Advanced Virtualization Techniques for High Performance Cloud Cyberinfrastructure

Andrew J. Younge, Geoffrey C. Fox
Pervasive Technology Institute, Indiana University
2719 E 10th St., Bloomington, IN 47408, U.S.A.
Email corresponding author: ajyounge@indiana.edu

***Abstract*—With the advent of virtualization and Infrastructure-as-a-Service (IaaS), the broader scientific computing community is considering the use of clouds for their scientific computing needs. This is due to the relative scalability, ease of use, advanced user environment customization abilities, and the many novel computing paradigms available for data-intensive applications. However, there is still a notable gap that exists between the performance of IaaS when compared to typical high performance computing (HPC) resources, limiting the applicability of IaaS for many potential users.**

This work proposes to bridge the gap between supercomputing and clouds using a few key aspects. First, we evaluate current hypervisors and their viability to run HPC workloads within current infrastructure. Next, we illustrate a mechanism to enable advanced accelerators such as GPUs in a Virtual Machine that can significantly enhance scientific computing problems. Furthermore, we are also able to support high speed, low latency inter-node communication through the use of InfiniBand within virtual machines. Upon evaluating these newfound features and leveraging the system within the OpenStack environment, we illustrate that cloud computing can perform at near-native speeds and support a broad range of scientific computing problems as never before.

I. Introduction

Many industry leaders have focused on leveraging the economies of scale from data center operations and advanced virtualization technologies to service two classes of problems: handling millions of user interactions concurrently or organizing, cataloging, and retrieving mountains of data in short order. The result of these efforts has led to the advent of cloud computing, which leverages data center operations, virtualization, and a unified and user-friendly interface to interact with computational resources. This is culminated in the *aaS mentality, where everything is delivered as a service. This model treats both data and compute resources as a commodity and places an immediate and well defined value on the cost of using large resources [1]. Users are able to scale their needs from a single small compute instance to thousands or more instances aggregated together in a single data center. Clouds also provide access to complex parallel resources through simple interfaces, which have enabled a new class of internet applications and tools ranging from social networking to cataloging the world's knowledge.

Scientific computing endeavours have created clusters, grids, and supercomputers as high performance computing (HPC) platforms and paradigms, which are capable of tackling non-trivial parallel problems. HPC resources continually strive for the best possible computational performance on the cusp of Moore's Law. This pursuit can be seen through the focus on cutting edge architectures, high-speed, low-latency interconnects, parallel languages, and dynamic libraries, all tuned to maximize computational efficiency. Performance has been the keystone of HPC since its conception, with many ways to evaluate performance.

The Top500 supercomputer ranking system, which has been running for over 20 years, is a hallmark to performance's importance in supercomputing. When characterizing many of the fastest computers on the Top500 list, we see a few trends emerge. The first immediate trend is the use of dense many-core systems, most commonly using commodity x86 CPUs and memory. Next, it is clear there is a large dependency for a high speed interconnect for any distributed memory supercomputer, often in the form of QDR or FDR InfiniBand fabrics using Mellanox adapters. Recently, there has also been a wave of accelerators, most notably Nvidia Tesla GPUs, that can increase computation of a given node by as much as two orders of magnitude in some cases. While these trends do not represent strict rules for supercomputers, they do illustrate that scientific computing advances when given similar hardware characteristics.

While cloud computing has taken hold within industry, many scientists and researchers are reluctant to leverage the power of clouds. This is largely due to perceived performance drawbacks and feature limitations in executing many scientific computing applications across virtual machines. As such, it comes evident that a new infrastructure is needed that provides the ease of use and ubiquity of cloud computing with the advanced performance and hardware characteristics found in high performance computing.

Providing a true HPC Cloud requires a multifaceted approach. First, we look to investigate the performance available in IaaS technologies today, including the overhead that exists with various hypervisors solutions, and discuss optimizations and best-practices for running fast and efficient virtual machines. Next, we look to leverage specialized hardware traditionally only available in HPC, such as and advanced GPU accelerators and high speed interconnects. Then, we provide these advances within an OpenStack cloud IaaS framework and evaluate its utility for scientific computing. Finally, we illustrate the value of a high performance cloud using both synthetic HPC benchmarks and real-world applications. In our work, we use the FutureGrid project [2], a distributed grid

and cloud testbed, as it provides an ideal test-bed to build an experimental IaaS based on real-world resources, eliminating the need for simulation or emulation.

