

## Building HPC Cloud with InfiniBand: Efficient Support in MVAPICH2 for KVM, Docker, Singularity, OpenStack, and SLURM

**Tutorial and Demo at MUG 2017** 

by

#### Xiaoyi Lu

The Ohio State University

E-mail: luxi@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~luxi

#### **Cloud Computing and Virtualization**



- Cloud Computing focuses on maximizing the effectiveness of the shared resources
- Virtualization is the key technology for resource sharing in the Cloud
- Widely adopted in industry computing environment
- Gartner and International Data Corporation (IDC) have issued a forecast predicting public cloud spending will balloon to \$203.4 billion by 2020 (Courtesy: https://www.mspinsights.com/doc/cloud-spending-will-reach-billion-in-0001)

#### **Overview of Hypervisor-/Containers-based Virtualization**







- Hypervisor-based virtualization provides higher isolation, multi-OS support, etc.
- Container-based technologies (e.g., Docker) provide lightweight virtualization solutions
- Container-based virtualization share host kernel by containers

### **Drivers of Modern HPC Cluster and Cloud Architecture**



Multi-/Many-core Processors

High Performance Interconnects -InfiniBand (with SR-IOV) <1usec latency, 200Gbps Bandwidth>



SSDs, Object Storage Clusters



- Multi-core/many-core technologies, Accelerators
- Large memory nodes
- Solid State Drives (SSDs), NVM, Parallel Filesystems, Object Storage Clusters
- Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
- Single Root I/O Virtualization (SR-IOV)



### Single Root I/O Virtualization (SR-IOV)

- Single Root I/O Virtualization (SR-IOV) is providing new opportunities to design HPC cloud with very little low overhead
- Allows a single physical device, or a Physical Function (PF), to present itself as multiple virtual devices, or Virtual Functions (VFs)
- VFs are designed based on the existing non-virtualized PFs, no need for driver change
- Each VF can be dedicated to a single VM through PCI pass-through
- Work with 10/40 GigE and InfiniBand



#### **Building HPC Cloud with SR-IOV and InfiniBand**

- High-Performance Computing (HPC) has adopted advanced interconnects and protocols
  - InfiniBand
  - 10/40/100 Gigabit Ethernet/iWARP
  - RDMA over Converged Enhanced Ethernet (RoCE)
- Very Good Performance
  - Low latency (few micro seconds)
  - High Bandwidth (200 Gb/s with HDR InfiniBand)
  - Low CPU overhead (5-10%)
- OpenFabrics software stack with IB, iWARP and RoCE interfaces are driving HPC systems
- How to Build HPC Clouds with SR-IOV and InfiniBand for delivering optimal performance?

# **Broad Challenges in Designing Communication and I/O Middleware for HPC on Clouds**

- Virtualization Support with Virtual Machines and Containers
  - KVM, Docker, Singularity, etc.
- Communication coordination among optimized communication channels on Clouds
  - SR-IOV, IVShmem, IPC-Shm, CMA, etc.
- Locality-aware communication
- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
- Scalable Collective communication
  - Offload; Non-blocking; Topology-aware
- Balancing intra-node and inter-node communication for next generation nodes (128-1024 cores)
  - Multiple end-points per node
- NUMA-aware communication for nested virtualization
- Integrated Support for GPGPUs and Accelerators
- Fault-tolerance/resiliency
  - Migration support with virtual machines
- QoS support for communication and I/O
- Support for Hybrid MPI+PGAS programming (MPI + OpenMP, MPI + UPC, MPI + OpenSHMEM, MPI+UPC++, CAF, ...)
- Energy-Awareness
- Co-design with resource management and scheduling systems on Clouds
  - OpenStack, Slurm, etc.

# **Approaches to Build HPC Clouds**

- MVAPICH2-Virt with SR-IOV and IVSHMEM
  - Standalone, OpenStack
- SR-IOV-enabled VM Migration Support in MVAPICH2
- MVAPICH2 with Containers (Docker and Singularity)
- MVAPICH2 with Nested Virtualization (Container over VM)
- MVAPICH2-Virt on SLURM
  - SLURM alone, SLURM + OpenStack

#### **Overview of the MVAPICH2 Project**

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Started in 2001, First version available in 2002.
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - Used by more than 2,800 organizations in 85 countries
  - More than 424,000 (> 0.4 million) downloads from the OSU site directly
  - Empowering many TOP500 clusters (Nov '16 ranking)
    - 1<sup>st</sup> ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 15<sup>th</sup> ranked 241,108-core cluster (Pleiades) at NASA
    - 20<sup>th</sup> ranked 519,640-core cluster (Stampede) at TACC
    - 44<sup>th</sup> ranked 76,032-core cluster (Tsubame 2.5) at Tokyo Institute of Technology and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - <u>http://mvapich.cse.ohio-state.edu</u>
- Empowering Top500 systems for over a decade
  - System-X from Virginia Tech (3<sup>rd</sup> in Nov 2003, 2,200 processors, 12.25 TFlops) ->
    Sunway TaihuLight at NSC, Wuxi, China (1<sup>st</sup> in Nov'16, 10,649,640 cores, 93 PFlops)



VM/Container-aware Design is already publicly available in MVAPICH2-Virt!

#### **Network Based Computing Laboratory**

#### **MVAPICH2 Software Family**

High-Performance Pa	sh-Performance Parallel Programming Libraries				
MVAPICH2	Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE				
MVAPICH2-X	Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime				
MVAPICH2-GDR	Optimized MPI for clusters with NVIDIA GPUs				
MVAPICH2-Virt	High-performance and scalable MPI for hypervisor and container based HPC cloud				
MVAPICH2-EA	Energy aware and High-performance MPI				
MVAPICH2-MIC Optimized MPI for clusters with Intel KNC					
Microbenchmarks					
ОМВ	Microbenchmarks suite to evaluate MPI and PGAS (OpenSHMEM, UPC, and UPC++) libraries for CPUs and GPUs				
Tools					
OSU INAM	Network monitoring, profiling, and analysis for clusters with MPI and scheduler integration				
OEMT	Utility to measure the energy consumption of MPI applications				

#### HPC on Cloud Computing Systems: Challenges Addressed by OSU So Far

Applications

**HPC and Big Data Middleware** 

HPC (MPI, PGAS, MPI+PGAS, MPI+OpenMP, etc.)

