Supporting High Performance Molecular Dynamics in Virtualized Clusters using IOMMU, SR-IOV, and GPUDirect

Andrew J. Younge
School of Informatics & Computing
Indiana University
Bloomington, IN 47408
Email: ajyounge@indiana.edu

John Paul Walters
Information Sciences Institute
University of Southern California
Arlington, VA 22203
Email: jwalters@isi.edu

Geoffrey C. Fox
School of Informatics & Computing
Indiana University
Bloomington, IN 47408
Email: gcf@indiana.edu

***Abstract*—Cloud infrastructure-as-a-Service paradigms have recently shown their utility for a vast array of computational problems, ranging from advanced web service architecutres to high throughput computing. However, many scientific computing applications have been slow to adapt to virtualized cloud frameworks. This is due to performance impacts of virtualization technologies, coupled with the lack of advanced hardware support necessary for running many high performance scientific appications at scale.**

By using KVM virtual machines that leverage both Nvidia GPUs and InfiniBand, we show that molecular dynamics simulations with LAMMPs and HOOMD run at near-native speeds. This experiment also illustrates how virtualized environments can support the latest parallel computing paradigms, including both MPI+CUDA and new GPUDirect RDMA functionality. Specific findings show initial promise in scaling of such applications to larger production deployments targeting large scale computational workloads.

I. Introduction

At present we stand at the inevitable intersection between High Performance Computing (HPC) and clouds. Various platform tools such as Hadoop and MapReduce, among others, have already percolated into data intensive computing within HPC [**?**]. In addition, there are efforts to support traditional HPC-centric scientific computing applications in virtualized cloud infrastructure. There are a multitude of reasons for supporting parallel computation in the cloud [**?**], including features such as dynamic scalability, specialized operating environments, simple management interfaces, fault tolerance, and enhanced quality of service, to name a few. The growing importance of supporting advanced scientific computing using cloud infrastructure can be seen by a variety of new efforts, including the NSF-funded XSEDE Comet resource at SDSC [**?**].

Nevertheless, there exists a past notion that virtualization used in today's cloud infrastructure is inherently inefficient. Historically, cloud infrastructure has also done little to provide the necessary advanced hardware capabilities that have become almost mandatory in supercomputers today, most notably advanced GPUs and high-speed, low-latency interconnects. The result of these notions has hindered the use of virtualized environments for parallel computation, where performance must be paramount.

A growing effort is currently underway that looks to systematically identify and reduce any overhead in virtualization technologies, so far with relative success [**?**], [**?**]. Thus, we see a consitantly diminishing overhead with virtualization, not only with traditional overheads [] but also with HPC workloads. While virtualization will almost always include some additionaly overhead in relation to its dynamic abilities, the eventual goal for supporting HPC in virtualized environments is to minimize any overhead wheneve rpossible.

To advance the placement of HPC applications on virtual machines, new efforts are emerging foscusing specificaly on key HPC hardware. By leveraging new virtualization tools such as IOMMU device passthrough and SR-IOV, we can now support the same common HPC hardware such as the latest Nvidia Tesla GPUs [**?**] as well as InfiniBand fabric [**?**].

Recent advances in hypervisor performance [**?**] coupled with the newfound availably of HPC hardware in virtual machines analogous to the most powerful supercomputers used today, we see can see the formation of a high performance cloud infrastructure. While our previous advances in this area have focused on single-node advancements, it is now imperative to ensure real-world applications can also operate at scale.

To start, we demonstrate running two molecular dynamics simulations, LAMMPS and HOOMD, in a virtual infrastructure complete with both Kepler GPUs and QDR InfiniBand. Both LHOOMD and LAMMPS are used extensively in some of the world's fastest supercomputers and represent a key simulation example that HPC supports today. We show that these applications are able to run at near-native speeds within a completely virtualized environment, demonstrating small performance impacts usually acceptable by many users. Furthermore, we illustrate the ability of such a virtualized environment to support cuttinge edge software tools such as RDMA GPUDirect, demonstrating cutting-edge technologies are indeed possible in a virtualized environment.

