Can Inter-VM Shmem Benefit MPI Applications on SR-IOV Based Virtualized Infiniband Clusters?[*]

Jie Zhang, Xiaoyi Lu, Jithin Jose, Rong Shi, and Dhabaleswar K. (DK) Panda

Department of Computer Science and Engineering,
The Ohio State University
{zhanjie,luxi,jose,shir,panda}@cse.ohio-state.edu

Abstract. Single Root I/O Virtualization (SR-IOV) technology has been introduced for high-performance interconnects such as InfiniBand. Recent studies mainly focus on performance characteristics of high-performance communication middleware (e.g. MPI) and applications on SR-IOV enabled HPC clusters. However, current SR-IOV based MPI applications do not take advantage of the locality-aware communication on intra-host inter-VM environment. Although Inter-VM Shared Memory (IVShmem) has been proven to support efficient locality-aware communication, the performance benefits of IVShmem for MPI libraries on virtualized environments are yet to be explored. In this paper, we present a comprehensive performance evaluation for IVShmem backed MPI using micro-benchmarks and HPC applications. The performance evaluations show that, through IVShmem, the performance of MPI point-to-point and collective operations can be improved up to 193% and 91%, respectively. The application performance can be improved up to 96%, compared to SR-IOV. The results further show that IVShmem just brings minor overhead compared to native environment.

Keywords: IVShmem, SR-IOV, Virtualization, MPI, InfiniBand.

1 Introduction

Distributed computing infrastructures are becoming increasingly virtualized, owing to the ease of system management and administration. They provide desirable features to meet demanding requirements of computing resources in modern computing systems, including server consolidation, performance isolation and ease of management, along with guaranteeing security, and live migration [21]. Virtual Machine (VM) technologies have already been widely adopted in industry computing environments, especially data-centers. For instance, data-center providers, Amazon's Elastic Compute Cloud (EC2) [1], rely on virtualization to consolidate computational resources for applications from different customers, with required Quality of Service guarantees on the same underlying hardware. Even though virtualization has gained significant momentum in the enterprise computing domain, its adoption in the High Performance Computing (HPC) domain remains lower. One of the biggest hurdles in realizing this objective comes from lower performance of virtualized I/O devices, offered by virtualized computing environments [13]. The performance of virtualized I/O devices is likely to be

[*] This research is supported in part by National Science Foundation grants #OCI-1148371, #CCF-1213084 and #CNS-1347189.

F. Silva et al. (Eds.): Euro-Par 2014, LNCS 8632, pp. 342–353, 2014.
© Springer International Publishing Switzerland 2014

the key driver in the adoption of virtualized cloud computing systems in HPC domains. High performance MPI libraries such as MVAPICH2 [19], OpenMPI [20], can provide sub-microsecond latencies. However, realizing such performance in virtualized environment is still a challenge.

The recently introduced Single Root I/O Virtualization (SR-IOV) [23] offers an attractive alternative for virtualizing I/O devices, when compared to existing software-based virtualization techniques. According to the SR-IOV specification, a PCIe device can present itself as multiple virtual devices and each virtual device can be dedicated to a single VM. Our earlier study [13] indicates that SR-IOV can attain near to native performance for inter-node point to point communication, at the MPI level. However, one of the main drawbacks of SR-IOV is that it does not support VM locality aware communication. Thus, inter-VM communications within the node also have to go through SR-IOV channel, leading to performance overheads. On the other hand, VM communication schemes such as Inter-VM shared memory (IVShmem) [16], offer shared memory backed communication for VMs within a single host. Consequently, we carry out a primitive-level experiment using Perftest-1.2.3 [2], as shown in Figure 1. The experiment compares the primitive level latencies between SR-IOV based IB communication and shared memory communication, and underscore the performance overheads. For 64 bytes message size, the latencies observed are 0.96 and 0.20 μs, for SR-IOV(IB-Send) and IVShmem, respectively. These performance overheads motivate this study, to explore whether IVShmem scheme can benefit MPI communication within a node on SR-IOV enabled InfiniBand clusters.

