

### A general-purpose virtualization service for HPC on cloud computing: an application to GPUs

*R.Montella, G.Coviello, G.Giunta\* G. Laccetti<sup>#</sup>, F. Isaila, J. Garcia Blas*°

\*Department of Applied Science – University of Napoli Parthenope \* Department of Mathematics and Applications – University of Napoli Federico II \*Department of Computer Science – University of Madrid Carlos III

# Outline

- Introduction and contextualization
- GVirtuS: Generic Virtualization Service
- GPU virtualization
- High performance cloud computing
- Who uses GVirtuS
- Conclusions and ongoing projects



### Introduction and contextualization

- High Performance Computing
- Grid computing
- Many core technology
- GPGPUs
- Virtualization
- Cloud computing





### High Performance Cloud Computing



## GVirtuS

- Generic Virtualization Service
- Framework for split-driver based abstraction components
- Plug-in architecture
- Independent form
  - Hypervisor
  - Communication
  - Target of virtualization
- High performance:
  - Enabiling transparent virtualization
  - Wth overall performances not too far from un-virtualized machines

## Split-Driver approach

- Split-Driver
  - Hardware access by priviledged domain.
  - Unpriviledged domains access the device using a frontend/backhend approach
- Frontend (FE):
  - Guest-side software component.
  - Stub: redirect requests to the backend.
- Backend (BE):
  - Mange device requests.
  - Device multiplexing.



### GVirtuS approach

- GVirtuS Frontend
  - Dyinamic loadable library
  - Same application binary interface
  - Run on guest user space

Unpriviledged Domain





- GVirtuS Backend
  - Server application
  - Run in host user space
  - Concurrent requests

## The Communicator

- Provides a high performance communication between virtual machines and their hosts.
- The choice of the hypervisor deeply affects the efficiency of the communication.



Hypervisor	FE/BE comm	Notes
No hypervisor	Unix Sockets	Used for testing purposes
Generic	TCP/IP	Used for communication testing purposes, but interesting
Xen	XenLoop	<ul> <li>•runs directly on the top of the hardware through a custom Linux kernel</li> <li>•provides a communication library between guest and host machines</li> <li>•implements low latency and wide bandwidth TCP/IP and UDP connections</li> <li>•application transparent and offers an automatic discovery of the supported VMS</li> </ul>
VMware	Virtual Machine Communication Interface (VMCI)	<ul> <li>commercial hypervisor running at the application level.</li> <li>provides a datagram API to exchange small messages</li> <li>a shared memory API to share data,</li> <li>an access control API to control which resources a virtual machine can access</li> <li>and a discovery service for publishing and retrieving resources.</li> </ul>
KVM/QEMU	VMchannel	•Linux loadable kernel module now embedded as a standard component •supplies a high performance guest/host communication

## An application: Virtualizing GPUs

### • GPUs

- Hypervisor independent
- Communicator independent
- GPU independent



## GVirtuS - CUDA

- Currently full threadsafe support to
  - CUDA drivers
  - CUDA runtime
  - OpenCL
- Partially supporting (it is needed more work)
  - OpenGL integration



### **GVirtuS – libcudard.so**



# **Choices and Motivations**



Hypervisor	FE/BE Comm	Open Src	Running as	Official CUDA Drivers
Xen	Xen Loon	Vec	Karnal	No
	Xell Loop	103		
VM-Ware	VMCI	No	Application	Shares the host OS ones
KVM/QEMU	vmSocket	Yes	Loadable Kernel Module	Shares the host OS ones

- We focused on VMware and KVM hypervisors.
- vmSocket is the component we have designed to obtain a high performance communicator
- **vmSocket** exposes Unix Sockets on virtual machine instances thanks to a QEMU device connected to the virtual PCI bus.

# vmSocket: virtual PCI device

- Programming interface:
  - Unix Socket
- Communication between guest and host:
  - Virtual PCI interface
  - QEMU has been modified

### cudaMemcpyHostToDevice

ntend

BackEnd



### **FE/BE interaction efficiency:**

•there is no mapping between guest memory and device memory
•the memory device pointers are never de-referenced on the host side
•CUDA kernels are executed on the BE where the pointers are fully consistent.

### **Performance Evaluation**

- CUDA Workstation
  - Genesis GE-i940 Tesla
  - i7-940 2,93 133 GHz fsb, Quad Core hyperthreaded 8 Mb cache CPU and 12Gb RAM.
  - 1 nVIDIA Quadro FX5800 4Gb RAM video card
  - 2 nVIDIA Tesla C1060 4 Gb RAM
- The testing system:
  - Ubuntu 10.04 Linux
  - nVIDIA CUDA Driver, and the SDK/Toolkit version 4.0.
  - VMware vs. KVM/QEMU (using different communicators).



