rCUDA:
desde máquinas virtuales a clústers mixtos CPU-GPU

Federico Silla
Universitat Politècnica de València
rCUDA: from virtual machines to hybrid CPU-GPU clusters

Federico Silla
Universitat Politècnica de València
What is rCUDA?
Basics of GPU computing

Remark:
GPUs can only be used within the node they are attached to
Remark:
GPUs can only be used within the node they are attached to.
A different approach: remote GPU virtualization
A different approach: remote GPU virtualization

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

rCUDA is a development by Universitat Politècnica de València
Access to remote GPU is transparent to applications: no source code modification is needed.
Basics of rCUDA

Access to remote GPU is transparent to applications: no source code modification is needed.
Basics or rCUDA

Access to remote GPU is transparent to applications: no source code modification is needed.

rCUDA is a development by Universitat Politècnica de València.
rCUDA supports RDMA transfers
rCUDA envision

- rCUDA allows a new vision of a GPU deployment, moving from the usual cluster configuration …

… to the following one:
Performance of rCUDA?
Performance of rCUDA

- K20 GPU and FDR InfiniBand
- K40 GPU and EDR InfiniBand
Performance of rCUDA

P100 GPU and EDR InfiniBand

BarraCUDA

CUDA-MEME

Lower is better

Lower is better

F. Silla @ HPC ADMINTECH 2018
Performance of data movements among GPUs
Performance of data movements among GPUs

Higher is better
Performance of data movements to/from GPUs

CUDA

rCUDA

NODE

GPU
Memory

CPU
Memory

PCle

NODE 0
rCUDA client

NODE 1
rCUDA server

GPU
Memory

CPU
Memory

PCle

Network fabric
Performance of data movements to/from GPUs

CPU to GPU

Higher is better

GPU to CPU
Performance of data movements to/from GPUs

**CPU to GPU**

**GPU to CPU**

Higher is better
Performance of data movements to/from GPUs

**CPU to GPU**

![Graph showing performance of data movements from CPU to GPU]

Higher is better

**GPU to CPU**

![Graph showing performance of data movements from GPU to CPU]
Performance of data movements to/from GPUs

Client side

Application

CUDA API

rCUDA client

TCP/IP module
InfiniBand module
RoCE module

Server side

New communication module in progress

rCUDA server

common communication API

CUDA libraries

CUDA driver

Software

Hardware

Network

GPU
Benefits of rCUDA?
Benefits of rCUDA:

1. Many GPUs for an application
2. Server consolidation
3. GPU acceleration for virtual machines
4. Increased cluster throughput
Providing many GPUs to an application with rCUDA

Interconnection Network
Providing many GPUs to an application with rCUDA

K20 GPUs and FDR InfiniBand

MonteCarlo multi-GPU program running in 14 NVIDIA Tesla K20 GPUs
Providing many GPUs to an application with rCUDA

Detected 64 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 (6442123264 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): 10=(65536), 2D=(65536,65535), 3D=(2048,2048,2048)
Max Layered Texture Size (dim) x layers: 10=(16384) x 2048, 2D=(16384,16384) x 2048
Total amount of constant memory: 65536 bytes
Total amount of shared memory per block: 4096 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
Providing many GPUs to an application with rCUDA

Work in progress!!

non-optimized (yet) version of rCUDA!!!
Server consolidation with rCUDA

![Diagram showing GPU utilization and server consolidation]
The GPU-Blast application is migrated up to 5 times among K40 GPUs.

- The aggregated volume of GPU data is 1300 MB (consisting of 9 memory regions).

The “Reference” line is the execution time of the application when using CUDA with a local GPU and without any migration.
Virtual machines may need access to GPUs

- How to access the GPU in the native domain from inside of virtual machines?
Virtual machines may need access to GPUs

- The GPU is assigned by using PCI passthrough exclusively to a single virtual machine
- Concurrent usage of the GPU is not possible
• If InfiniBand is available, the rCUDA server can be placed in another node

• Several GPUs can be provided to the VMs, either in a single remote node or in several remote nodes
Using rCUDA to access the GPU

- When InfiniBand is not available, the rCUDA server may be placed in the native domain and the rCUDA client would be placed inside the VMs.

- High performance network is not available.

- This configuration allows the use of more than one GPU at the host.

- When InfiniBand is not available, the rCUDA server may be placed in the native domain and the rCUDA client would be placed inside the VMs.

- The virtual network provided by the hypervisor would be used to exchange data between the rCUDA clients and the rCUDA server.
Using rCUDA to access the GPU
Increased cluster throughput

One rCUDA box serves multiple clients

Network

GPU

GPU
Increased cluster throughput

1. BarraCUDA
2. CUDA-MEME
3. CUDASW++
4. GPU-Blast
5. Gromacs
6. Magma

Lower is better - 58%
Increased cluster throughput
One more benefit: Heterogeneous\textsuperscript{2} environments
rCUDA is available for the x86, POWER and ARM processors
Performance of rCUDA on ARM systems
From ARM to x86 with rCUDA
Work in progress. A couple of applications have been already analyzed:

1. **Cloverleaf**: a mini-app that solves the compressible Euler equations on a Cartesian grid

2. **Flow**: a mini-app that implements a 2D hydrodynamics simulator
Application performance: Cloverleaf

Single node executions

Lower is better

Estimation over multiple nodes
Rough energy estimation:

- ThunderX TDP = 80 watts
- P100 TDP = 250 watts
- Xeon TDP = 140 watts

40*80 versus 1*80+3*250+2*140
3200 watts versus 1110 watts

Application performance: Cloverleaf

Lower is better

Estimation over multiple nodes
Lower is better

Single node executions

Estimation over multiple nodes
Application performance: Flow

Rough energy estimation:
- ThunderX TDP = 80 watts
- P100 TDP = 250 watts
- Xeon TDP = 140 watts

60*80 versus 1*80+3*250+2*140
4800 watts versus 1110 watts

Single node executions

Lower is better

Estimation over multiple nodes
Hybrid CPU-GPU clusters
Hybrid CPU-GPU clusters

High density ARM-based nodes

ARM-based servers

Servers with GPUs

Network
Hybrid CPU-GPU clusters

High density ARM-based nodes

ARM-based servers

rCUDA clients

rCUDA servers

Client side | Server side

Application

CUDA API

rCUDA client

rCUDA server

CUDA libraries

Software

Hardware

Network

F. Silla @ HPC ADMINTECH 2018
rcuda is a development by Universitat Politècnica de València, Spain

More than 900 requests worldwide

Get a free copy of CUDA at

http://www.rCUDA.net

@rcuda
· Tony Díaz · Pablo Higueras · Javier Prades · Jaime Sierra
· Cristian Peñaranda · Federico Silla · Carlos Reaño

rCUDA is a development by Universitat Politècnica de València, Spain
Funded by Agencia Valenciana de la Innovación, Generalitat Valenciana