Resource Management and Scheduling Systems for Cloud Computing (OpenStack Nova, Heat; Slurm)

Communication and I/O Library							
Communication Channels (SR-IOV, IVShmem, IPC-Shm, CMA)	Locality- and NUMA-aware Communication	Virtualization (Hypervisor and Container)					
Fault-Tolerance & Consolidation (Migration)	QoS-aware	Future Studies					
Networking Technologies (InfiniBand, Omni-Path, 1/10/40/100 GigE and Intelligent NICs)	Commodity Computing System Architectures (Multi- and Many-core architectures and accelerators)	Storage Technologies (HDD, SSD, NVRAM, and NVMe-SSD)					

Network Based Computing Laboratory

# **Overview of OSU Solutions for Building HPC Clouds - MVAPICH2-Virt with SR-IOV and IVSHMEM**



**Network Based Computing Laboratory** 

MUG 2017

12

#### **Intra-Node Inter-VM Performance**



Latency at 64 bytes message size: SR-IOV(IB\_Send) - 0.96µs, IVShmem - 0.2µs

Can IVShmem scheme benefit MPI communication within a node on SR-IOV enabled InfiniBand clusters?

#### **Overview of MVAPICH2-Virt with SR-IOV and IVSHMEM**

- Redesign MVAPICH2 to make it virtual machine aware
  - SR-IOV shows near to native performance for inter-node point to point communication
  - IVSHMEM offers shared memory based data access across co-resident VMs
  - Locality Detector: maintains the locality information of co-resident virtual machines
  - Communication Coordinator: selects the communication channel (SR-IOV, IVSHMEM) adaptively



J. Zhang, X. Lu, J. Jose, R. Shi, D. K. Panda. Can Inter-VM Shmem Benefit MPI Applications on SR-IOV based Virtualized InfiniBand Clusters? Euro-Par, 2014

J. Zhang, X. Lu, J. Jose, R. Shi, M. Li, D. K. Panda. High Performance MPI Library over SR-IOV Enabled InfiniBand Clusters. HiPC, 2014

#### **MVAPICH2-Virt with SR-IOV and IVSHMEM over OpenStack**

- OpenStack is one of the most popular open-source solutions to build clouds and manage virtual machines
- Deployment with OpenStack
  - Supporting SR-IOV configuration
  - Supporting IVSHMEM configuration
  - Virtual Machine aware design of MVAPICH2 with SR-IOV
- An efficient approach to build HPC Clouds with MVAPICH2-Virt and OpenStack

J. Zhang, X. Lu, M. Arnold, D. K. Panda. MVAPICH2 over OpenStack with SR-IOV: An Efficient Approach to Build HPC Clouds. CCGrid, 2015



#### Intra-node Inter-VM Point-to-Point Performance



- Compared to SR-IOV, up to 84% and 158% improvement on Latency & Bandwidth
- Compared to Native, 3%-8% overhead on both Latency & Bandwidth

#### **Application Performance (NAS & P3DFFT)**



- Proposed design delivers up to 43% (IS) improvement for NAS
- Proposed design brings 29%, 33%, 29% and 20% improvement for INVERSE, RAND, SINE and SPEC

#### **Application-Level Performance on Chameleon**



- 32 VMs, 6 Core/VM
- Compared to Native, 2-5% overhead for Graph500 with 128 Procs
- Compared to Native, 1-9.5% overhead for SPEC MPI2007 with 128 Procs

# **Approaches to Build HPC Clouds**

- MVAPICH2-Virt with SR-IOV and IVSHMEM
  - Standalone, OpenStack
- SR-IOV-enabled VM Migration Support in MVAPICH2
- MVAPICH2 with Containers (Docker and Singularity)
- MVAPICH2 with Nested Virtualization (Container over VM)
- MVAPICH2-Virt on SLURM
  - SLURM alone, SLURM + OpenStack

### **Overview of OSU Solutions for Building HPC Clouds: SR-IOVenabled VM Migration Support in MVAPICH2**



**Network Based Computing Laboratory** 

#### **Execute Live Migration with SR-IOV Device**

	[root@sandy1:migration]\$	
	[root@sandy1:migration]\$ssh sandy3-vm1 lspci	
	root@sandy3-vm1's password:	
	00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)	
	00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]	
	00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]	
	00:01.2 USB controller: Intel Corporation 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)	
	00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)	
	00:02.0 VGA compatible controller: Cirrus Logic GD 5446	
-	00.03.0 Ethernet controller: Red Hat, Inc Virtio network device	
	00:04.0 Infiniband controller: Mellanox Technologies MT27700 Family [ConnectX-4 Virtual Function]	
-	00.05.0 Unclassified device [00ff]: Red Hat, Inc Vintio memory balloon	
	[root@sandy1:migration]\$	
-	[root@sandy1:migration]\$virsh migrateliverdma-pin-allmigrateuri rdma://sandy3-ib sandy1-vm1 qemu://sandy3-ib/system 🛸	-
_	error: Requested operation is not valid: domain has assigned non-USB host devices	- 1
1		
	[root@sandy1:miaration]\$	

0

### **Overview of Existing Migration Solutions for SR-IOV**

	Platform	NIC	No Guest OS Modification	Device Driver Independent	Hypervisor Independent
Zhai, etc (Linux bonding driver)	Ethernet	N/A	Yes	No	Yes (Xen)
Kadav, etc (shadow driver)	Ethernet	Intel Pro/1000 gigabit NIC	No	Yes	No (Xen)
Pan, etc (CompSC)	Ethernet	Intel 82576, Intel 82599	Yes	No	No (Xen)
Guay, etc	InfniBand	Mellanox ConnectX2 QDR HCA	Yes	No	Yes (Oracle VM Server (OVS) 3.0.)
Han	Ethernet	Huawei smart NIC	Yes	No	No (QEMU+KVM)
Xu, etc (SRVM)	Ethernet	Intel 82599	Yes	Yes	No (VMware EXSi)

Can we have a hypervisor-independent and device driver-independent solution for InfiniBand based HPC Clouds with SR-IOV?