II. Hypervisor Performance

As with the HPC industry, performance must be a first class function of any cloud architecture for scientific computing. From a computational perspective, the amount of overhead introduced by virtualization needs to be minimized or eliminated entirely. This applies to typical floating-point operations common in MPI applications and represented in the Top500 list via two well known industry standard performance benchmark suites; HPCC and SPEC [3]. These two benchmark environments are recognized for their standardized reproducible results in the HPC community and provide a means to stress and compare processor, memory, inter-process communication, and overall performance and throughput of a system.

In Figure 1 from [4], we can see the comparison of Xen, KVM, and Virtual Box compared to native bare-metal performance of a single physical machine. First, we see that native is capable of around 73.5 Gflops which, with no optimizations, achieves 75% of the theoretical peak performance. Xen, KVM and VirtualBox perform at 49.1, 51.8 and 51.3 Gflops, respectively when averaged over 20 runs. However Xen, unlike KVM and VirtualBox, has a high degree of variance between runs. This is an interesting phenomenon for two reasons. First, this may impact performance metrics for other HPC applications and cause errors and delays between even pleasingly-parallel applications and add to reducer function delays. Second, this wide variance breaks a key component of Cloud computing providing a specific and predefined quality of service. If performance can sway as widely as what occurred for Linpack, then this may have a negative impact on users.

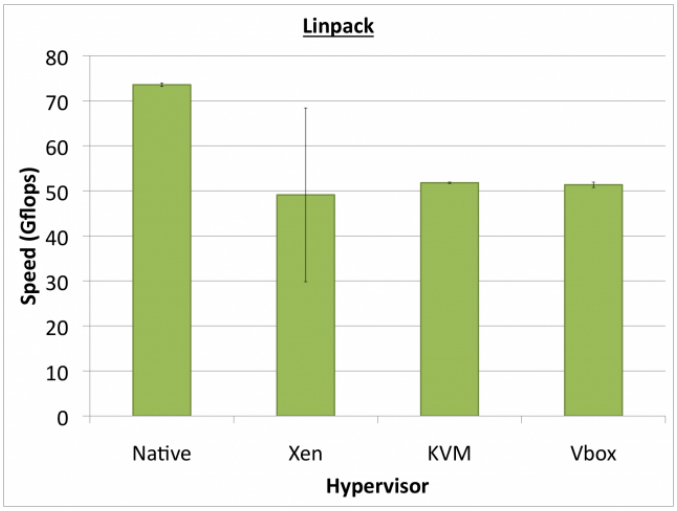

Fig. 1. Linpack performance

Next, we turn to another key benchmark within the HPC community, Fast Fourier Transforms (FFT). Unlike the synthetic Linpack benchmark, FFT is a specific, purposeful benchmark which provides results which are often regarded as more relative to a user's real-world application than HPL. From Figure 2, we can see rather distinct results from what was previously provided by HPL. Looking at Star and Single FFT, its clear performance across all hypervisors is roughly equal to bare-metal performance, a good indication that HPC applications may be well suited for use on VMs. The results for MPI FFT also show similar results, with the exception of Xen, which has a decreased performance and high variance as seen in the HPL benchmark. Our current hypothesis is that there is an adverse affect of using Intel's MPI runtime on Xen, however the investigation is still ongoing.

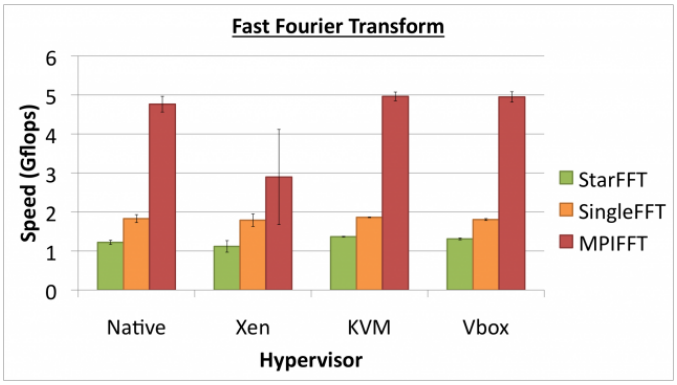

Fig. 2. Fast Fourier Transform performance

While Xen is historically regarded as the most widely used hypervisor, especially within academic clouds and grids, it's performance lags when compared to either KVM or VirtualBox. In particular, Xen's wide and unexplained fluctuations in performance throughout the series of benchmarks suggests that Xen may not be the best choice for scientific computing applications which benefit most form consistent for distributed memory applications. KVM rates the best across all performance benchmarks, making it the optimal choice for *general* deployment in an HPC environment. Furthermore, this work's illustration of the variance in performance among each benchmark and the applicability of each benchmark towards new applications may make possible the ability to preemptively classify applications for accurate prediction towards the ideal virtualized Cloud environment.

