Benefits of remote GPU virtualization: the rCUDA perspective

Federico Silla
Technical University of Valencia
Spain
What is “remote GPU virtualization”?
Basics of GPU computing

Basic behavior of CUDA

Application
CUDA libraries
GPU
GPU
GPU
GPU
Basics of GPU computing
Remote GPU virtualization

No GPU

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

rCUDA is a development by Technical University of Valencia.
Basics of remote GPU virtualization

Client side | Server side

Application

CUDA Runtime API

client engine

server engine

CUDA libraries

Software

Hardware

Network

GPU
Remote GPU virtualization allows a new vision of a GPU deployment, moving from the usual cluster configuration:

Remote GPU virtualization envision

Interconnection Network

Logical connections

Interconnection Network

Logical configuration
rCUDA impact on applications

- Several applications executed with CUDA and rCUDA
  - K20 GPU and FDR InfiniBand
  - K40 GPU and EDR InfiniBand

Lower is better
Benefits of “remote GPU virtualization”?
Benefit #1: more GPUs for a single application
1: more GPUs for a single application

- GPU virtualization is useful for multi-GPU applications

Without GPU virtualization:
- Only the GPUs in the node can be provided to the application

With GPU virtualization:
- Many GPUs in the cluster can be provided to the application

Logically, more GPUs for a single application.
1: more GPUs for a single application

Detected 4 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 (644213264 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=чу65536,65535), 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!
1: more GPUs for a single application

- MonteCarlo Multi-GPU (from NVIDIA samples)

FDR InfiniBand + NVIDIA Tesla K20

Graphs showing performance with respect to the number of GPUs:
- Higher is better for Options per second
- Lower is better for Computation Time (ms)
Benefit #2: busy CPUs do not block GPUs
2: busy CPU cores do not block GPUs

Physical configuration

Logical connections

Logical configuration
Benefit #3: increased cluster performance with Slurm
3: increased cluster performance with Slurm

- GPUs can be shared among jobs running in remote clients
  - Job scheduler required for coordination
  - Slurm was selected
Applications used for tests:

- GPU-Blast (21 seconds; 1 GPU; 1599 MB)
- LAMMPS (15 seconds; 4 GPUs; 876 MB)
- MCUDA-MEME (165 seconds; 4 GPUs; 151 MB)
- GROMACS (167 seconds)
- NAMD (11 minutes)
- BarraCUDA (10 minutes; 1 GPU; 3319 MB)
- GPU-LIBSVM (5 minutes; 1 GPU; 145 MB)
- MUMmerGPU (5 minutes; 1 GPU; 2804 MB)

Workload size: 400 jobs
3: increased cluster performance with Slurm

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

16 GPU nodes + 1 controller node
3: increased cluster performance with Slurm
Benefit #4: GPU migration
Job granularity instead of GPU granularity
Benefit #5: virtual machines can easily access GPUs
5: virtual machines can easily access GPUs

- The GPU is assigned by using PCI passthrough exclusively to a single virtual machine
- Concurrent usage of the GPU is not possible

![Diagram showing computer hosting several KVM virtual machines]
5: virtual machines can easily access GPUs

Diagram showing a computer hosting several KVM virtual machines with rCUDA client vGPU access to GPUs through an InfiniBand Fabric. The diagram illustrates the network availability for high and low performance scenarios.
5: virtual machines can easily access GPUs

FDR InfiniBand + K20 !!
Benefit #6: cheaper cluster upgrade
6: cheaper cluster upgrade

- Let’s suppose that a cluster without GPUs needs to be upgraded to use GPUs

No GPU

- GPUs require large power supplies
  - Are power supplies already installed in the nodes large enough?
- GPUs require large amounts of space
  - Does current form factor of the nodes allow to install GPUs?

The answer to both questions is usually “NO”
Approach 1: augment the cluster with some CUDA GPU-enabled nodes → only those GPU-enabled nodes can execute accelerated applications
6: cheaper cluster upgrade

Approach 2: augment the cluster with some rCUDA servers → all nodes can execute accelerated applications

GPU-enabled
6: cheaper cluster upgrade

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

16 nodes without GPU + 1 node with 4 GPUs
6: cheaper cluster upgrade

![Bar charts showing execution time, energy consumption, and average GPU utilization for large workload comparing CUDA and rCUDA.]
Conclusions
• Benefits of remote GPU virtualization:
  1. More GPUs for a single application
  2. Busy CPUs do not block GPUs
  3. Increased cluster performance with Slurm
  4. Migration of GPU jobs
  5. Virtual machines can easily access GPUs
  6. Cheaper cluster upgrade

rCUDA is a development by Technical University of Valencia
Get a free copy of rCUDA at
http://www.rcuda.net
More than 650 requests world wide

@rcuda_

rCUDA is a development by Technical University of Valencia
Thanks!
Questions?

rCUDA is a development by Technical University of Valencia