#### **High Performance SR-IOV enabled VM Migration Framework**



- Consist of SR-IOV enabled IB Cluster and External Migration Controller
- Detachment/Re-attachment of virtualized devices: Multiple parallel libraries to coordinate VM during migration (detach/reattach SR-IOV/IVShmem, migrate VMs, migration status)
- **IB Connection**: MPI runtime handles IB connection suspending and reactivating
- Propose Progress Engine (PE) and Migration Thread based (MT) design to optimize VM migration and MPI application performance

#### J. Zhang, X. Lu, D. K. Panda. High-Performance Virtual Machine Migration Framework for MPI Applications on SR-IOV enabled InfiniBand Clusters. IPDPS, 2017

#### **Parallel Libraries in Controller**



#### • Network Suspend Trigger:

- Issue VM migration request from admin
- Implement on top of MPI startup channel for scalability
- Ready-to-Migrate Detector and Migration Done Detector:
  - Periodically detect states on IVShmem regions
  - Maintain a counter to keep track of all the states, utilize MPI reduce to collect
- Migration Trigger:
  - Carrying out VM live migration operation
  - Detaching/re-attaching SR-IOV VF and IVShmem devices
- Network Reactivate Notifier:
  - Work after VM migration to the targeted host
  - Notify MPI processes to reactivate communication channels

#### **Network Based Computing Laboratory**

### **Performance Evaluation of VM Migration Framework**



Breakdown of VM migration

- Compared with the TCP, the RDMA scheme reduces the total migration time by 20%
- Total time is dominated by `Migration' time; Times on other steps are similar across different schemes
- Proposed migration framework could reduce up to 51% migration time •

### **Performance Evaluation of VM Migration Framework**



- Migrate a VM from one machine to another while benchmark is running inside
- Proposed MT-based designs perform slightly worse than PE-based designs because of lock/unlock
- No benefit from MT because of NO computation involved

#### **Overlapping Evaluation**



- fix the communication time and increase the computation time
- 10% computation, partial migration time could be overlapped with computation in MTtypical

Network Bada companyation porcentage increases, mogezohance for overlapping

## **Performance Evaluation with Applications**



- 8 VMs in total and 1 VM carries out migration during application running
- Compared with NM, MT- worst and PE incur some overhead
- MT-typical allows migration to be completely overlapped with computation

# **Approaches to Build HPC Clouds**

- MVAPICH2-Virt with SR-IOV and IVSHMEM
  - Standalone, OpenStack
- SR-IOV-enabled VM Migration Support in MVAPICH2
- MVAPICH2 with Containers (Docker and Singularity)
- MVAPICH2 with Nested Virtualization (Container over VM)
- MVAPICH2-Virt on SLURM
  - SLURM alone, SLURM + OpenStack

### **Overview of OSU Solutions for Building HPC Clouds: MVAPICH2** with Containers (Docker and Singularity)



Network Based Computing Laboratory

#### **Benefits of Containers-based Virtualization for HPC on Cloud**



- Container has less overhead than VM
- BFS time in Graph 500 significantly increases as the number of container increases on one host. Why?
  J. Zhang, X. Lu, D. K. Panda. Performance Characterization of Hypervisor- and Container-Based Virtualization for HPC on SR-IOV Enabled InfiniBand Clusters. IPDRM, IPDPS Workshop, 2016
  Network Based Computing Laboratory

#### **Containers-based Design: Issues, Challenges, and Approaches**

- What are the performance bottlenecks when running MPI applications on multiple containers per host in HPC cloud?
- Can we propose a new design to overcome the bottleneck on such container-based HPC cloud?
- Can optimized design deliver near-native performance for different container deployment scenarios?
- Locality-aware based design to enable CMA and Shared memory channels for MPI communication across co-resident containers



J. Zhang, X. Lu, D. K. Panda. High Performance MPI Library for Container-based HPC Cloud on InfiniBand Clusters. ICPP, 2016

#### **MPI Point-to-Point and Collective Performance**



- Two containers are deployed on the same socket and different socket
- 256 procs totally (4 pros/container, 64 containers across 16 nodes evenly)
- Up to 79% and 86% improvement for Point-to-Point and MPI\_Allgather, respectively (Cont-Opt vs. Cont-Def)
- Minor overhead, compared with Native performance (Cont-\*-Opt vs. Native-\*)

#### **Application-Level Performance on Docker with MVAPICH2**



- 64 Containers across 16 nodes, pining 4 Cores per Container
- Compared to Container-Def, up to 11% and 73% of execution time reduction for NAS and Graph 500
- Compared to Native, less than 9 % and 5% overhead for NAS and Graph 500

### **MVAPICH2 Intra-Node and Inter-Node Point-to-Point Performance on Singularity**



- Less than 8% overhead on latency
- Less than 3% overhead on BW

#### **MVAPICH2** Collective Performance on Singularity



- 512 Processes across 32 nodes
- Less than 8% and 9% overhead for Bcast and Allreduce, respectively
# **Application-Level Performance on Singularity with MVAPICH2**



- 512 Processes across 32 nodes
- Less than 7% and 6% overhead for NPB and Graph500, respectively

# **Approaches to Build HPC Clouds**

- MVAPICH2-Virt with SR-IOV and IVSHMEM
  - Standalone, OpenStack
- SR-IOV-enabled VM Migration Support in MVAPICH2
- MVAPICH2 with Containers (Docker and Singularity)
- MVAPICH2 with Nested Virtualization (Container over VM)
- MVAPICH2-Virt on SLURM
  - SLURM alone, SLURM + OpenStack