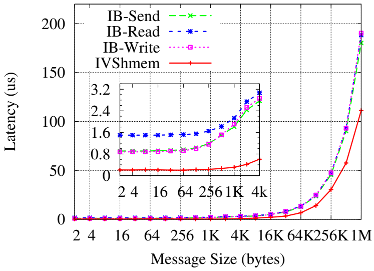


Fig. 1. Primitive-Level Latency Comparison between SR-IOV enabled IB and IVShmem

In this paper, we study the performance characteristics of IVShmem and explore its applicability in VM locality aware communication for MPI libraries on SR-IOV enabled InfiniBand clusters. We propose a high performance prototype design of MPI library, for intra-host inter-VM communication using IVShmem. Then we conduct a comprehensive performance evaluation using micro-benchmarks and HPC applications. The evaluation results indicate that IVShmem scheme has big potential to benefit intra-host inter-VM communication on SR-IOV enabled InfiniBand clusters. This paper mainly focuses on evaluating the performance improvement potential of IVShmem backed MPI communication, on SR-IOV based InfiniBand clusters. We make the following key contributions as part of this paper:

1. Identify the performance overheads associated with SR-IOV for intra-host inter-VM communication
2. Detailed performance evaluations of IVShmem, and exploring its performance improvement potential for VM locality aware communication
3. Performance analysis and scalability evaluations of IVShmem backed MPI library using micro-benchmarks and HPC applications

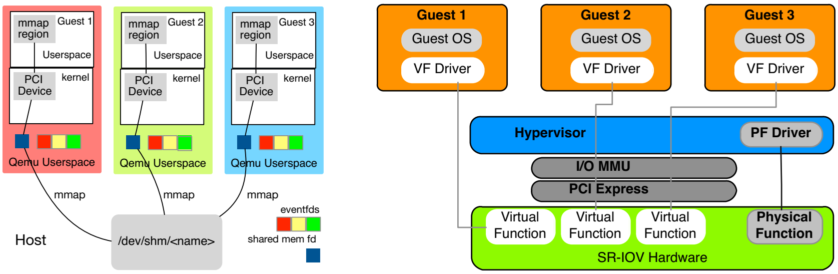
4. Performance comparisons between IVShmem backed and native mode MPI libraries, using HPC applications

The evaluation results indicate that IVShmem can improve point to point and collective operations by up to 193% and 91%, respectively. The application execution time can be decreased by up to 96%, compared to SR-IOV. The results further show that IVShmem just brings small overheads, compared with native environment.

The rest of the paper is organized as follows. Section 2 provides an overview of IVShmem, SR-IOV, and InfiniBand. Section 3 describes our prototype design and evaluation methodology. Section 4 presents the performance analysis results using micro-benchmarks and applications, scalability results, and comparison with native mode. We discuss the related work in Section 5, and conclude in Section 6.

2 Background

Inter-VM Shared Memory (IVShmem) (e.g. Nahanni) [16] provides zero-copy access to data on shared memory of co-resident VMs on KVM platform. IVShmem is designed and implemented mainly in system calls layer and its interfaces are visible to user space applications as well. As shown in Figure 2(a), IVShmem contains three components: the guest kernel driver, the modified QEMU supporting PCI device, and the POSIX shared memory region on the host OS. The shared memory region is allocated by host POSIX operations and mapped to QEMU process address space. The mapped memory in QEMU can be used by guest applications by being remapped to user space in guest VMs. Evaluation results illustrate that both micro-benchmarks and HPC applications can achieve better performance with IVShmem support.



(a) Inter-VM Shmem Mechanism [16]

(b) SR-IOV Mechanism [23]

Fig. 2. Overview of Inter-VM Shmem and SR-IOV Communication Mechanisms

Single Root I/O Virtualization (SR-IOV) is a PCI Express (PCIe) standard which specifies the native I/O virtualization capabilities in PCIe adapters. As shown in Figure 2(b), SR-IOV allows a single physical device, or a Physical Function (PF), to present itself as multiple virtual devices, or Virtual Functions (VFs). Each virtual device can be dedicated to a single VM through the PCI pass-through, which allows each VM to directly access the corresponding VF. Hence, SR-IOV is a hardware-based approach to implement I/O virtualization. Furthermore, VFs are designed based on the existing non-virtualized PFs. Therefore, the drivers of the current adapters can also be used to drive the VFs in a portable manner.

InfiniBand [12] is an industry standard switched fabric designed for interconnecting nodes in HPC clusters. The TOP500 rankings released in November 2013 indicate that more than 41% of the computing systems use InfiniBand as their primary high performance interconnect.

3 Prototype Design and Evaluation Methodology

In this section, we first propose the prototype design for IVShmem based MPI communication and then discuss various dimensions for evaluating the performance impact of IVShmem for intra-host inter-VM communication on SR-IOV based InfiniBand clusters. The results of evaluation for each dimension are described in Section 4.