Following these efforts, we hope to ensure upstream infrastructure projects such as OpenStack [?] are able to make effective and quick use of these features, allowing users to build private cloud infrastructure to support high performance distributed computational workloads.

II. Background and Related Work

NOTE: first introduce virtualization, and I/O featuresets. Then enable GPUs. Then bring in SR-IOv and infiniBand. Finally, discuss applications and GPUDirect.

Virtualization technologies and hypervisors have been seen widespread deployment in support of a vast array of applications. This ranges from public commercial Cloud deployments such as Amazon EC2 [?], Microsoft Azure, and Google's Cloud Platform [?] to private deployments within colocation facilities, corporate data centers, and even national scale cyberinfrastructure initiatives. All these support look to support various use cases and applications such as web servers, ACID and BASE databases, online object storage, and even distributed systems, to name a few.

The use of virtualization and hypervisors specifically support various HPC solutions has been studied with mixed results. In [?], it is found that there is a great deal of variance between hypervisors when running various distributed memory and MPI applications, finding that KVM overall performed well across an array of HPC benchmarks. Furthermore, some applications may not may well into default virtualized evnironments, such as High Performance Linpack [?].

Recently, various CPU architectures have added support for I/O virtualization mechanisms directly in the CPU ISA through the use of an I/O memory management unit (IOMMU). Often, this is referred to as PCI Passthrough, as it enabled devices on the PCI-Express bus to be passed directly to a specific virtual machine (VM). Specific hardware implementations include Intel's VT-d [?], AMD's IOMMU [?] from x86_64 architectures, and even more recently ARM System MMU [?]. All of these implementations effectively look to aid in the usage of DMA-capable hardware to be used within a specific virtual machine. Using these features, a wide array of hardware can be utilized directly within VMs and enable fast and efficient computation and I/O capabilities.

With PCI Passthrough, a PCI device is handed directly to a running (or booting) thereby relinquishing complete control of the device within the hostentirely. This is different form typical VM usage where hardware is emulated in the host and used in a guest VM, such as with bridged ethernet adapters or emulated VGA devices. Performing PCI Passthrough requires the host to seize the device upon boot using a specialized driver to effectively block normal driver initialization. In the instance of the KVM hypervisor, this is done using the *vfio and* pci_stub drivers. Then, this driver relinquishes control to the VM, whereby normal device drivers initiate the hardware and enable the device for use within a given VM.

A. GPU Passthrough

Nvidia GPUs comprise the single most common accelerator in the Nov 2014 Top 500 List [?] and represent an increasing shift towards accelerators for HPC applications. Recently efforts have been seen to support such GPU accelerators directly within VMs using IOMMU technologies, with implementations now available with KVM [?], Xen [?] and VMWare [?]. These efforts have shown that GPUs can achieve up to 99% of their bare metal performance when passed to a virtual machine using PCI passthrough [?]. These works demonstrate PCI passthrough performance across a range of hypervisors and GPUs, but were limited to single node performance.

B. SR-IOV and InfiniBand

With almost all parallel HPC applications, the interconnect fabric necessary to enable efficient communication between processors becomes a central requirement to acheiving good performance. Specifically, a high bandwidth link is needed for distruted processors to share large amounts of data. Futhermore, low latency becomes equally important for speeding small message communications and resolving large barriers within parallelized code. One such interconnect, InfiniBand, has become the most common implementation used within the Top500 list. Previously, InfiniBand was inaccessable to virtualized environments.

Supporting I/O interconnects in VMs has been aided by Single Root I/O Virtualization (SR-IOV), whereby multiple virtual PCI functions are created in hardware to represent a single PCI device. These virtual functions (VFs) can then be passed to a VM and used as if it had direct access to that PCI device. SR-IOV allows for the virtualization and multiplexing to be done within the hardware, allowing for higher performance and greater control.