# **Overview of OSU Solutions for Building HPC Clouds: MVAPICH2** with Nested Virtualization



**Network Based Computing Laboratory** 

# **Nested Virtualization: Containers over Virtual Machines**

VM1	··-·	VM2							
Container1	Container2	Container3	Container4						
App Stack	App Stack	ll App Stack	App Stack						
bins/      libs	bins/								
Docke	r Engine	Docke	Docker Engine						
Re	dhat	Ut	ountu						
	Hypervisor								
Host OS									
	Hardware								

- Useful for live migration, sandbox application, legacy system integration, software deployment, etc.
- Performance issues because of the redundant call stacks (two-layer virtualization) and isolated physical resources

# **Usage Scenario of Nested Virtualization**



- VM provides good isolation and security so that the applications and workloads of users A and B will not interfere with each other
- Root permission of VM can be given to do special configuration
- Docker brings an effective, standardized and repeatable way to port and distribute the applications and workloads

### **Multiple Communication Paths in Nested Virtualization**



- Different VM placements introduce multiple communication paths on container level
  - 1. Intra-VM Intra-Container (across core 4 and core 5)
  - 2. Intra-VM Inter-Container (across core 13 and core 14)
  - 3. Inter-VM Inter-Container (across core 6 and core 12)
  - 4. Inter-Node Inter-Container (across core 15 and the core on remote node)

### **Performance Characteristics on Communication Paths**



- Two VMs are deployed on the same and different socket, respectively
- \*-Def and Inter-VM Inter-Container-1Layer have similar performance
- Still large gap compared to native performance with just 1layer design

1Layer\* - J. Zhang, X. Lu, D. K. Panda. High Performance MPI Library for Container-based HPC Cloud on InfiniBand, ICPP, 2016

MUG 2017

# **Challenges of Nested Virtualization**

• How to further reduce the performance overhead of running applications on the nested virtualization environment?

• What are the impacts of the different VM/container placement schemes for the communication on the container level?

• Can we propose a design which can adapt these different VM/container placement schemes and deliver near-native performance for nested virtualization environments?

# **Overview of Proposed Design in MVAPICH2**



Two-Layer Locality Detector: Dynamically detecting MPI processes in the coresident containers inside one VM as well as the ones in the co-resident VMs

### Two-Layer NUMA Aware Communication Coordinator: Leverage nested locality info, NUMA architecture info and message to select appropriate communication channel

J. Zhang, X. Lu, D. K. Panda. Designing Locality and NUMA Aware MPI Runtime for Nested Virtualization based HPC Cloud with SR-IOV Enabled InfiniBand, VEE, 2017

### Inter-VM Inter-Container Pt2Pt (Intra-Socket)



- 1Layer has similar performance to the Default
- Compared with 1Layer, 2Layer delivers up to 84% and 184% improvement for latency and BW

## Inter-VM Inter-Container Pt2Pt (Inter-Socket)



- 1-Layer has similar performance to the Default
- 2-Layer has near-native performance for small msg, but clear overhead on large msg
- Compared to 2-Layer, Hybrid design brings up to 42% and 25% improvement for latency and BW, respectively

#### MUG 2017

### **Applications Performance**



- 256 processes across 64 containers on 16 nodes
- Compared with Default, enhanced-hybrid design reduces up to 16% (28,16) and 10% (LU) of execution time for Graph 500 and NAS, respectively
- Compared with the 1Layer case, enhanced-hybrid design also brings up to 12% (28,16) and 6% (LU) performance benefit.

# **Approaches to Build HPC Clouds**

- MVAPICH2-Virt with SR-IOV and IVSHMEM
  - Standalone, OpenStack
- SR-IOV-enabled VM Migration Support in MVAPICH2
- MVAPICH2 with Containers (Docker and Singularity)
- MVAPICH2 with Nested Virtualization (Container over VM)
- MVAPICH2-Virt on SLURM
  - SLURM alone, SLURM + OpenStack

# **Overview of OSU Solutions for Building HPC Clouds: MVAPICH2-Virt on SLURM and OpenStack**



Network Based Computing Laboratory



# **Need for Supporting SR-IOV and IVSHMEM in SLURM**

- Requirement of managing and isolating virtualized resources of SR-IOV and IVSHMEM
- Such kind of management and isolation is hard to be achieved by MPI library alone, but much easier with SLURM
- Efficient running MPI applications on HPC Clouds needs SLURM to support managing SR-IOV and IVSHMEM
  - Can critical HPC resources be efficiently shared among users by extending SLURM with support for SR-IOV and IVSHMEM based virtualization?
  - Can SR-IOV and IVSHMEM enabled SLURM and MPI library provide bare-metal performance for end applications on HPC Clouds?

# **Architecture Overview**



### **SLURM SPANK Plugin based Design**



VM Configuration Reader –

Register all VM configuration options, set in the job control environment so that they are visible to all allocated nodes.

 VM Launcher – Setup VMs on each allocated nodes.

- File based lock to detect occupied VF and exclusively allocate free VF

- Assign a unique ID to each IVSHMEM and dynamically attach to each VM

VM Reclaimer – Tear down
 VMs and reclaim resources

#### Network Based Computing Laboratory

# **SLURM SPANK Plugin with OpenStack based Design**



- VM Configuration Reader VM options register
- VM Launcher, VM Reclaimer Offload to underlying OpenStack infrastructure
  - PCI Whitelist to passthrough free VF to VM
  - Extend Nova to enable IVSHMEM when launching VM

J. Zhang, X. Lu, S. Chakraborty, D. K. Panda. SLURM-V: Extending SLURM for Building Efficient HPC Cloud with SR-IOV and IVShmem. Euro-Par, 2016

#### Network Based Computing Laboratory

## **Application-Level Performance on Chameleon (Graph500)**



- 32 VMs across 8 nodes, 6 Core/VM
- EASJ Compared to Native, less than 4% overhead with 128 Procs
- SACJ, EACJ Also minor overhead, when running NAS as concurrent job with 64 Procs