3.1 Prototype Design

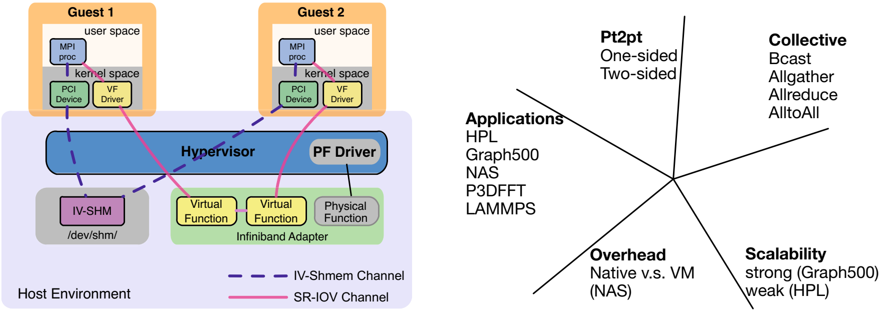
As introduced in Section 2, SR-IOV and IVShmem are two different mechanisms that can be used for intra-host inter-VM communication. To better illustrate, the two inter-VM communication schemes are presented in Figure 3(a). For SR-IOV scheme, which is shown in the solid line, each VM is configured with a dedicated Virtual Function, so that an MPI process in Guest-1 can communicate with another MPI process in Guest-2 without concerning whether Guest-2 is co-located with Guest-1 in a same physical node or not. This does not deliver the best approach to high performance communication. In order to take advantage of shared memory between VMs co-located in a given host, guest VMs need to detect which VMs are co-located with themselves, so that they can map the same memory region into their own memory spaces. Based on what we discussed in Section 2, IVShmem provides a mechanism to expose a host memory region to all co-resident VMs as virtual PCI devices. And finally, this memory region can be mapped to user spaces of guest systems. We implement a prototype MPI library by utilizing IVShmem. Therefore, the communication between co-resident VMs can happen along the IVShmem channel as shown in the dashed line in Figure 3(a), instead of the SR-IOV channel, as shown in the solid line.

3.2 Evaluation Dimensions

We follow a five-pronged approach to evaluate the performance improvement potential of IVShmem for intra-host inter-VM communication on SR-IOV based InfiniBand clusters, as shown in Figure 3(b).

Point to Point Communication: Point to point communication is a basic communication scheme in MPI communication. On virtualized environments with SR-IOV support, our earlier studies [13] showed related performance evaluations. In this paper, we mainly evaluate the performance improvement potential of IVShmem for point to point communication including both two-sided and one-sided operations.

Collective Communication: Collective communication is an important and frequently used communication scheme of MPI. However, current SR-IOV solution does



(a) Inter-VM Communication Channels (b) Evaluation Dimensions

Fig. 3. Inter-VM Communication Channels and Evaluation Dimensions

not take advantage of the locality aware collective communication on intra-host inter-VM environment, which leads to performance overhead. Therefore, we evaluate the performance improvement potential of IVShmem for four widely used collective operations across VMs on a single node in this paper.

Application Execution Time: MPI has established itself as the de-facto standard of programming model for HPC applications. Clearly, the performance of MPI libraries will significantly impact the execution time of these HPC applications. Thus, we choose five representative HPC applications (as shown in Table 1) to evaluate the performance benefits of IVShmem.

Table 1. Representative HPC Applications for Evaluation

Name	Description
P3DFFT	Parallel Three-Dimensional Fast Fourier Transforms, dubbed P3DFFT [5], is a library for large-scale computer simulations in a wide range of sciences, such as physics, climatology and chemistry.
HPL	High Performance Linpack (HPL) is the parallel implementation of Linpack [7] and the performance measure for ranking the computer systems of the Top 500 supercomputer list.
LAMMPS	LAMMPS stands for Large-scale Atomic/Molecular Massively Parallel Simulator [22]. It is a classical molecular dynamics simulator from Sandia National Laboratory.
Graph500	Graph500 [24] is one of the representative benchmarks of Data intensive supercomputer applications. It exhibits highly irregular communication pattern.
NAS	NAS [3] contains a set of benchmarks which are derived from the computing kernels, which is common on Computational Fluid Dynamics (CFD) applications. These represent the class of regular iterative HPC applications.

Virtual Machine Scalability: As the emergence of virtualization technology, we can achieve easier system management and performance isolation. However, the performance characteristics might vary significantly as the number of VMs increase. This paper evaluates the performance impact of IVShmem scheme by adjusting the number of VMs within a physical node in SR-IOV enabled InfiniBand clusters.

