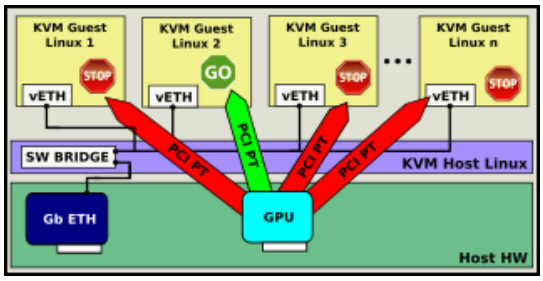
Deploying rCUDA in cloud computing environments

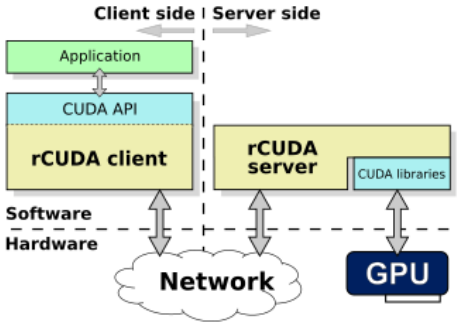
The use of GPUs in cloud computing environments is increasingly becoming mainstream. CUDA applications being executed inside a virtual machine (VM) can be accelerated by using the GPUs in the host computer. However, assigning the GPUs in the host to the several VMs being concurrently run in that server presents an important burden: efficiently sharing the GPU among the VMs. In this regard, although NVIDIA GRID GPUs address many of the issues related to cloud computing, these GPUs cannot be concurrently shared among several VMs for CUDA acceleration.

How can a virtual machine access CUDA GPUs?

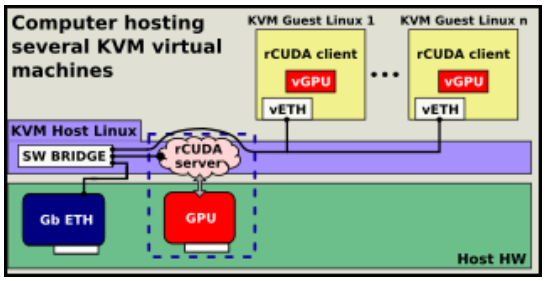
A virtual machine can access the GPUs in the host by means of the **PCI passthrough technique**. This mechanism assigns a PCI device to one of the VMs being executed in the server. However, this **assignment is made in an exclusive way**. That is, once a GPU is assigned to a VM, it cannot be assigned to other VMs until the device is detached from the former VM. In this regard, the PCI passthrough mechanism prevents the concurrent usage of GPUs among VMs.

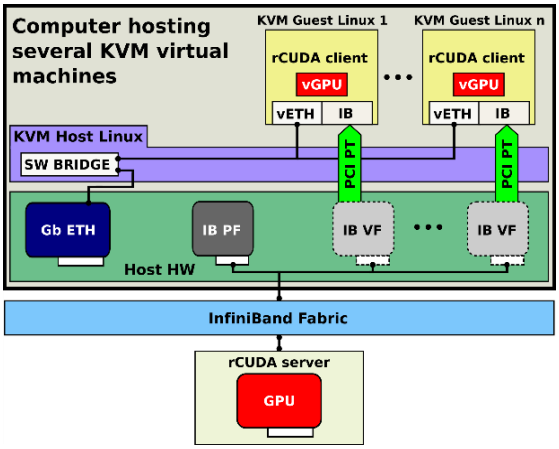


How does rCUDA avoid the limitations of PCI passthrough?



rCUDA is designed following a client-server approach. The client side of rCUDA is a library that replaces the NVIDIA CUDA library. The rCUDA client is placed in the same computer where the application is executed. On the other hand, the rCUDA server is executed in the computer owning the GPU. rCUDA client and server communicate through the network. Additionally, the rCUDA server allows several clients to concurrently share a GPU. In a cloud computing scenario, rCUDA clients are placed inside the VMs whereas the rCUDA server is placed in the native domain owning the GPU. The PCI passthrough is not required. In this way, GPUs can be concurrently shared by all the VMs. Two configurations are possible. In the first one (figure on the left) the GPU is located in the same host executing the VMs. In this case the virtual network provided by the hypervisor is used to communicate the rCUDA client and server. In the second scenario (figure on the right), an InfiniBand network fabric is available in the cluster. In this case, the rCUDA server can be placed in another computer in the cluster. Notice that a RoCE fabric could also be used.





Which performance can be expected?

When using rCUDA, a network is placed between the application and the GPU. Therefore, we can expect some impact on performance. Furthermore, this impact will greatly depend on the characteristics of the network. The figure on the right shows the overhead of using the above configurations with respect to executing the applications using CUDA in a native domain (no VM involved). It can be seen that, in average, performance loss is lower than 2%. The figure also shows that using a remote GPU across an InfiniBand network provides better performance than using a local GPU. The InfiniBand fabric used in the figure was an FDR network. With better InfiniBand fabrics, such as EDR InfiniBand, performance loss is expected to decrease.