In reviewing the results, we find KVM is the best overall choice for use within HPC Cloud environments. KVM's feature-rich experience and near-native performance makes it a natural fit for deployment in an environment where usability and performance are paramount. Within the FutureGrid project specifically, we hope to deploy the KVM hypervisor across our Cloud platforms in the near future, as it offers clear benefits over the current Xen deployment. Furthermore, we expect these findings to be of great importance to other public and private Cloud deployments, as system utilization, Quality of Service, operating cost, and computational efficiency could all be improved through the careful evaluation of underlying virtualization technologies.

III. PCI Passthrough and InfiniBand

CPU and memory utilization is only one aspect of application performance for many scientific computing applications. Another key focal point is the I/O bandwidth and interconnect

performance. Advanced networking technologies have been relatively unavailable in clouds, which often only support Gigabit Ethernet (the only notable exception is the use of 10GbE in Amazon Cluster Compute Instances). InfiniBand and MMP interconnects, the backbone of the HPC industry, have largely gone unused in clouds. As such, a comprehensive system for connecting and interweaving virtual machines though a high speed interconnects is paramount to the success of a high performance IaaS architecture.

Two technologies have become available to enable virtualzied environments to leverage the same interconnects as many supercomputers; hardware-assisted I/O virtualization (VT-d or IOMMU) and Single Root I/O Virtualization (SR-IOV). With VT-d, PCI-based hardware passed directly to a guest VM, thereby removing the overhead of communicating with the host OS through emulated drivers. This is also the same mechanisms that enable GPU passthrough, however with different usage scenarioes. When leveraging SR-IOV, VMs gain direct access to InfiniBand adapters via virtual pci functions and use drivers without emulation or modification to achieve near-native I/O performance. With the use of SR-IOV, multiple virtual functions can be assigned directly to different VMs, enabling a sharing of the interconnect fabric. This enables IaaS providers to leverage InfiniBand interconnects for applications that utilize RDMA, IB Verbs, or even IP while simultaneously providing a guaranteed QoS based on SR-IOV configuration.

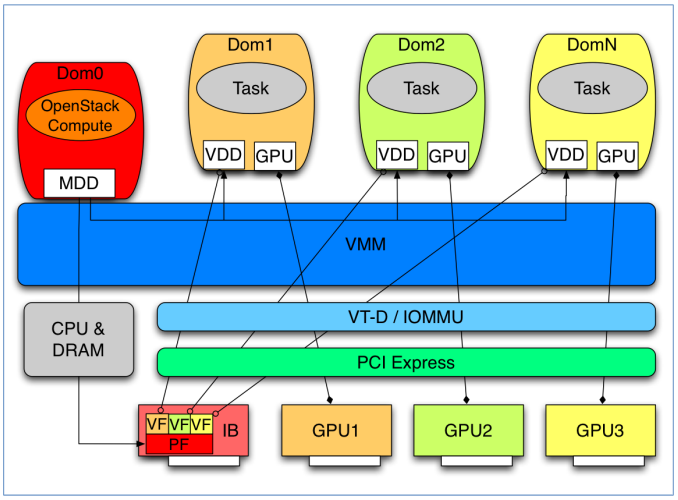

Fig. 3. PCI Passthrough and SR-IOV for InfiniBand and GPUs

Some research is already underway in evaluating SR-IOV enabled InfiniBand clusters. In [5], performance looks to be significantly better than the best-practices of 10GbE networks, however some initial overhead exists. The viability of InfiniBand in Cloud IaaS can also be seen in the new SDSC Comet system recently funded by the NSF [6]. In the collaborative experiment, we are looking at tuning mechanisms for InfiniBand that can reduce the overhead within the KVM hypervisor [7]. Using the OSU MPI microbenchmarks, we've found that Polling mode, Interrupt Coalescing, and Shared Receive Queue (SRQ) limits all have a measurable effect on throughput and latency within VMs.