Performance Overhead: Earlier studies indicate that high performance VM environments are able to achieve low cost of CPU and memory virtualization [25]. I/O virtualization, however, leads to longer I/O latency, since I/O devices are usually shared by multiple VMs within a host. In this paper, we evaluate the performance overheads of SR-IOV and IVShmem compared to native environment.

4 Performance Evaluation

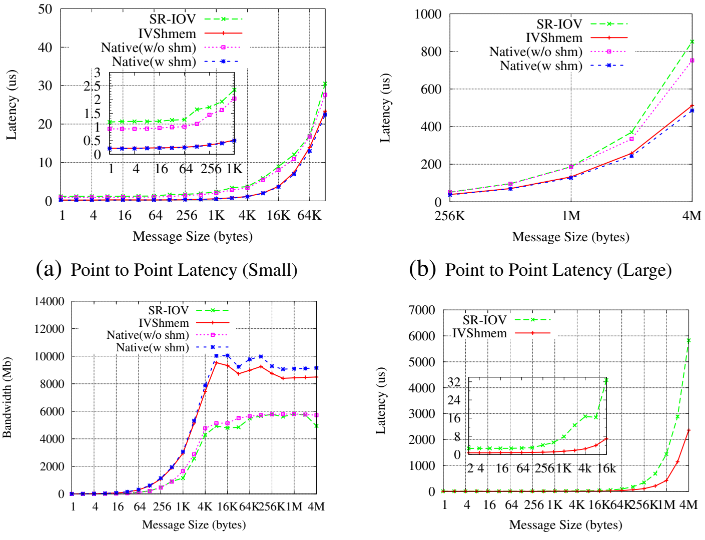
In this section, we describe our experimental testbed and discuss our evaluation of two-sided and one-sided point to point, collective operations, and HPC applications. Since this paper focuses on performance evaluation of IVShmem scheme on InfiniBand clusters with SR-IOV support, we use one node with multiple cores for evaluation.

4.1 Experiment Setup

Our testbed is an InfiniBand cluster, where each node has dual 8-core 2.6 GHz Intel Xeon E5-2670 (Sandy Bridge) processors with 20MB L3 shared cache, 32 GB main memory and equipped with Mellanox ConnectX-3 FDR (56 Gbps) HCAs with PCI Express Gen3 interfaces. We use RedHat Enterprise Linux Server release 6.4 (Santiago) with kernel 2.6.32-279.19.1.el6.x86_64 as the host OS.

We use the Mellanox OpenFabrics Enterprise Distribution MLNX_OFED_LINUX 2.1-1.0.0 to provide the InfiniBand interface with SR-IOV support and KVM as the Virtual Machine Monitor (VMM). Each VM is pinned to a single core and has 1.5 GB main memory. The OS used in each VM is RedHat Enterprise Linux Server release 6.4 (Santiago) with kernel 2.6.32-131.0.15.el6.x86_64.

All applications and libraries used in this study are compiled with gcc 4.4.6 compiler. All MPI communication performance experiments use MVAPICH2 2.0rc1 and OSU Micro-Benchmarks. Experimental results are averaged by 5 runs to ensure fair comparison. Our tests are conducted with different numbers of VMs on one node, 8 for power of two case, and 15 for full-subscribed case (while reserving one core for host OS).



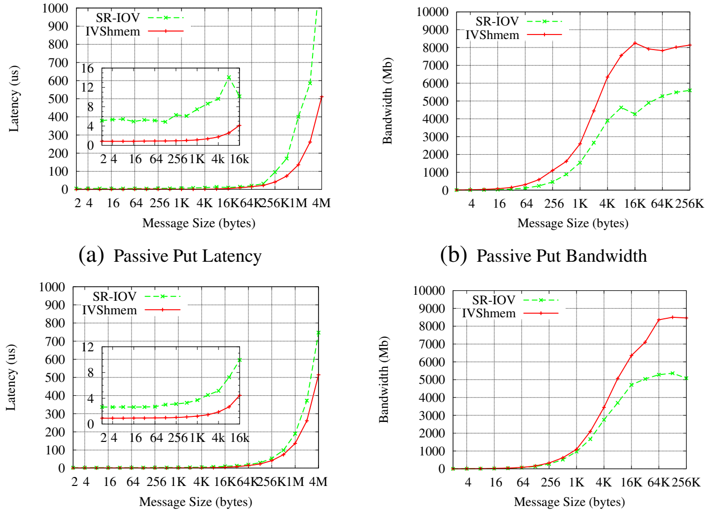
(a) Point to Point Latency (Small)

(b) Point to Point Latency (Large)

(c) Point to Point Bandwidth

(d) Multi-Pair Latency (14 Pairs)