MUG 2017

### **Available Appliances on Chameleon Cloud\***

leoncloud.org/appliances/			e	Anglianaa	Description
tion Chameleon Daily KG G The default Chameleon appliance	CUDA appliance based on CentOS 7	Chameleon bare-metal image	Chameleon FPGA Runtime	Appliance	Description
CentOS 7 KVM SR-IOV	CentOS 7 SR-IOV MVAPICH2- Vit	customized with Docker to run containers.	COMPSs 1.3 CC-CentOS7	CentOS 7 KVM SR- IOV	Chameleon bare-metal image customized with the KVM hypervisor and a recompiled kernel to enable SR-IOV over InfiniBand. https://www.chameleoncloud.org/appliances/3/
Our chameteon bare-metal image customized with the KM hypervisor and a recompiled kernel to enable SR-IOV over Infiniband.	The CentOS 7 SR-IOV MVAPICH2-Virt appliance is built from the CentOS 7 KVM SR-IOV appliance and additionally contains MVAPICH2-Virt library	The CentOS 7 SR-IOV RDMA-Hadoop appliance is built from the CentOS 7 appliance and additionally contains RDMA-Hadoop library.	OMIPS is a tax based programming model for distributed platforms.	MPI bare-metal cluster complex appliance (Based on Heat)	This appliance deploys an MPI cluster composed of bare metal instances using the MVAPICH2 library over InfiniBand. https://www.chameleoncloud.org/appliances/29/
Hello World complex appliance A basic complex appliance deploying an NFS server with one client	MPI + SR-IOV KVM cluster MPI cluster of KVM virtual machines using the MVAPICI2-Virte library and SR-IOV enabled InfiniBand	MPI bare-metal cluster Bare-metal MPI cluster using the MVAPICH2 library over infiniBand.	MPI bare-metal cluster (MPICH3) Bare-metal MPI cluster using the MPICH3 implementation	MPI + SR-IOV KVM cluster (Based on Heat)	This appliance deploys an MPI cluster of KVM virtual machines using the MVAPICH2-Virt implementation and configured with SR-IOV for high-performance communication over InfiniBand. https://www.chameleoncloud.org/appliances/28/
NFS share An appliance deploying an NFS server with a configurable number of clients	OpenStack Mitaka (DevStack) OpenStack Mitaka with DevStack over one controller node and a configurable number of compute nodes	Ubuntu 14.04 Chameleon-supported Ubuntu 14.04 UTS image		CentOS 7 SR-IOV RDMA-Hadoop	The CentOS 7 SR-IOV RDMA-Hadoop appliance is built from the CentOS 7 appliance and additionally contains RDMA-Hadoop library with SR-IOV. https://www.chameleoncloud.org/appliances/17/

- Through these available appliances, users and researchers can easily deploy HPC clouds to perform experiments and run jobs with
  - High-Performance SR-IOV + InfiniBand
  - High-Performance MVAPICH2 Library over bare-metal InfiniBand clusters
  - High-Performance MVAPICH2 Library with Virtualization Support over SR-IOV enabled KVM clusters
  - High-Performance Hadoop with RDMA-based Enhancements Support

[\*] Only include appliances contributed by OSU NowLab

### **MPI Complex Appliances based on MVAPICH2 on Chameleon**



# **Demos on Chameleon Cloud**

- A Demo of Deploying MPI Bare-Metal Cluster with InfiniBand
- A Demo of Deploying MPI KVM Cluster with SR-IOV enabled InfiniBand
- Running MPI Programs on Chameleon

### **Login to Chameleon Cloud**



#### https://chi.tacc.chameleoncloud.org/dashboard/auth/login/

### **Create a Lease**

Chame	eleon	СН	I-816821 <del>*</del>						💄 zhanjie 🔻
Project	^	Le	ases					2	_
Compute	~							🛗 Lease Calendar 🕇	Create Lease Delete Leases
Network	~		Lease name	Start date	End date	Action	Status	Reason	Actions
Orchestration	~		zj-16-ib	2016-11-01 18:50 UTC	2016-11-07 23:50 UTC	START	COMPLETE	Successfully started lease	Update Lease 🔻
Object Store	~	0	sg-1	2016-10-25 18:54 UTC	2016-11-03 18:00 UTC	UPDATE	COMPLETE	Successfully updated lease	Update Lease 👻
Reservations	^	Displa	aying 2 items						
	Leases								
Identity	~								

### **Create a Lease**

Chame	leon	🔳 CH-816821 <del>-</del>	Create New Lease		×	🛔 zhanjie 👻
Project Compute	^ ~	Leases	Name *	Description:	Lease Calendar + Create Le	ase Delete Leases
Network	Ý	Lease n	Start Date * 😡	Time zone setting		Actions
Orchestration	~	🗌 zj-16-ib	Start Time (24 hour) * @	Your timezone is currently configured as <b>UTC</b> . If you need to update your timezone please go to your User Settings.	fully started lease	Update Lease 👻
Reservations	^	Displaying 2 items	End Date * <b>O</b>	Enter the start and end in your current time zone and they will be converted to UTC.	stully updated lease	Update Lease 👻
	Leases		yyyy-mm-dd End Time (24 hour) * •	For specific node reservations, you can find the node UUID using Resource Discovery on the user		
Identity	×		hh:mm Resource Type *	portal.		
			Physical Host			
			2			
			Maximum Number of Hosts * <b>@</b>			
			Reserve Specific Node <b>O</b>			
			Node Type to Reserve * 🚱			



Compute	~								Delete Lesses
Network	~		Lease name	Start date	End date	Action	Status	Reason	Actions
Orchestration	~		zj-16-ib	2016-11-01 18:50 UTC	2016-11-07 23:50 UTC	START	COMPLETE	Successfully started lease	Update Lease 👻
Object Store	~		sg-1	2016-10-25 18:54 UTC	2016-11-03 18:00 UTC	UPDATE	COMPLETE	Successfully updated lease	Update Lease 👻
Reservations	^	Displa	aying 2 items						
	Leases								
dentity	~								

