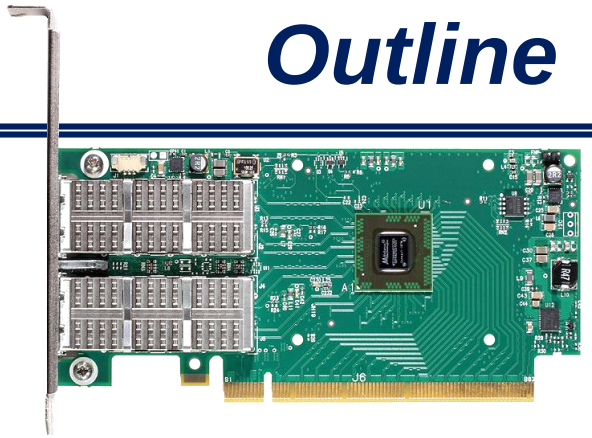
```
export RCUDA_NETWORK=IB
cd $HOME/rCUDA/bin
./rCUDAd
```

- Run CUDA program using rCUDA over IB:

```
export RCUDA_NETWORK=IB
cd $HOME/NVIDIA_CUDA_Samples/1_Utilities/bandwidthTest
./bandwidthTest
```

- Live demonstration:
  - bandwidthTest using IB
  - Bandwidth is no more a problem!!

Outline



- What is rCUDA?
- Installing and using rCUDA
- rCUDA over HPC networks
  - InfiniBand
- **How taking benefit from rCUDA**
  - **Sample scenarios**
- SLURM integration
- Questions & Answers

How taking benefit from rCUDA

- Sample scenarios:
  - Typical behavior of CUDA applications: moving data to the GPU and performing a lot of computations there to compensate the overhead of having moved the data
    - This benefits rCUDA: more computations, less rCUDA overhead
  - Scalable applications: more GPUs, less execution time
    - rCUDA can use all the GPUs of the cluster, while CUDA only can use the ones directly connected to one node: for some applications, rCUDA can get better results than with CUDA
  - Heterogeneous clusters: access to GPU servers from ATOM, ARM…
    - rCUDA can be used to access GPU servers in x86 or Power8 machines, from different systems and architectures (ATOM, ARM, Intel-D…)

How taking benefit from rCUDA

- Three main types of applications:
  - Bandwidth bounded: more transfers, more rCUDA overhead
  - Computations bounded: more computations, less rCUDA overhead
  - Intermediate

Tensorflow

- GPU vs. remote GPU
  - Overhead of remote GPUs?
- Live demonstration:
  - Tensorflow with CUDA
  - Tensorflow with rCUDA

Tensorflow

- CPU vs. remote GPU
  - What is better: a local CPU or a remote GPU?
- Live demonstration:
  - Tensorflow on CPU (without CUDA)

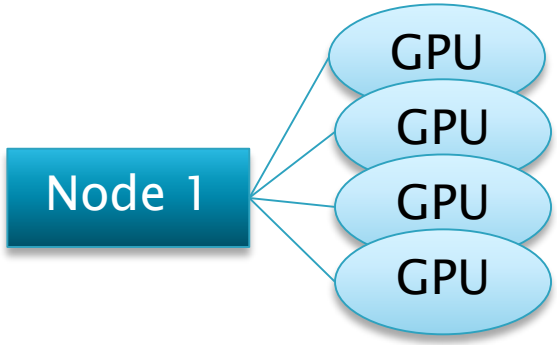
How taking benefit from rCUDA

- Sample scenarios:
  - Typical behavior of CUDA applications: moving data to the GPU and performing a lot of computations there to compensate the overhead of having moved the data
    - This benefits rCUDA: more computations, less rCUDA overhead
  - Scalable applications: more GPUs, less execution time
    - rCUDA can use all the GPUs of the cluster, while CUDA only can use the ones directly connected to one node: for some applications, rCUDA can get better results than with CUDA
  - Heterogeneous clusters: access to GPU servers from ATOM, ARM…
    - rCUDA can be used to access GPU servers in x86 or Power8 machines, from different systems and architectures (ATOM, ARM, Intel-D…)