Fig. 4. Two-sided Point to Point Performance



(a) Passive Put Latency

(b) Passive Put Bandwidth

(c) Passive Get Latency

(d) Passive Get Bandwidth

Fig. 5. One-sided Point-to-Point Performance

4.2 Point to Point Communication Performance

In this section, we evaluate the MPI level point to point performance for intra-node inter-VM communication in terms of latency and bandwidth. Figure 4(a) and Figure 4(b) show the two-sided point to point latencies of small and large message sizes, respectively. We can observe that, IVShmem based MPI library achieves lower latency for both small and large message sizes, compared to the SR-IOV. For example, the latency of SR-IOV is 1.2μs, while it is 0.22μs for IVShmem at 4 bytes message size. The experimental results indicate that the latency based on IVShmem can be decreased up to 82%, compared to that of SR-IOV. With respect to point to point bandwidth, we can see from Figure 4(c) that IVShmem can significantly improve the bandwidth for various message size ranging from 1 byte to 4 MB. The improvement is up to 158%. The peak bandwidth that IVShmem can achieve is near to 10 GB per sec, while it is around 6 GB per sec for SR-IOV. We also evaluate the performance gains that comes from using shared memory instead of InfiniBand for intra-node communication in native environment. Compared to not using shared memory (w/o shm), the performance of native MPI can be improved by enabling shared memory (w shm) up to 77% and 191% in terms of latency and bandwidth. From these, we can see that the performance gains of using IVShmem instead of SR-IOV for intra-node communication in the virtualized environment matches the gains that we observed in the native environment here.

Another important point we can observe is that IVShmem attains near to native performance in terms of latency and bandwidth. The latency overheads compared to native performance are 3%-5% at small message sizes. For example, the latencies for IVShmem and native at 256 bytes message size are 0.35μs and 0.34μs, respectively. The overhead is only 3%. We also present the evaluation results of multi-pair latency (7 pairs) in Figure 4(d). At 4 bytes message size, the latency of IVShmem is 0.77μs, while it is 2.72μs for SR-IOV. When the message size varies from 1 byte to 4 MB, IVShmem can decrease the latency by up to 86%, compared to SR-IOV. Thus, IVShmem can significantly improve the point to point communication performance for MPI library compared to SR-IOV, and can also achieve near to native performance.

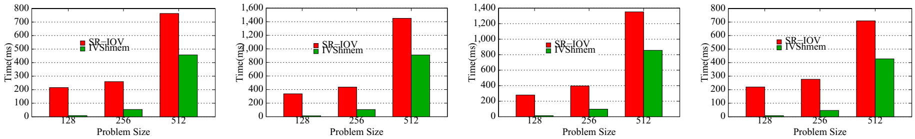
The recent MPI standard [18] has introduced one-sided communication model. In this model, one process's memory can be updated directly by another process. Unlike MPI two-sided communication model in which both sender and receiver are involved for data transfer, one-sided communication allows one process to specify all necessary parameters, and synchronization is done explicitly to ensure the completion of communication. As it can be seen from Figure 5(a) and Figure 5(b), IVShmem based MPI one-sided passive Put operation achieves lower latency and higher bandwidth, compared to SR-IOV. The latency is decreased up to 85% at 1 KB message size, while bandwidth can be improved up to 193% at 16 bytes message size. Similarly, the evaluation results shown in Figure 5(c) and 5(d) indicate that IVShmem also benefits one-sided passive Get operation in terms of latency and bandwidth. Similar performance improvements are observed for passive Get operation. The results indicate that IVShmem scheme can significantly improve performance of one-sided and two-sided point-to-point communications operations.

4.3 Collective Communication Performance

We select four widely used collective communication operations in our evaluations: Broadcast, Allgather, Allreduce and Alltoall. Figure 11(a) to Figure 11(d) show that, compared to SR-IOV, IVShmem significantly cuts down the latencies of the above four collective operations across 15 VMs. For example, at 4 bytes message size, the latency of broadcast operation for IVShmem is $0.5\mu s$, while it is $4.15\mu s$ for SR-IOV. From 1 byte to 1 MB message size, the latencies can be decreased up to 91%, 87%, 85% and 88% through IVShmem for the above four collective operations, respectively. Based on our experimental evaluations, IVShmem can remarkably improve MPI collective communication performance within one node.

