Designing and Building Efficient HPC Cloud with Modern Networking Technologies on Heterogeneous HPC Clusters

Jie Zhang
Dr. Dhabaleswar K. Panda (Advisor)

Department of Computer Science & Engineering
The Ohio State University
Outline

• Introduction
• Problem Statement
• Detailed Designs and Results
• Impact on HPC Community
• Conclusion
Cloud Computing and Virtualization

- Cloud Computing focuses on maximizing the effectiveness of the shared resources
- Virtualization is the key technology behind
- Widely adopted in industry computing environment
- IDC Forecasts Worldwide Public IT Cloud Services spending will reach $195 billion by 2020

(Courtesy: http://www.idc.com/getdoc.jsp?containerId=prUS41669516)
Drivers of Modern HPC Cluster and Cloud Architecture

- Multi-/Many-core technologies
- Accelerators (GPUs/Co-processors)
- Large memory nodes
- Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
- Single Root I/O Virtualization (SR-IOV)
Single Root I/O Virtualization (SR-IOV)

- 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

Single Root I/O Virtualization (SR-IOV) is providing new opportunities to design HPC cloud with very little low overhead through bypassing hypervisor
Does it suffice to build efficient HPC cloud with only SR-IOV?

NO.

- Not support locality-aware communication, co-located VMs still has to use SR-IOV channel
- Not support VM migration because of device passthrough
- Not properly manage and isolate critical virtualized resource
Problem Statements

• Can MPI runtime be redesigned to provide virtualization support for VMs/Containers when building HPC clouds?
• How much benefits can be achieved on HPC clouds with redesigned MPI runtime for scientific kernels and applications?
• Can fault-tolerance/resilience (Live Migration) be supported on SR-IOV enabled HPC clouds?
• Can we co-design with resource management and scheduling systems to enable HPC clouds on modern HPC systems?
Research Framework

HPC Applications

MPI Runtime
- Locality Aware Communication
- NUMA Aware Communication
- Migration Support
- GPUs/Coprocessors Support
- SMP/IVShmem Channel
- CMA Channel
- SR-IOV Channel
- Accelerator Channel

HPC Clouds (VM, Container, Nested Cloud)
- SLURM
  - SPAN
- OpenStack
  - Nova
  - Heat
  - Glance
  - Neutron
- Parallel Migration Controller

Modern HPC System Architecture
- Multi-core Processor
- SR-IOV enabled InfiniBand Network
- Omni-Path Network
- GPUs & Coprocessors
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 2,825 organizations in 85 countries
  - More than 432,000 (> 0.4 million) downloads from the OSU site directly
  - Empowering many TOP500 clusters (Jul ‘17 ranking)
    - 1st ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 15th ranked 241,108-core cluster (Pleiades) at NASA
    - 20th ranked 522,080-core cluster (Stampede) at TACC
    - 44th ranked 74,520-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)
  - http://mvapich.cse.ohio-state.edu
Locality-aware MPI Communication with SR-IOV and IVShmem

- MPI library running in native and virtualization environments
- In virtualized environment
  - Support shared-memory channels (SMP, IVShmem) and SR-IOV channel
  - Locality detection
  - Communication coordination
  - Communication optimizations on different channels (SMP, IVShmem, SR-IOV; RC, UD)

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
SR-IOV-enabled VM Migration Support on HPC Clouds

```
[root@sandy1:migration]$ ssh sandy3-vm1 lspci
root@sandy3-vm1's password:
00:00.0 Host bridge: Intel Corporation 440FX - 8244FX PME [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 [D0FF]: Red Hat, Inc Virtio memory balloon
[root@sandy1:migration]$
[root@sandy1:migration]$ srirch migrate --live --rdma-pin-all --migrateuri rdma://sandy3-ib sandy1-vm1 qemu://sandy3-ib/system
error: Requested operation is not valid: domain has assigned non-USB host devices
[root@sandy1:migration]$ 
```
High Performance SR-IOV enabled VM Migration Framework for MPI Applications

Two Challenges
1. Detach/re-attach virtualized devices
2. Maintain IB Connection

Challenge 1: Multiple parallel libraries to coordinate with VM during migration (detach/reattach SR-IOV/IVShmem, migrate VMs, migration status)

Challenge 2: 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

Proposed Design of MPI Runtime

No-migration

Progress Engine Based

Migration-thread based
Typical Scenario

Migration-thread based
Worst Scenario

Network Based Computing Laboratory  SC 2017 Doctoral Showcase
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
High Performance MPI Communication for Nested Virtualization

Two-Layer Locality Detector:
Dynamically detecting MPI processes in the co-resident 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 and D. K. Panda, Designing Locality and NUMA Aware MPI Runtime for Nested Virtualization based HPC Cloud with SR-IOV Enabled InfiniBand, The 13th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE ’17), April 2017
Two-Layer NUMA Aware Communication Coordinator

- **Nested Locality Loader** reads locality info of destination process from Two-Layer Locality Detector
- **NUMA Loader** reads info of VM/container placements to decide on which NUMA node the destination process is pinning
- **Message Parser** obtains message attributes, e.g., message type and message size
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.
Typical Usage Scenarios

Exclusive Allocations
Sequential Jobs (EASJ)

Exclusive Allocations
Concurrent Jobs (EACJ)

Shared-host Allocations
Concurrent Jobs (SACJ)
Slurm-V Architecture Overview

1. SR-IOV virtual function
2. IVSHMEM device
3. Network setting
4. Image management
5. Launching VMs and check availability
6. Mount global storage, etc.
### Alternative Designs of Slurm-V

- **Slurm SPANK Plugin based design**
  - Utilize SPANK plugin to read VM configuration, launch/reclaim VM
  - File based lock to detect occupied VF and exclusively allocate free VF
  - Assign a unique ID to each IVSHMEM device and dynamically attach to each VM
  - Inherit advantages from Slurm: coordination, scalability, security

- **Slurm SPANK Plugin over OpenStack based design**
  - Offload VM launch/reclaim to underlying OpenStack framework
  - PCI Whitelist to passthrough free VF to VM
  - Extend Nova to enable IVSHMEM when launching VM
  - Inherit advantage from both OpenStack and Slurm: component optimization, performance
Applications Performance

Graph500 with 64 Procs across 8 Nodes on Chameleon

- 32 VMs across 8 nodes, 6 Cores/VM
- EASJ - Compared to Native, less than 4% overhead
- SACJ, EACJ – less than 9% overhead, when running NAS as concurrent job with 64 Procs
Impact on HPC and Cloud Communities

• Designs available through MVAPICH2-Virt library http://mvapich.cse.ohio-state.edu/download/mvapich/virt/mvapich2-virt-2.2-1.el7.centos.x86_64.rpm

• Complex Appliances available on Chameleon Cloud
  – MPI bare-metal cluster: https://www.chameleoncloud.org/appliances/29/
  – MPI + SR-IOV KVM cluster: https://www.chameleoncloud.org/appliances/28/

• Enables users to easily and quickly deploy HPC clouds and perform jobs with high performance

• Enables administrators to efficiently manage and schedule cluster resource
Conclusion

• Addresses key issues on building efficient HPC clouds
• Optimizes MPI communication on various HPC clouds
• Presents designs of live migration to provide fault-tolerance on HPC clouds
• Presents co-designs with resource management and scheduling systems
• Demonstrates the corresponding benefits on modern HPC clusters
• Broader outreach through MVAPICH2-Virt public releases and complex appliances on Chameleon Cloud testbed
Thank You! & Questions?

zhang.2794@osu.edu

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

MVAPICH Web Page
http://mvapich.cse.ohio-state.edu/