## Select Complex Appliance - MPI Bare-Metal Cluster



Network Based Computing Laboratory

### **Get Template of MPI Bare-Metal Appliance**

#### Appliances / MPI bare-metal cluster



Author

Name: Network Based Computing Lab, The Ohio State University Contact: appliances@chameleoncloud.org

#### Save URL of Template (Will be used later) C https://www.chameleoncloud.org/appliances/api/appliances/29/template $\leftarrow$ Poeal MVAPICH2 Virtualization Chameleon Daily Google Scholar C Technology News -... heat template version: 2015-10-15 description: Bare-metal MPI cluster using the MVAPICH2 implementation parameters: key name: type: string label: Key name description: Name of a key pair to enable SSH access to the instance default: default constraints: - custom constraint: nova.kevpair reservation id: type: string description: ID of the Blazar reservation to use for launching instances constraints: - custom constraint: blazar.reservation node count: type: number label: Node count description: Number of physical nodes default: 1 constraints: - range: { min: 1 } description: There must be at least one physical node. resources: mpi keypair: type: OS::Nova::KeyPair properties: save\_private\_key: true name: str replace: template: mpi stack id params: stack\_id: { get\_param: "OS::stack\_id" } instance floating ip: type: OS::Nova::FloatingIP properties: pool: ext-net

#### Network Based Computing Laboratory



Char	<b>nele</b> on	CH-816821 -				💄 zhanjie 🔫
Project	^	Stacks			_	
Compute	~			Filter	+ Launch Stack	Preview Stack
Network	~	Stack Name	Created	Updated	Status	Actions
Orchestration	^		No ite	ms to display.		
	Stacks	Displaying 0 items				
	Resource Types					
Object Store	~					
Reservations	~					
Identity	~					

### **Use Saved Template URL as Source**

Chan	neleon	🔳 CH-816821 👻					🛔 zhanjie 👻
Project	^	Stacks	Select Template	^			
Compute	~		Template Source *	Description:	Q	+ Launch Stack	Preview Stack
Network	~	Stack N	Template URL Ø	Use one of the available template source options to specify the template to be used in creating this stack.	Status		Actions
Orchestration	^		eoncloud.org/appliances/api/appliances/29/template				
	Stacks	Displaying 0 items	Environment Source				
	Resource Types		File \$				
Object Store	~		Choose File No file chosen				
Reservations	~						
Identity	~			Cancel Next			

# **Input Stack Information**

Cha	meleon	🗂 CH-816821 👻					👗 zhanjie 👻
Durient		Stacks	Launch Stack	×			
Compute	~ ~	Oldone	Stack Name * 🛛	Description:	Q	+ Launch Stack	Preview Stack
Network	~	Stack N	Creation Timeout (minutes) * 🛛	Create a new stack with the provided values.	Status		Actions
Orchestration	<u> </u>		60				
	Stacks	Displaying 0 items	Rollback On Failure				
	Resource Types		Password for user "zhanjie" * 0				
Object Store	~		Key name 🛿				
Reservations	×		Select a key pair				
Identity	~		Node count @				
			reservation_id * 0				
			Select Reservation				
				Cancel Launch			

### **Use Created Lease**

Cha	meleon	🔲 CH-816821 👻						🌲 zhanjie 🔻
			Launch Stack		×			
Project	^	Stacks			-			
Compute	~		Stack Name * 🕢	Description:		Q	+ Launch Stack	Preview Stack
Network	~	0. 1.1	mpi-bare-metai	Create a new stack with the provided values.				
		Stack N	Creation Timeout (minutes) * @			Status		Actions
Orchestration	1 ^		60					
	Stacks	Displaying 0 items	□ Rollback On Failure <b>@</b>					
	Resource Types		Password for user "zhanjie" * 0					
Object Store	Ŷ		•••••					
			Key name 😧					
Reservations	~		jzmac 💠					
Identity	~		Node count 🕑					
			2					
			reservation_id *	1				
			zj-16-ib (4129fe81-0592-40ba-8a38-eeeff9655085) 🗘					
				4				
				Cancel Launc				

## **Stack Creation In Progress**

	СН	I-816821 <del>-</del>									🏝 zhanjie	•
Project ^	Sta	acks										
Compute ~				Filter C	٦ 4	Launch Stack	Preview Stack	Check Stacks	Suspend Stacks	Resume Stacks	× Delete Stack	ks
Network ~	Created			Created		Updated		Status			Actions	
Orchestration ^		mpi-bare-metal		0 minutes		Never		Create In Progress	6		Check Stack	•
Stacks	Displa	aying 1 item										
Resource Types												
Object Store ~												
Reservations ~												
Identity ~												

### **Stack Details**

Chameleon	CH-816821 - Azhanjie -
Project ^	Stack Details: mpi-bare-metal
Compute ~	Check Stack 🗸
Network ~	Topology Overview Resources Events Template
Orchestration ^ Stacks	mpi-bare-metal Create In Progress
Resource Types Object Store   Reservations   Identity	
# **Demos on Chameleon Cloud**

- A Demo of Deploying MPI Bare-Metal Cluster with InfiniBand
- A Demo of Deploying MPI KVM Cluster with SR-IOV enabled InfiniBand
- Running MPI Programs on Chameleon

### Select Complex Appliance - MPI KVM with SR-IOV Cluster



Network Based Computing Laboratory

#### Get Template of MPI KVM with SR-IOV Appliance

MPI + SR-IOV KVM cluster

unch Complex Appliance at CHI@UC Launch Complex Appliance at CHI@TACC

#### Description

This appliance deploys an MPI cluster of KVM virtual machines using the MVAPICH2-Virt implementation and configured with SR-IOV for high-performance communication over InfiniBand.