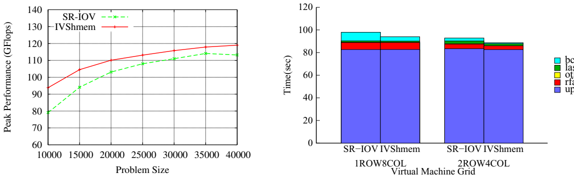
4.4 Application Performance

As discussed in Section 3, many of the HPC applications rely on MPI performance. In this section, therefore, we evaluate the performance benefit of IVShmem using real HPC applications. According to above evaluations on four collective communication operations, we use several HPC applications, each one as a representative mainly corresponding to one or two particular collective operations. From Figure 6(a) to Figure 6(d), we depict the evaluation results of different test programs in P3DFFT library, which are `test_inverse.c`, `test_rand.c`, `test_sine.c` and `test_spec.c`. The `inverse` evaluation results using 15 VMs are shown in Figure 6(a). As we can see, the execution times can be decreased by 96%, 79%, 40% through IVShmem for input size 128, 256, 512, respectively. The execution times of `rand` also can be reduced by 96%, 76%, 37%. Similar results can be observed for `sine` and `spec`. This is because the majority of the total execution time is spent in `MPI_Alltoall` operation. However, as the problem size increases, the proportion of communication drops down, and thus the performance improvement decreases. The evaluation results indicate that IVShmem can effectively reduce the execution time of the above four P3DFFT test programs. And it also verifies the evaluation results of collective communication in Section 4.3.



(a) P3DFFT inverse (b) P3DFFT rand (c) P3DFFT sine (d) P3DFFT spec

Fig. 6. P3DFFT Performance on 15 VMs



(a) Peak GFlops (15 VMs) (b) Time Decomposition (8 VMs)

Fig. 7. HPL Performance

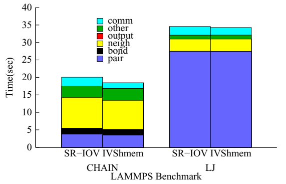


Fig. 8. LAMMPS Performance

The HPL evaluation results are presented in Figure 7. Here, we first measure the peak performance achieved by launching tests on 15 VMs as shown in Figure 7(a). Both SR-IOV and IVShmem achieve peak performance when the problem size is larger than 40,000. In our evaluations, IVShmem outperforms SR-IOV by around 4%-18% in GFLOPS, for various experiments. To better analyze the communication cost, we decomposed the time of HPL benchmark by using 8 VMs with various VM grid configuration. From Figure 7(b), we observe that the main communication benefit in HPL is coming from Broadcast. Through IVShmem, the broadcast latency can be decreased by 66% and 50% for 2x4 and 1x8 grids, respectively.

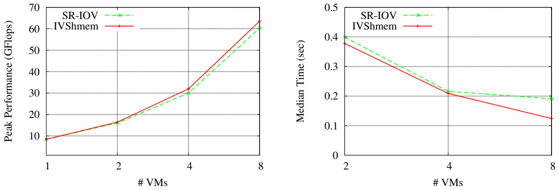
We also profile the time decomposition of Chain and LJ benchmark in LAMMPS. Figure 8 shows that IVShmem can decrease the communication time by 36% and 13% for Chain and LJ, respectively. And the total execution time can be decreased by up to 8% for Chain.

4.5 Virtual Machine Scalability

In this section, we evaluate the virtual machine scalability to explore the performance impact on increasing the number of virtual machines in a single host. Such evaluation helps to determine the optimal number of virtual machines to be deployed within a single host. We measure the weak scalability of HPL with fixed memory usage of each VM and increasing number of VMs. Figure 9(a) shows that IVShmem brings 2%-7% benefits compared to SR-IOV. We also use Graph500 benchmarks to evaluate the strong scalability of IVShmem and SR-IOV. As shown in Figure 9(b), IVShmem exhibits better scalability and decreases the execution time up to 35%, compared to SR-IOV.

4.6 Performance Overhead

For performance overhead evaluation, we used NAS to run seven different computing kernels of class B: IS, MG, CG, LU, FT, BT and SP. The first 5 kernels ran across 8 VMs, while BT and SP ran across 9 VMs, based on the requirement of these two application kernels. It can be noted from Figure 10, IVShmem reduces the execution



