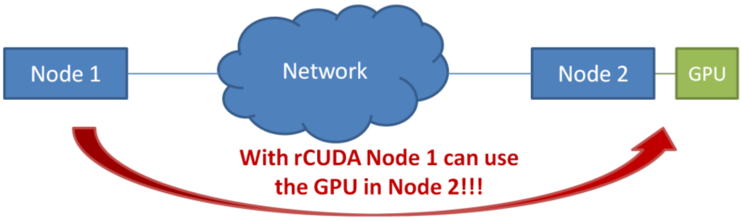
rCUDA remote CUDA

rCUDA Frequently Asked Questions



What is rCUDA?

rCUDA (remote CUDA) is a middleware that allows CUDA applications to use GPUs located in remote nodes **without any modifications** to original CUDA applications.

Which are the main rCUDA features?

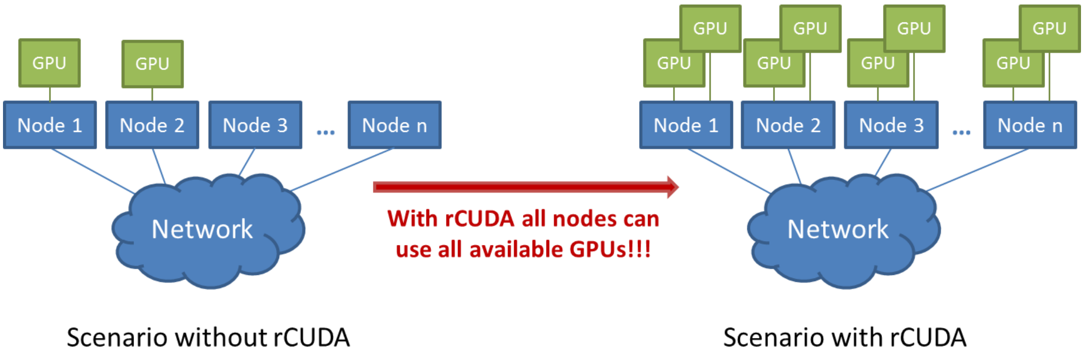
- Support for Linux OS
- Support for CUDA 8.0 and the associated CUDA libraries
- Enhanced communications over InfiniBand and RoCE networks
- Graphics interoperability is not supported (OpenGL, Direct3D, ...)
- 64-bit ARM and Powe8 rCUDA versions available (can interact with 64-bit x86 architectures)

How can I start using rCUDA?

Just visit **www.rCUDA.net** and complete the software request form.

How many remote GPUs could my application use?

Your application will be able to use **all the GPUs available in the cluster**, as far as it is appropriately programmed to leverage more than one GPU. Briefly, if your regular CUDA application is able to make use of as many GPUs as available in the box, then with rCUDA it will be able to use all the GPUs in the cluster.



Which network fabrics are supported?

rCUDA has a modular, layered architecture which supports runtime-loadable, network-specific communication libraries. rCUDA currently provides communication modules for **Ethernet (TCP/IP), InfiniBand RDMA and RoCE RDMA.**

Do I have to modify my application code in order to execute it with rCUDA?

No code modification is required in order to remotely accelerate your application with rCUDA. rCUDA is based on the use of dynamic libraries that replace the CUDA ones and, therefore, your application code will be appropriately linked to the rCUDA code when started. Furthermore, the rCUDA library will automatically locate the rCUDA servers by using environment variables.

Is it possible to access a remote GPU installed in an x86 box from an ARM box?

Yes, it is possible. You can accelerate your ARM-based system by attaching to it a GPU box based on a regular x86 processor. It is also possible to mix the x86 and Power8 architectures. You can send your GPU kernels from an application being executed in an x86 box to a GPU installed in a Power8 system. The contrary is also possible.

What if I have more questions?

More information available at www.rCUDA.net or at info@rCUDA.net