## **GVirtuS-CUDA runtime performances**

#### **Results:**

O: No virtualization, no accleration (blank)
1: Acceleration without virtualization (target)

•2,3: Virtualization with no acceleration

 •4...6: GPU acceleration, Tcp/Ip communication ⇒ Similar performances due to communication overhead

•7: GPU acceleration using GVirtuS, Unix Socket based communication

•8,9: GVirtuS virtualization

⇒Good performances, no so far from the target ⇒4..6 better performances than 0

### **Evaluation:**

**CUDA SDK benchmarks** 

#	Hypervisor	Comm.	Histogram	matrixMul	scalarProd
0	Host	CPU	100.00%	100.00%	100.00%
1	Host	GPU	9.50%	9.24%	8.37%
2	Kvm	CPU	105.57%	99.48%	106.75%
3	VM-Ware	CPU	103.63%	105.34%	106.58%
4	Host	Тср/Ір	67.07%	52.73%	40.87%
5	Kvm	Тср/Ір	67.54%	50.43%	42.95%
6	VM-Ware	Тср/Ір	67.73%	50.37%	41.54%
7	Host	AfUnix	11.72%	16.73%	9.09%
8	Kvm	vmSocket	15.23%	31.21%	10.33%
9	VM-Ware	vmcl	28.38%	42.63%	18.03%

Computing times as Host-CPU rate

## **Distributed GPUs**

### Hilights:

•Using theTcp/Ip Communicator FE/BE could be on different machines.

•Real machines can access remote GPUs.

### **Applications**:

•GPU for embedded systems as network machines

•High Performance Cloud Computing



### **High Performance Cloud Computing**

- Ad hock performance test for benchmarking
- Virtual cluster on local computing cloud
- Benchmark:
  - Matrix-matrix multiplication
  - 2 parallelims levels: distributed memory and GPU

### • Results:





Input data size (a=10<sup>3</sup>)

### GVirtuS in the world

- GPU support to OpenStack cloud software
  - Heterogeneous cloud computing John Paul Walters et Al. University of Southern California / Information Sciences Institute
  - <u>http://wiki.openstack.org/HeterogeneousGpuAcceleratorSupport</u>
- HPC at NEC Labs of America in Princeton, University of Missouri Columbia, Ohio State University Columbus
  - Supporting GPU Sharing in Cloud Environments with a Transparent Runtime Consolidation Framework
  - Awarded as the best paper at HPDC2011



### Download, taste, contribute!

#### **UniParthenope Open Source Lab**

... only who writes ugly code doesn't want to show it ....



MAR gVirtuS 04 & cig = 96 comments

### A General Purpose GPU transparent virtualization component

gVirtuS tries to fill the gap between in-house hosted computing clusters, equipped with GPGPUs devices, and pay-for-use high performance virtual clusters deployed via public or private computing clouds. gVirtuS allows an instanced virtual machine to access GPGPUs in a transparent way, with an overhead slightly greater than a real machine/GPGPU setup. gVirtuS is hypervisor independent, even it currently virtualizes nVIDIA CUDA based GPUs, it is not limited to a specific brand technology. The performance of the components of gVirtuS is assessed through a suite of tests in different deployment scenarios, such as providing GPGPU power to cloud computing based HPC clusters and sharing remotely hosted GPGPUs among HPC nodes.

#### Common pages

- Installation
- Downloads

#### Contacts

- Department of Applied Science (http://dsa.uniparthenope.it)
- Applied High-Performance Scientific Computing Research Laboratory (http://dsa.uniparthenope.it/Imncp)
- · University of Napoli Parthenope (http://www.uniparthenope.it)
- Mailinglist: http://osl.uniparthenope.it/mailman/listinfo/gvirtus
- · Prof. Giulio Giunta (giulio.giunta@uniparthenope.it)
- Dr. Raffaele Montella (raffaele.montella@uniparthenope.it)
- Giuseppe Agrillo (giuseppe.agrillo@uniparthenope.it)
- Giuseppe Coviello (giuseppe.coviello@uniparthenope.it)



Projects

 AbstractInstrument
 Code
 gVirtuS
 Installation
 Downloads
 Code

 JaClouX

Installation

Pages

· About

Events

Downloads Documentation and

Tutorials

Code

Petroula Code

WW3Utilities

Code GradsSharp

Code

· People

#### Recent Posts

- JaClouX beta-0.1
- UniParthenope OpenSource Lab Party: Talk Schedule

Hereine and the second second second

• <u>http://osl.uniparthenope.it/projects/gvirtus</u> GPL/LGPL License

### Conclusions

- The GVirtuS generic virtualization and sharing system enables thin Linux based virtual machines to use hosted devices as nVIDIA GPUs.
- The GVirtuS-CUDA stack permits to accelerate virtual machines with a small impact on overall performance respect to a pure host/gpu setup.
- GVirtuS can be easily extended to other enabled devices as high performance network devices

## **Ongoing projects**

- Elastic Virtual Parallel File System
- MPI wrap for High Performance Cloud Computing
- XEN Support (is a big challenge!)