It accepts the following parameters:

- · key\_name: name of a key pair to enable SSH access to the instance (defaults to "default")
- reservation\_id: ID of the Blazar reservation to use for launching instances
- total\_nodes: Number of physical nodes to launch
- total\_vms: Number of virtual machines to create
- vcpu\_per\_vm: Number of VCPUs per virtual machine
- memory\_per\_vm: Amount of memory size per virtual machine (in GiB)

#### The following outputs are provided:

first\_instance\_ip: The public IP address of the first bare-metal instance. Login with the command 'ssh cc@first\_instance\_ip'.

To check the VM / IP mapping list, run the following command:

cat /home/cc/vm-ip\_mapping.dat

To run an MPI program, first login to one VM using command "ssh root@vm\_ip", then execute the following command, assuming you have compiled a program called mpi.out:

mpirun\_rsh -np <nprocs> -hostfile vmhosts MV2\_VIRT\_USE\_IVSHMEM=1 ./mpi.out

In some cases, the library path of the MVAPICH2-Virt package needs to be exported as follows before running MPI programs:

export LD\_LIBRARY\_PATH=/opt/mvapich2-virt/lib64:\$LD\_LIBRARY\_PATH

Refer to the MVAPICH2-Virt user guide for more details on running MPI programs.

Keywords

SR-IOV; MVAPICH2-Virt

Chameleon Supported

Template

& Get Template

#### Save URL of Template (Will be used later)

 $\leftarrow \rightarrow$ C https://www.chameleoncloud.org/appliances/api/appliances/28/template 🛅 MVAPICH2 🗎 Virtualization 📄 Chameleon 📄 Daily 🛐 Google Scholar 📵 Technology News -... Deal

heat template version: 2015-10-15

description: MPI cluster using KVM + SR-IOV and the MVAPICH2-Virt implementation

```
parameters:
 key name:
   type: string
   label: Kev name
   description: Name of a key pair to enable SSH access to the instance
   default: default
   constraints:
   - custom_constraint: nova.keypair
  reservation id:
   type: string
   description: ID of the Blazar reservation to use for launching instances
   constraints:
   - custom constraint: blazar.reservation
  total nodes:
   type: number
   description: Number of physical nodes to launch
   default: 1
   constraints:
     - range: { min: 1 }
       description: There must be at least one physical node.
  total vms:
   type: number
   description: Number of virtual machines to create
   default: 1
   constraints:
     - range: { min: 1 }
       description: There must be at least one virtual machine.
  vcpu_per_vm:
   type: number
   description: Number of VCPUs per virtual machine
   default: 2
   constraints:
     - range: { min: 1 }
       description: There must be at least one VCPU per virtual machine.
  memory per vm:
   type: number
   description: Amount of memory size per virtual machine (in GiB)
   default: 4
   constraints:
     - range: { min: 1 }
       description: There must be at least one VCPU per virtual machine.
```

☆ 🥝 向 🤇

#### Launch Stack

Cha	meleon	🔲 CH-816821 🕶	Launch Stack	\$	×	👗 zhanjie 😁
Project	^	Stacks	Stack Name * 🛛	Description:		
Compute	~		Creation Timeout (minutes) * 🛛	Create a new stack with the provided values.	spend Stacks	Resume Stacks × Delete Stacks
Network	~	Stack N	60			Actions
Orchestration	1 ^	D baremet	Rollback On Failure @			Check Stack 👻
	Stacks	Displaying 1 item	Password for user "zhanjie" * 🕢			
	Resource Types		۲			
Object Store	, v		Key name 🛛			
Object Store			Select a key pair 💠			
Reservations	×		memory_per_vm 😧			
Identity	~		4			
			reservation_id * 😡			
			Select Reservation			
			total_nodes 🚱			
			1			
			total_vms 0			
			1			
			vcpu_per_vm 😧			
			2			

# **Instances in Stack (Spawning ...)**

Chameleon	CH	I-816821 <del>•</del>											👗 zhanjie 👻
Project ^	Ins	stanc	es										
Compute ^					Insta	ince Nami \$	Filter	Filt	er 🔒 Laun	ch Instance	× Termina	te Instances	More Actions -
Overview	0	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions	
Images Access & Security	•	mpi- instance- 0	CC-CentOS7- SRIOV-MVAPICH2- Virt		baremetal	jzmac	Build	climate:4129fe81-0592- 40ba-8a38- eeeff9655085	Spawning	No State	2 minutes	Associate	Floating IP
Network ·		mpi- instance- 1	CC-CentOS7- SRIOV-MVAPICH2- Virt		baremetal	jzmac	Build	climate:4129fe81-0592- 40ba-8a38- eeeff9655085	Spawning	No State	2 minutes	Associate	Floating IP -
Object Store ~	•	node1	osu-swift	10.40.0.24 Floating IPs: 129.114.108.229	baremetal	shashank- bash	Active	climate:1c24edd4-6db2- 4003-ab0a- 2dcdf7055078	None	Running	6 days, 16 hours	Disassocia	te Floating IP 👻
Reservations ~	Displa	aying 3 items											
Identity ~													

# **Demos on Chameleon Cloud**

- A Demo of Deploying MPI Bare-Metal Cluster with InfiniBand
- A Demo of Deploying MPI KVM Cluster with SR-IOV enabled InfiniBand
- Running MPI Programs on Chameleon

# Create Stack Successfully Login to the first instance with Floating IP

	C C	H-816821 <del>*</del>										👗 zhanjie 👻
Project  A Instances												
Compute ^						Instance Nam \$ Filter			🗅 Lau	Launch Instance     X Terminate Instances     More Actions		
Overview		Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
Images Access & Security	0	mpi- instance- 0	CC-CentOS7-SRIOV- MVAPICH2-Virt	10.40.0.144 Floating IPs: 129.114.108.97	baremetal	jzmac	Active	climate:4129fe81-0592- 40ba-8a38-eeeff9655085	None	Running	41 minutes	Disassociate Floating IP
Network · Orchestration ·	0	mpi- instance- 1	CC-CentOS7-SRIOV- MVAPICH2-Virt	10.40.0.143	baremetal	jzmac	Active	climate:4129fe81-0592- 40ba-8a38-eeeff9655085	None	Running	41 minutes	Associate Floating IP
Object Store ~ Reservations ~	0	node1	osu-swift	10.40.0.24 Floating IPs: 129.114.108.229	baremetal	shashank- bash	Active	climate:1c24edd4-6db2- 4003-ab0a-2dcdf7055078	None	Running	6 days, 17 hours	Disassociate Floating IP
Identity ~	Dis	playing 3 items										