SR-IOV has been used in conjunction with Ethernet etensively to provide high performance 10Gb TCP/IP connectivity within VMs [?], offering near-native bandwidth and advanced QoS features not easily obtained through emulated Ethernet offerings. Currently Amazon EC2 offers a high performance VM solution utilizing SR-IOV enabled 10Gb Ethernet adapters. However, Ethernet dos not offer the high bandwidth or low latency typically found with InfiniBand.

Recently SR-IOV support for InfiniBand has been added by Mellanox in the ConnectX series adapters. Initial evaluation of SR-IOV InfiniBand within KVM VMs has proven has found point-to-point bandwidth to be near-native, but up to 30% latency overhead latency for small messages [?], [?], [?]. However, even with the noted overhead, this still signifies up to an order of magnitude difference in latency between InfiniBand and Ethernet with VMs. Furthermore, advanced configuration of SR-IOV enabled Infiniband fabric has taken shape, with recent research showing up to a 30% reduction latency [?]. However, real application performance has not yet been well understood.

III. Application and Experimental Setup

Using two molecular dynamics tools, LAMMPS [?] and HOOMD [?], we demonstrate a high performance *system*. That is, we combine PCI passthrough for Nvidia Kepler-class GPUs with QDR Infiniband SR-IOV and show that high performance molecular dynamics simulations are achievable within a virtualized environment.

For the first time, we also demonstrate Nvidia GPUDirect technology within such a virtual environment. Thus, we look to not only illustrate that virtual machines provide a flexible high performance infrastructure for scaling scientific workloads including MD simulations, but also that the latest HPC features and programming environments are also available in this same model.

A. Node configuration

To support the use of Nvidia GPUs and InfiniBand within a VM, specific and exact configuration is needed. This node configuration is illustrated in figure 1, and while our implementation is specific to KVM, this setup represents a design that can be hypervisor agnostic.

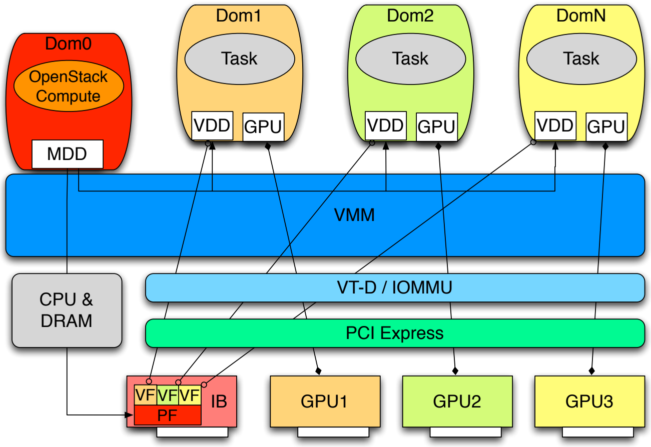

Fig. 1. LAMMPS RHODO & LJ Performance

A Kepler GPU is passed through using PCI Passthrough and directly initiated within the VM. While this specific implementation used only a single GPU, it is also possible to include as many GPUs as one can fit within the PCI Express bus if desired. As the GPU is used by the VM, an onboard VGA device was used by the host and a standard Cirris VGA was emulated in the guest OS.

CentOS 6.4 with a 3.13 upstream Linux kernel was used as the host OS with the KVM hypervisor. The native bare-metal base system and all guest VMs are composed of a CentOS 6.4 installation with a 2.6.32-358.23.2 stock kernel, MVAPICH 2.0 GDR, and CUDA version 5.5. Each guest was allocated 20 GB of RAM and a full socket (8 cores) as well as a single InfiniBand virtual function and 1 Kepler GPU per VM.

