Benefits of remote GPU virtualization: the rCUDA perspective

1st Briefly, what is "remote GPU virtualization"?

CUDA application
CUDA libraries

With rCUDA (remote CUDA) applications are not limited to local GPUs but they can use any GPU in the cluster. This is known as remote GPU virtualization.

2nd Tell me more about rCUDA!

Remote GPU virtualization frameworks follow a client-server distributed approach. The client part of the middleware is installed in the cluster node executing the application requesting GPU services, whereas the server side runs in the computer owning the actual GPU. In this manner, the client receives a CUDA request from the accelerated application and appropriately processes and forwards it to the remote server. In the server node, the request is received and interpreted. Then it is forwarded to the GPU, which completes the execution of the request and provides the associated results to the server process. In turn, the server sends back the results to the client. In order to use rCUDA, applications do not need to be modified. They only must be linked against the rCUDA libraries instead of the CUDA libraries.

3rd Is rCUDA a slower version of CUDA?

rCUDA works on top of CUDA. However, with rCUDA the GPU is much farther away because the GPU is not at the other end of the PCIe link but it is located at other node of the cluster, having a network fabric in between. This longer distance introduces some overhead. How large is that overhead? The figure shows that it is lower than 4% for most applications.

4th Main rCUDA characteristics

rCUDA is binary compatible with CUDA 9.1 and implements the CUDA Runtime and Driver APIs. It supports the libraries included within CUDA (cuBLAS, cuDNN, etc). rCUDA supports several underlying network technologies (currently TCP/IP and InfiniBand). Data exchange between rCUDA clients and servers is pipelined in order to attain high performance. Regarding the InfiniBand communication module, it is based on the InfiniBand Verbs API in order to achieve high performance. GPUDirect RDMA is also supported.

The rCUDA middleware is available for the x86-64, 64-bit ARM and OpenPower processor architectures. Furthermore, rCUDA clients and rCUDA servers for different processor architectures can interact among them. For instance, it is possible to run the application in an x86-64 box while using a remote GPU in an OpenPower system.

You can obtain a free copy of rCUDA at www.rcuda.net

rCUDA is a development by Technical University of Valencia
**5th Which are the main benefits of using remote GPU virtualization?**

The testbed used in the experiments below consists of several 1027GR-TRF Supermicro nodes featuring two Intel Xeon E5-2620v2 processors. They also include an FDR InfiniBand adapter and one NVIDIA Tesla K20 GPU. Some tests use a server containing 4 NVIDIA Tesla K20 GPUs and one FDR InfiniBand adapter.

**Benefit #1: more GPUs for a single application**

When rCUDA is used, a non-MPI application programmed to leverage several GPUs can use all the GPUs in the cluster from a single node. 64 GPUs were assigned with rCUDA to a single non-MPI application (the deviceQuery sample by NVIDIA).

**Benefit #2: increased cluster performance with Slurm**

The Slurm workload manager can be modified in order to leverage the virtual GPUs provided by rCUDA. In this way, Slurm schedules the concurrent usage of GPUs among applications, that transparently share them. In order to assess the performance of a cluster using BarraCUDA, and MUMmer. Notice that the only difference between the green and blue bars is the use of the rCUDA middleware, which is distributed at no cost by Technical University of Valencia.

**Benefit #3: GPU migration**

Migrating jobs reduces overall energy consumption by making use of the consolidation concept. Unfortunately, migrating applications that use GPUs is a complex task. On the contrary, when using rCUDA it is very simple to migrate GPU-accelerated applications because this middleware intercepts all of the CUDA calls performed by the application and thus the consumption of resources in the GPU can be easily tracked.

**Benefit #4: virtual machines can easily access GPUs**

Applications running inside virtual machines (VMs) can only make use of high-performance GPUs (such as K20) if the GPUs are assigned to the VM by using the PCI-passthrough technique. Unfortunately, this technique assigns GPUs to VMs in an exclusive way.

rCUDA makes it possible to simultaneously assign a GPU to several VMs. Two possible scenarios: 1) VMs access the GPUs located in the host; 2) InfiniBand is available in the cluster and thus applications can access GPUs located in other nodes of the cluster. The plot shows the overhead with respect to using CUDA in a native domain.

**Benefit #5: cheaper cluster upgrade**

Thanks to the remote GPU virtualization technique, clusters not including GPUs can be updated for using GPUs by attaching to them one or more computers containing GPUs. As can be seen in the plots below, when a box with 4 GPUs is attached to a cluster with 16 non-GPU enabled nodes, the overall performance with rCUDA is much higher than with CUDA because all the nodes in the original cluster can make concurrent use of the GPUs in the new box. On the contrary, when CUDA is used, applications must wait a lot of time until they get access to the GPUs in the new box.