IV. Accelerators and GPUs

Within HPC, there is a substantial movement toward dedicated accelerator cards such as general purpose graphical processing units (GPGPUs, or GPUs) to enhance scientific computation problems by an upwards of two orders of magnitude. This is accomplished through dedicated programming environments, compilers, and libraries such as CUDA from Nvidia as well as the OpenCL effort [8]. When combining GPUs in an otherwise typical HPC environment or supercomputer, major gains in performance and computational ability have been reported in numerous fields ranging from Astrophysics to Bioinformatics.

Infrastructure-as-a-Service (IaaS) clouds have the potential to democratize access to the latest, fastest, and most powerful computational accelerators similar to those in supercomputing today. However, given the complexity surrounding the choice of GPUs, host systems, and hypervisors, it is perhaps no surprise that Amazon is the only major cloud provider offering customers access to GPU-enabled instances. Furthermore, the only current solution uses outdated hardware and an infrastructure filled with performance slowdowns and implicit overhead. However recently, we have pioneered the ability to provide GPUs directly within a Xen VM infrastucture using PCI-Passthrough that can perform at near-native performance. There are alternative solutions that attempt to virtualize the GPU through an API [9]–[12]. These solutions, typically based on library interposition, essentially redirect the CUDA, OpenCL, OpenGL, or DirectX library calls in order to access GPUs via a network or other shared memory device. This approach works well for some workloads, including desktop virtualization, but is insufficient for performance critical code.

To evaluate the effectiveness of GPU-enabled VMs within Xen, two different machines were used to represent present and upcoming Nvidia GPUs. These machines represent the present Fermi series GPUs along with the recently released Kepler series GPUs, providing a well-rounded experimental environment. The SHOC Benchmark suite [13] was used to extensively evaluate performance across each test case. The SHOC benchmarks were chosen because they provide a higher level of evaluation regarding GPU performance than the sample applications provided in the Nvidia SDK, and can also evaluate OpenCL performance in similar detail. Furthermore, the provide a similar translation between the CPU-based HPCC benchmarks seen in the previous section.

Figure 4 examines the Fast Fourier Transform (FFT), and the traditionally HPC-centric Matrix Multiplication implementations. For all benchmarks that do not take into account the PCI-Express (pcie) bus transfer time, we see notable speedups when using the Kepler GPUs as expected. Interestingly, performance of within Xen VMs is consistently less than 1%, confirming the synthetic MaxFlops benchmark above. However, we do see some performance impact when calculating the total FLOPS with the pcie bus in the equation. This performance decrease ranges significantly for the C2075-series GPU, roughly about a 15% impact for FFT and a 5% impact for Matrix Multiplication. This overhead in pcie runs is not as pronounced for the Kepler K20m test environment, with

near-native performance in all cases (less than 1%).

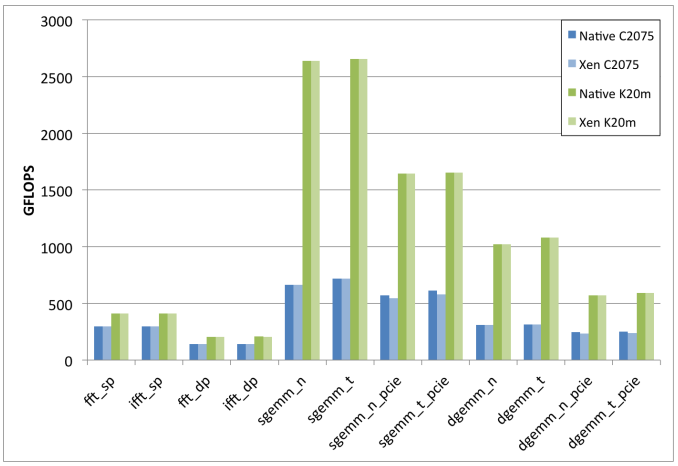

Fig. 4. GPU Fast Fourier Transform & Matrix Multiplication performance

The method of direct PCI Passthrough of GPUs directly to a guest virtual machine running on a tuned Xen hypervisor shows initial promise for an ubiquitous solution in Cloud Infrastructure. As expected with any new technology, we can see that there is a small overhead in using PCI Passthrough of GPUs within VMs, compared to native bare-metal usage. The Kepler K20m GPU-enabled VMs operated at near-native performance for all runs, with a 1.2% reduction at worst in performance. The Kepler based C2075 VMs experience more overhead due to the NUMA configuration that impacted PCI-Express speeds. However, when the overhead of the PCI-Express bus is not considered, the C2075 computations perform at near-native speeds. Recently, the same mechanisms of GPU passthrough have become available with the KVM hypervisor, and our initial testing has shown further improved performance, especially with the PCI Express transfers. Overall, we expect many mid-tier scientific computing groups to benefit the most from the ability to use GPUs in a scientific cloud infrastructure.