B. Cluster Configuration

Our test environment is composed of 4 servers each with a single Nvidia Kepler-class GPU. Two servers are equipped with K20 GPUs, while the other two servers are equipped with K40 GPUs. Each server is composed of 2 Intel Xeon E5-2670 CPUs, 48GB of DDR3 memory, and Mellanox ConnectX-3 QDR Infiniband. CPU sockets and memory are split evenly between two NUMA nodes. For these experiments, both the GPUs and Infiniband adapters are attached to NUMA node 1 and both the guest VMs and the base system utilized identical software stacks.

In order to effectively test MD simulations in LAMMPS and HOOMD beyond single-node tests, ronment. Bespin includes 4 blades, each with 2 Intel Xeron E5-2670 CPUs, 48Gb DDR3 memory, Mellanox ConnectX3 QDR InfiniBand cards, and a mixture of Nvidia Kepler series K20 and K40 GPUs. While the effective experimental hardware allocation remains relatively low compared to production runs of either application, it does allow for a useful evaluation at a larger scale than preivously evaluated as well as a valued extrapolation to larger resources.

C. Molecular Dynamics Applications

JP to Write

IV. Results

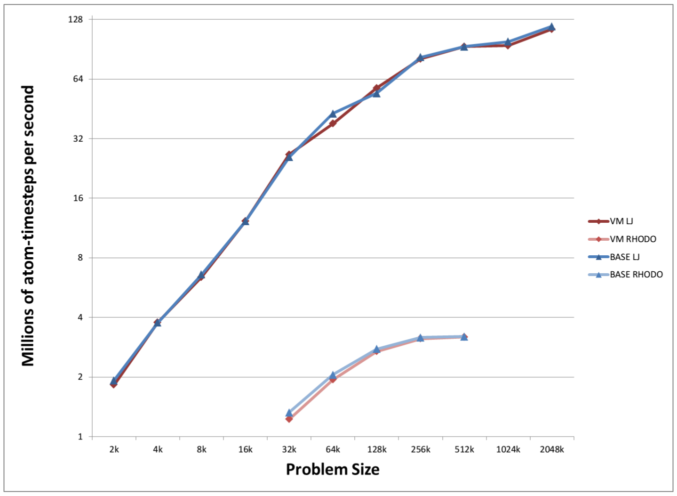

Fig. 2. LAMMPS RHODO & LJ Performance

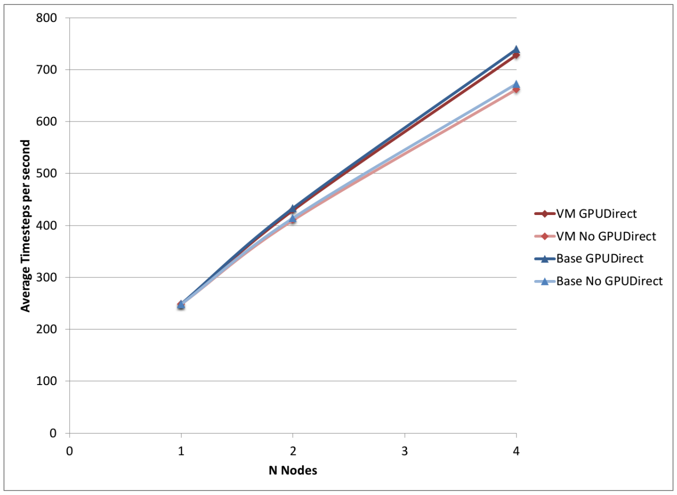

Fig. 3. HOOMD LJ Performance with 256k Simulation

Figure 2 shows two of the most common LAMMPS algorithms used; the Lennard-Jones potential (LJ) and the Rhodopsin protein in solvated lipid bilayer benchmark (Rhodo), both running with the GPU package across 8 cores per GPU. Here we see that both benchmarks scale remarkably well in the virtualized KVM guest environment. Compared to the base system performance, the VMs running LAMMPs acheive 96.7% and 99.3% efficiency for LJ and Rhodo, respectively when running across all nodes. These low overheads illustrate the utility of running LAMPS on cloud infrastructure, and also hold promise for other hybrid MPI + CUDA applications to also scale well in a virtualized environment.