(a) HPL Weak Scalability (b) Graph500 Strong Scalability

Fig. 9. Virtual Machine Scalability

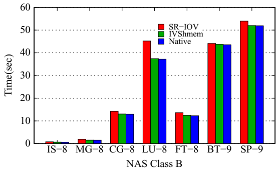


Fig. 10. Performance Overhead Comparison

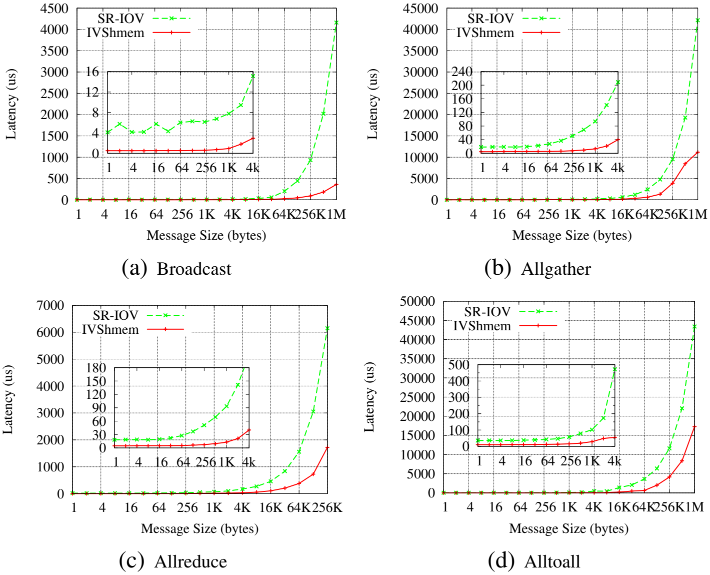
times for NAS Parallel Benchmarks - IS (21%), MG (19%), LU (17%), compared to SR-IOV. We also ran them on native environment, and we observe that IVShmem only introduces around 5% overhead compared to native performance. Our evaluation results indicate that IVShmem introduces a small overhead.

5 Related Work

I/O virtualization can be broadly classified into two categories – software based and hardware based. Earlier studies such as [17] and [4] have shown network performance evaluation of software-based approaches in Xen. Studies [14,6,11] have demonstrated that SR-IOV is significantly better than software-based solutions for 10GigE networks. In [14], the authors have provided a detailed performance evaluation on the environment of SR-IOV capable 10GigE in KVM. They have studied several important factors that affect network performance in both virtualized and native systems. Further, studies [9,15,10] with Xen have demonstrated the ability to achieve near-native performance in VM-based environment for HPC.

Our previous study of the performance characteristics of using SR-IOV with InfiniBand [13] has shown that while SR-IOV enables low-latency communication, MPI libraries need to be designed carefully and offer advanced features for improving intra-node, inter-VM communication. Previously, we proposed designs for improving intra-node inter-VM communication by using an Inter-VM Communication Library (IVC) and re-designed the MVAPICH2 library to leverage the features offered by the IVC [8]. However, this solution was based on the Xen platform and did not show the studies with SR-IOV enabled InfiniBand clusters. In addition, an implementation of IVShmem [16] provided the detailed introduction of Nahanni, a IVShmem implementation. Based on the implementation, the authors developed the MPI-Nahanni user-level library, which is ported to the Nemesis channel in MPICH2 library. Their design used memory-mapped shared memory provided by Nahanni in order to accelerate inter-VM communication on the same host.

Different from the previous work, this paper presents a comprehensive performance improvement potential study of IVShmem for intra-host inter-VM communication based on MVAPICH2 library on SR-IOV enabled InfiniBand clusters. Performance evaluation shows promising results of IVShmem based MPI communication using point to point, collective micro-benchmarks and several representative HPC applications. This paper is the first paper to carry out performance studies with IVShmem on SR-IOV enabled InfiniBand clusters.



(a) Broadcast

(b) Allgather

(c) Allreduce

(d) Alltoall

Fig. 11. Collective Communication Performance on 15 VMs

6 Conclusion and Future Work

In this paper, we have studied the performance improvement potential of IVShmem for intra-host inter-VM MPI communication. We have briefly introduced the prototype design of a high performance MPI library for intra-host inter-VM communication using IVShmem. And then we have conducted detailed performance evaluations using MPI micro-benchmarks and representative HPC applications. Our performance evaluations using micro-benchmarks show that IVShmem based MPI library improves point to point (two-sided and one-sided) and collective performance by up to 193% and 91%, respectively. Application evaluation results indicate that based on IVShmem, the execution times of NAS, P3DFFT, LAMMPS benchmarks were decreased by up to 21%, 96%, 8%, respectively, compared to SR-IOV. And the peak performance of HPL is improved by 18% using IVShmem. The evaluations using Graph500 and NAS also demonstrate that IVShmem based MPI library shows good scalability and introduces minor overhead, compared to native performance.

In the future, we plan to continue our research along this direction, and provide a high performance MPI library design to dynamically switch between IVShmem and SR-IOV for efficiently supporting locality aware MPI communication across nodes on SR-IOV enabled InfiniBand clusters.