V. Direction

Currently implementation and experimentation is under way to evaluate the applicability of a high performance IaaS architecture. Research is currently leveraging the OpenStack IaaS framework, as it represents an open-source, scalable, community driven software stack with a wide consortium of users [14]. Work includes supporting the VM creation complete with GPUs and high speed InfiniBand adapters, with the proper mechanisms at the hypervisor and libvirt API levels. Currently, further investigation is needed on how to best address issues with Non-Uniform Memory Access (NUMA) performance issues, and create enable NUMA-aware VM placement for optimal performance.

This work proposes building a heterogeneous, high performance IaaS by concentrating on virtualization performance, heterogeneous hardware and GPUs, high speed interconnects, and advanced scheduling. Given the increasingly decreasing CPU and memory overhead along with InfiniBand and GPU integration, we hope to show the overhead involved in using cloud infrastructure is becoming negligible. Next steps include moving from benchmarks to real-world applications that are comparable to current cluster systems. Furthermore we look to demonstrate Nvidia's GPU-Direct mechanism that allows for increased speed parallel GPU computation by using InfiniBand's RDMA for inter-node GPU memory transfers, a feature only recently seen even on today's supercomputers.

Acknowledgment

This document was developed with support from the National Science Foundation (NSF) under Grant No. 0910812 to Indiana University for "FutureGrid: An Experimental, High Performance Grid Test-bed" Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. Andrew J. Younge also greatly acknowledges support from The Persistent Systems Fellowship of the School of Informatics and Computing at Indiana University.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] "FutureGrid," Web page. [Online]. Available: http://portal.futuregrid.org

[3] P. Luszczek, D. Bailey, J. Dongarra, J. Kepner, R. Lucas, R. Rabenseifner, and D. Takahashi, "The HPC Challenge (HPCC) benchmark suite," in *SC06 Conference Tutorial*. Citeseer, 2006.

[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)*, July 2011.

[5] 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*. IEEE Computer Society, 2013, pp. 385–392.

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

[7] M. Musleh, V. Pai, J. P. Walters, A. J. Younge, and S. P. Crago, "Bridging the virtualization performance gap for hpc using sriov for infiniband," Information Sciences Institute, Tech. Rep., 2014.

[8] J. E. Stone, D. Gohara, and G. Shi, "Opencl: A parallel programming standard for heterogeneous computing systems," *Computing in science & engineering*, vol. 12, no. 3, p. 66, 2010.

[9] J. Duato, A. J. Pena, F. Silla, R. Mayo, and E. S. Quintana-Orti, "rcuda: Reducing the number of gpu-based accelerators in high performance clusters," in *High Performance Computing and Simulation (HPCS), 2010 International Conference on*. IEEE, 2010, pp. 224–231.

[10] L. Shi, H. Chen, J. Sun, and K. Li, "vcuda: Gpu-accelerated high-performance computing in virtual machines," *Computers, IEEE Transactions on*, vol. 61, no. 6, pp. 804–816, 2012.

[11] V. Gupta, A. Gavrilovska, K. Schwan, H. Kharche, N. Tolia, V. Talwar, and P. Ranganathan, "GViM: GPU-accelerated virtual machines," in *HPCVirt '09 Proceedings of the 3rd ACM Workshop on System-level Virtualization for High Performance Computing*, 2009, pp. 17–24.

[12] G. Giunta, R. Montella, G. Agrillo, and G. Coviello, "A GPGPU transparent virtualization component for high performance computing clouds," in *Euro-Par 2010 - Parallel Processing*, ser. Lecture Notes in Computer Science, P. DAmbra, Ed. Springer, 2010, vol. 6271, pp. 379–391.

[13] A. Danalis, G. Marin, C. McCurdy, J. S. Meredith, P. C. Roth, K. Spafford, V. Tipparaju, and J. S. Vetter, "The scalable heterogeneous computing (shoc) benchmark suite," in *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units*. ACM, 2010, pp. 63–74.

[14] Rackspace, "Openstack," WebPage, Jan 2011. [Online]. Available: http://www.openstack.org/