In Figure 3 we show the performance of a Lennard-Jones liquid simulation with 256K particles running under HOOMD. HOOMD includes support for CUDA-aware MPI implementations via GPUDirect. The MVAPICH 2.0 GDR implementation enables a further optimization by supporting RDMA for GPUDirect. From Figure 3 we can see that HOOMD simulations, both with and without GPUDirect, perform very near-native. The GPUDirect results at 4 nodes achieve 98.5% of the base system's performance. The non-GPUDirect results achieve 98.4% efficiency at 4 nodes. Further, GPUDirect RDMA shows a clear advantage over the non-GPUDirect implementation, achieving a 9% performance boost in both the native a virtualized experiments.

V. Conclusion

With the advent of cloud infrastructure, the ability to run large-scale parallel scientific applications has become possible but limited due to both performance and hardware availability issues. In this work we show that advanced HPC-oriented hardware such as the latest Nvidia GPUs and InfiniBand fabric are now available within a virtualized infrastructure. Our results find MPI + CUDA applications run at near-native performance compared to traditional non-virtualized HPC infrastructure, with just an averaged 1.9% and 1.5% overhead for LAMMPs and HOOMD, respectively. Moving forward, we show the utility of GPUDirect RDMA for the first time in a cloud environment with HOOMD. Effectively, we look to pave the way for large-scale virtualized cloud Infrastructure to support a wide array of advanced scientific computation commonly found running on many supercomputers today.

References

[1] S. Jha, J. Qiu, A. Luckow, P. K. Mantha, and G. C. Fox, "A tale of two data-intensive paradigms: Applications, abstractions, and architectures," in *Proceedings of the 3rd International Congress on Big Data*, 2014.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[3] M. Norman and R. Moore, "NSF awards 12 million to sdsc to deploy comet supercomputer," Web page, 2013.

[4] A. J. Younge, R. Henschel, J. T. Brown, G. von Laszewski, J. Qiu, and G. C. Fox, "Analysis of Virtualization Technologies for High Performance Computing Environments," in *Proceedings of the 4th International Conference on Cloud Computing (CLOUD 2011)*. Washington, DC: IEEE, 2011.

[5] J. P. Walters, A. J. Younge, D.-I. Kang, K.-T. Yao, M. Kang, S. P. Crago, and G. C. Fox, "GPU-Passthrough Performance: A Comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL Applications," in *Proceedings of the 7th IEEE International Conference on Cloud Computing (CLOUD 2014)*. Anchorage, AK: IEEE, 2014.

[6] L. Vu, H. Sivaraman, and R. Bidarkar, "Gpu virtualization for high performance general purpose computing on the esx hypervisor," in *Proceedings of the High Performance Computing Symposium*, ser. HPC '14. San Diego, CA, USA: Society for Computer Simulation International, 2014, pp. 2:1–2:8. [Online]. Available: http://dl.acm.org/citation.cfm?id=2663510.2663512

[7] J. Jose, M. Li, X. Lu, K. C. Kandalla, M. D. Arnold, and D. K. Panda, "SR-IOV support for virtualization on infiniband clusters: Early experience," in *Cluster Computing and the Grid, IEEE International Symposium on*, 2013, pp. 385–392.

[8] S. Plimpton, P. Crozier, and A. Thompson, "Lammps-large-scale atomic/molecular massively parallel simulator," *Sandia National Laboratories*, 2007.

[9] J. Anderson, A. Keys, C. Phillips, T. Dac Nguyen, and S. Glotzer, "Hoomd-blue, general-purpose many-body dynamics on the gpu," in *APS Meeting Abstracts*, vol. 1, 2010, p. 18008.