References

1. Amazon EC2, `http://aws.amazon.com/ec2/`
2. CPMD Consortium, `http://www.openfabrics.org/downloads/perftest/`
3. NAS Parallel Benchmarks,
`http://www.nas.nasa.gov/Resources/Software/npb.html`

4. Apparao, P., Makineni, S., Newell, D.: Characterization of Network Processing Overheads in Xen. In: Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, VTDC 2006. IEEE Computer Society, Washington, DC (2006)
5. Pekurovsky, D.: P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions. SIAM Journal on Scientific Computing 34(4), C192–C209 (2012)
6. Dong, Y., Yang, X., Li, J., Liao, G., Tian, K., Guan, H.: High Performance Network Virtualization with SR-IOV. Journal of Parallel and Distributed Computing (2012)
7. Dongarra, J.J., Duff, L.S., Sorensen, D.C., Vorst, H.A.V.: Numerical Linear Algebra for High Performance Computers. Society for Industrial and Applied Mathematics (1998)
8. Huang, W., Koop, M.J., Gao, Q., Panda, D.K.: Virtual Machine aware Communication Libraries for High Performance Computing. In: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing, SC 2007, pp. 9:1–9:12. ACM, New York (2007)
9. Huang, W., Liu, J., Abali, B., Panda, D.K.: A Case for High Performance Computing with Virtual Machines. In: Proceedings of the 20th Annual International Conference on Supercomputing, ICS 2006, New York, NY, USA (2006)
10. Huang, W., Liu, J., Koop, M., Abali, B., Panda, D.: Nomad: Migrating OS-bypass Networks in Virtual Machines. In: Proceedings of the 3rd International Conference on Virtual Execution Environments, VEE 2007, New York, NY, USA (2007)
11. Huang, Z., Ma, R., Li, J., Chang, Z., Guan, H.: Adaptive and Scalable Optimizations for High Performance SR-IOV. In: Proceedings of 2012 IEEE International Conference on Cluster Computing (CLUSTER), pp. 459–467. IEEE (2012)
12. Infiniband Trade Association, `http://www.infinibandta.org`
13. Jose, J., Li, M., Lu, X., Kandalla, K., Arnold, M., Panda, D.: SR-IOV Support for Virtualization on InfiniBand Clusters: Early Experience. In: Proceedings of 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 385–392 (May 2013)
14. Liu, J.: Evaluating Standard-Based Self-Virtualizing Devices: A Performance Study on 10 GbE NICs with SR-IOV Support. In: Proceedings of IEEE International Symposium on Parallel & Distributed Processing (IPDPS), pp. 1–12. IEEE (2010)
15. Liu, J., Huang, W., Abali, B., Panda, D.K.: High Performance VMM-bypass I/O in Virtual Machines. In: Proceedings of the Annual Conference on USENIX 2006 Annual Technical Conference, ATC 2006, Berkeley, CA, USA (2006)
16. Macdonell, A.C.: Shared-Memory Optimizations for Virtual Machines. PhD Thesis. University of Alberta, Edmonton, Alberta, Fall (2011)
17. Menon, A., Santos, J.R., Turner, Y., Janakiraman, G.J., Zwaenepoel, W.: Diagnosing Performance Overheads in the Xen Virtual Machine Environment. In: Proceedings of the 1st ACM/USENIX International Conference on Virtual Execution Environments, VEE 2005, pp. 13–23. ACM, New York (2005)
18. MPI Forum: MPI: A Message Passing Interface. In: Proceedings of Supercomputing (1993)
19. MVAPICH2: High Performance MPI over InfiniBand and iWARP, `http://mvapich.cse.ohio-state.edu/`
20. OpenMPI: Open Source High Performance Computing, `http://www.open-mpi.org/`
21. Rosenblum, M., Garfinkel, T.: Virtual Machine Monitors: Current Technology and Future Trends. Computer 38(5), 39–47 (2005)
22. Plimpton, S.: Fast Parallel Algorithms for Short-Range Molecular Dynamics. J. Comp. Phys. 117, 1–19 (1995)
23. Single Root I/O Virtualization, `http://www.pcisig.com/specifications/iov/single_root`
24. The Graph500, `http://www.graph500.org`
25. Huang, W., Liu, J.X., Abali, B., Panda, D.K.: A Case for High Performance Computing with Virtual Machines. In: The Proceedings of 20th Annual International Conference on Supercomputing (ICS), Queensland, Australia, June 28-30 (2006)