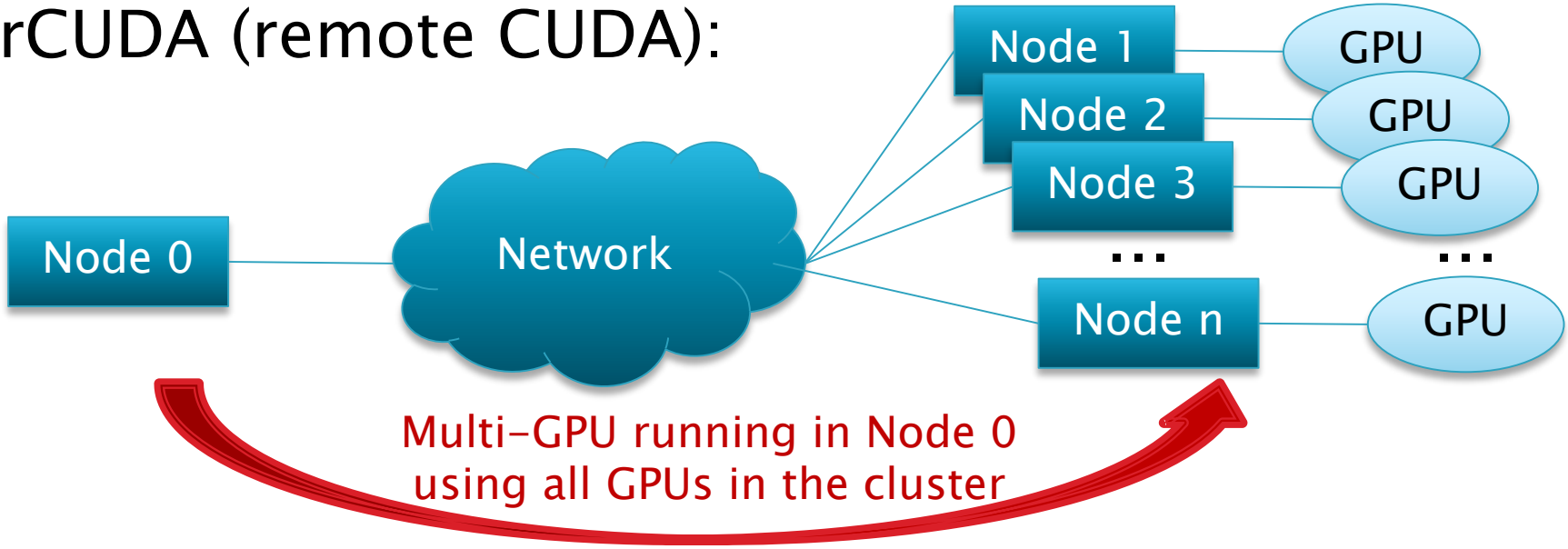
Multi-GPU scenario

- CUDA:

Multi-GPU App running in Node 1 using all the GPUs in the node



- rCUDA (remote CUDA):



Multi-GPU Configuration

- Configure rCUDA for Multi-GPU:

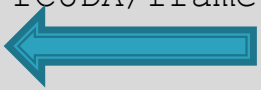
```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/framework/rCUDAl:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=5
export RCUDA_DEVICE_0=node1:0
export RCUDA_DEVICE_1=node1:1
export RCUDA_DEVICE_2=node2:0
export RCUDA_DEVICE_3=node3:0
export RCUDA_DEVICE_4=node4:0
```

  - Check configuration by running deviceQuery sample

Multi-GPU Configuration

- Configure rCUDA for Multi-GPU:

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/framework/rCUDAl:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=5
export RCUDA_DEVICE_0=node1:0
export RCUDA_DEVICE_1=node1:1
export RCUDA_DEVICE_2=node2:0
export RCUDA_DEVICE_3=node3:0
export RCUDA_DEVICE_4=node4:0
```



Number of remote GPUs

  - Check configuration by running deviceQuery sample

Multi-GPU Configuration

- Configure rCUDA for Multi-GPU:

```
export PATH=$PATH:/usr/local/cuda/bin
export LD_LIBRARY_PATH=$HOME/rCUDA/framework/rCUDAl:$LD_LIBRARY_PATH
export RCUDA_DEVICE_COUNT=5
export RCUDA_DEVICE_0=node1:0
export RCUDA_DEVICE_1=node1:1
export RCUDA_DEVICE_2=node2:0
export RCUDA_DEVICE_3=node3:0
export RCUDA_DEVICE_4=node4:0
```

Location of each GPU

  - Check configuration by running deviceQuery sample

Multi-GPU Tensorflow

- Live demonstration:
  - deviceQuery sample with multiple GPUs
  - Multi-GPU Tensorflow

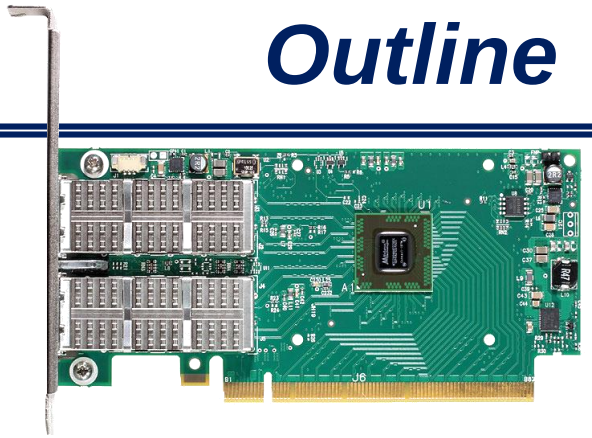
How taking benefit from rCUDA

- Sample scenarios:
  - Typical behavior of CUDA applications: moving data to the GPU and performing a lot of computations there to compensate the overhead of having moved the data
    - This benefits rCUDA: more computations, less rCUDA overhead
  - Scalable applications: more GPUs, less execution time
    - rCUDA can use all the GPUs of the cluster, while CUDA only can use the ones directly connected to one node: for some applications, rCUDA can get better results than with CUDA
  - Heterogeneous clusters: access to GPU servers from ATOM, ARM…
    - rCUDA can be used to access GPU servers in x86 or Power8 machines, from different systems and architectures (ATOM, ARM, Intel-D…)

How taking benefit from rCUDA

- Heterogeneous clusters:
  - Access from low power nodes (Atom, ARM , Intel D…) to x86 GPU accelerated nodes
  - Access from no-Power8 nodes to Power8 GPU accelerated nodes

Outline



- What is rCUDA?
- Installing and using rCUDA
- rCUDA over HPC networks
  - InfiniBand
- How taking benefit from rCUDA
  - Sample scenarios
- **SLURM integration**
- Questions & Answers

SLURM integration

- Example [1]:

```
$> srun -rcuda-mode=shar -gres=rgpu:4:100M job.sh
```

Modes:
- "shar": GPU shared with other jobs
- "excl": GPU exclusive for job

[1] "Improving the management efficiency of GPU workloads in data centers through GPU virtualization". S. Iserte, J. Prades, C. Reaño and F. Silla. Accepted for publication in CCPE journal. https://doi.org/10.1002/cpe.5275

SLURM integration

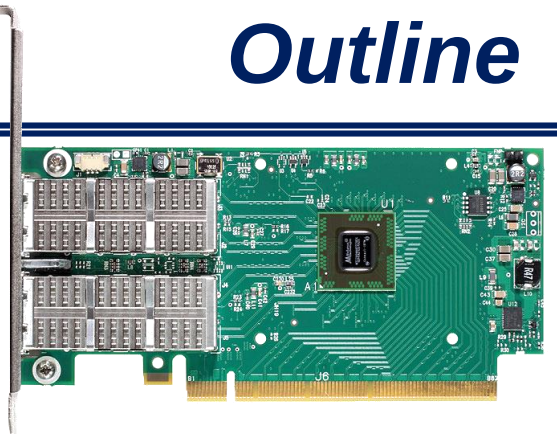
- Example [1]:

```
$> srun -rcuda-mode=shar -gres=rgpu:4:100M job.sh
```

GRES (Generic resource):
- "rgpu": remote/virtual GPU
- "4": requested number of virtual GPUs
- "100M": requested GPU memory for each virtual GPU

[1] "Improving the management efficiency of GPU workloads in data centers through GPU virtualization". S. Iserte, J. Prades, C. Reaño and F. Silla. Accepted for publication in CCPE journal. https://doi.org/10.1002/cpe.5275

Outline



- What is rCUDA?
- Installing and using rCUDA
- rCUDA over HPC networks
  - InfiniBand
- How taking benefit from rCUDA
  - Sample scenarios
- SLURM integration
- **Questions & Answers**

Get a free copy of rCUDA at

http://www.rcuda.net

 **@rcuda_**