#### **Login Instance with Floating IP**

2. cc@mpi-instance-0:~ (ssh)
[Jie@MBA:~]\$ ssh cc@129.114.108.97
Last login: Thu Nov 3 08:09:10 2016 from 75-22-113-253.lightspeed.uparoh.sbcglobal.net
[cc@mpi-instance-0 ~]\$

#### **SSH to Other Instances**

● ● ● 2. cc@mpi-instance-1:~ (ssh)
[Jie@MBA:~]\$ ssh cc@129.114.108.97
Last login: Thu Nov 3 08:09:10 2016 from 75-22-113-253.lightspeed.uparoh.sbcglobal.net
[cc@mpi-instance-0 ~]\$ cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
10.40.0.143 mpi-instance-1
10.40.0.144 mpi-instance-0
[cc@mpi-instance-0 ~]\$ ssh mpi-instance-1
Last login: Thu Nov 3 08:03:47 2016 from mpi-instance-0
[cc@mpi-instance-1 ~]\$

### **Compile MPI Program – Hello World**

	2.	cc@mpi-instanc	e-0:~ (ssh)
#in	iclude <stdio.h></stdio.h>		<pre>[cc@mpi-instance-0 ~]\$ mpicc -o hello_world hello_world.c</pre>
#LT			hello_world
			[cc@mpi-instance-0 ~]\$
int	: main (int argc, char** argv)		
ł			
	int rank, size;		
	MPT Trit (Range Range):		
	MPT_INTE (adlyc, adlyv), MPT Comm rank (MPT COMM WORLD, &rank).		
	MPI_Comm_size (MPI_COMM_WORLD, &size):		
	printf( "Hello world from process %d of %d\n", rank	, size );	
	<pre>MPI_Finalize();</pre>		
	return 0;		
}			
2			
~			
~			

#### **Distribute Executable to Other Instances**

● ● ● 2. cc@mpi-i	nstance-0:~ (ssh)	
<pre>[cc@mpi-instance-0 ~]\$ scp hello_world mpi-instance-1:.</pre>	[cc@mpi-instance-1 ~]\$ ls hello_world	
hello_world 100% 12KB 12.1KB/s 00:00	hello_world	
<pre>[cc@mpi-instance-0 ~]\$ ls hello_world hello_world</pre>	[cc@mpi-instance-1 ~]\$	
[cc@mpi-instance-0 ~]\$		

#### **Run MPI Program – Hello World**



#### **Run OSU MPI Benchmarks – Latency and Bandwidth**

X root@mpi-instance	#1 × cc@mpi-instance-0: #2		
[cc@mpi-instar	ce-0 ~]\$ mpirun_rsh -np 2 -hostfile ho	ts /opt/mvapich2/l [cc@mpi-instance-0 ~]\$ mpirun_rsh -np 2 -hostfile hosts /op	t/mvapich2/
ibexec/osu-mic	ro-benchmarks/mpi/pt2pt/osu_latency	libexec/osu-micro-benchmarks/mpi/pt2pt/osu_bw	
# OSU MPI Late	ncy Test v5.3	# OSU MPI Bandwidth Test v5.3	
# Size	Latency (us)	<pre># Size Bandwidth (MB/s)</pre>	
3	1.26	1 3.50	
L	1.29	2 7.12	
2	1.30	4 14.59	
4	1.31	8 28.68	
8	1.30	16 57.70	
16	1.32	32 113.70	
32	1.34	64 225.90	
64	1.38	128 439.72	
128	1.47	256 844.32	
256	1.92	512 1628.04	
512	2.03	1024 2938.93	
1024	2.35	2048 4602.78	
2048	2.86	4096 5481.66	
4096	3.33	8192 5780.06	
8192	4.76	16384 5905.00	
16384	7.02	32768 5912.54	
32768	10.71	65536 6150.79	
65536	16.04	131072 6233.94	
131072	26.50	262144 6298.78	
262144	47.28	524288 6319.60	
524288	89.24	1048576 6321.27	
1048576	172.01	2097152 6268.38	
2097152	340.85	4194304 6264.26	
4194304	677.87	[cc@mpi-instance-0 ~]\$	
[cc@mpi-instar	ce-0 ~]\$		

# **Next Steps of MVAPICH2-Virt**



**Network Based Computing Laboratory** 

MUG 2017

#### Conclusions

- MVAPICH2-Virt over SR-IOV-enabled InfiniBand is an efficient approach to build HPC Clouds
  - Standalone, OpenStack, Slurm, and Slurm + OpenStack
  - Support Virtual Machine Migration with SR-IOV InfiniBand devices
  - Support Virtual Machine, Container (Docker and Singularity), and Nested Virtualization
- Very little overhead with virtualization, near native performance at application level
- Much better performance than Amazon EC2
- MVAPICH2-Virt is available for building HPC Clouds
  - SR-IOV, IVSHMEM, Docker and Singularity, OpenStack
- Appliances for MVAPICH2-Virt are available for building HPC Clouds
- Demos on NSF Chameleon Cloud
- Future releases for supporting running MPI jobs in VMs/Containers with SLURM, etc.
- SR-IOV/container support and appliances for other MVAPICH2 libraries (MVAPICH2-X, MVAPICH2-GDR, ...)

# **Thank You!**

luxi@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~luxi





#### Network-Based Computing Laboratory

http://nowlab.cse.ohio-state.edu/

MVAPICH: MPI over InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE <a href="http://mvapich.cse.ohio-state.edu/">http://mvapich.cse.ohio-state.edu/</a>