Deployment of Virtual InfiniBand Clusters with Multi-tenancy for Cloud Computing

Viktor Mauch

Steinbuch Centre for Computing (SCC)
Karlsruhe Institute of Technology (KIT)
Email: viktor.mauch@kit.edu

***Abstract*—Today, most high performance computing (HPC) systems are equipped with high-speed interconnects providing low communication and synchronization latencies in order to run tightly coupled parallel computing jobs. They are typically managed and operated by individual institutions and offer a fixed capacity and static runtime environment with a limited selection of applications, libraries and system software components. On the contrary, a cloud-based Infrastructure-as-a-Service (IaaS) model for HPC resources promises more flexibility, as it enables elastic on-demand provisioning of virtual clusters and allows users to modify the runtime environment down to the operating system level. The goal of this research effort is the general demonstration of a prototypic HPC IaaS system allowing automated provisioning of virtualized HPC resources while retaining high and predictable performance. We present an approach to use high-speed cluster interconnects like InfiniBand within an IaaS environment. Our prototypic system is based on the cloud computing framework Openstack in combination with the Single Root - I/O Virtualization (SR-IOV) mechanism for PCI device virtualization. Our evaluation shows that, with this approach, we can successfully provide dynamically isolated partitions consisting of multiple virtual machines connected over virtualized InfiniBand devices. Users are put in the position to request their own virtualized HPC cluster on demand. They are able to extend or shrink the assigned infrastructure and to change the runtime environment according to their needs. To ensure the suitability for HPC applications, we evaluate the performance of a virtualized cluster compared to a physical environment by running latency and High-Performance Linpack (HPL) benchmarks.**

Keywords–HPC, InfiniBand, Cloud Computing, Virtualization, Openstack

I. INTRODUCTION

In recent years, cloud computing [1][2] has influenced significantly most parts of information technology. The consumption of applications, services and infrastructure provided by public operators, has increased dramatically. However, still today the demand of High Performance Computing (HPC) resources is typically covered by local installations provided and used by single institutions. Such physically operated clusters have disadvantages. Due to specific requirements regarding performance and scope, it is common to deploy a predefined, fixed runtime environment with specific applications, libraries, job schedulers and operating systems. As a result, users are limited to implement customized application scenarios based on modifications of the underlying operating system or other important core runtime libraries. Furthermore, demand is fluctuating, resulting in periods where physical resources are underutilized or overloaded.

A High Performance Cloud Computing (HPC2) [3] model based on an Infrastructure-as-a-Service (IaaS) delivery solution promises more flexibility and efficiency in terms of cost and energy consumption. It allows moving away from physically owned but underutilized HPC clusters designed for peak workloads to vitualized elastic HPC resources leased from a consolidated large HPC computing center working near full capacity. The deployment of virtual machines (VM) allows users to securely gain administrative privileges and customize the runtime environment according to their specific demands. Incurred costs are associated directly by a *pay-as-you-go* model with the corresponding resource usage or with the responsible user respectively.

In the next Section, we discuss challenges concerning HPC in the cloud, followed by the architecture description in Section III. Section IV includes some detailed information about our prototypic implementation. Based on that, a performance evalution is provided in Section V. Conclusion and outlook can be found in Sections VI and VII.

II. HPC CLOUD CHALLENGES

Providing cloud-based HPC services raises difficult challenges. Virtualization, the core technique of general purpose IaaS offerings, certainly achieves the desired elasticity and multi-tenancy. On the other hand, virtualized environments are associated with a higher overhead and may lead to unpredictable variations in performance. Early studies [4][5] concerning the evaluation of the Elastic Compute Cloud (EC2) standard services, provided by the Amazon Web Services (AWS), confirm these observations. Further research work [6][7] concerning the execution of HPC applications in contemporary non-HPC IaaS environments has identified the network performance as the primary hindrance to implement virtualized HPC clusters. Therefore, I/O virtualization is one of the key challenges of providing HPC cloud resources with high-speed interconnect support. Since 2010, Amazon provides so-called *cluster compute* instances for EC2, which are equipped with a 10GbE interconnect and thus are more capable to handle typical HPC tasks with an acceptable performance [8]. In theory, this service allows to build up a virtualized HPC cluster listed in the TOP500 list, which was demonstrated by Amazon for advertising purposes. Over 1064 instances with 17024 cores reached place 42 of the TOP500 list (November

ISBN: 978-1-61208-388-9

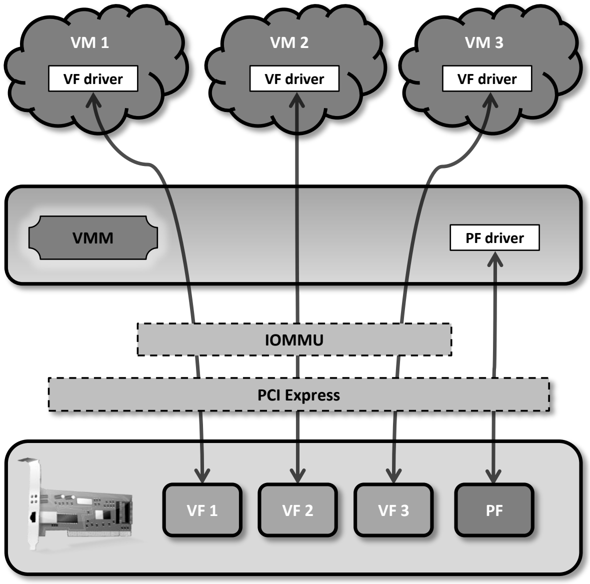

Figure 1. Virtualized I/O device access via SR-IOV. The PCI device presents itself as several *virtual functions* which are allocated to the VMs via PCI passthrough.

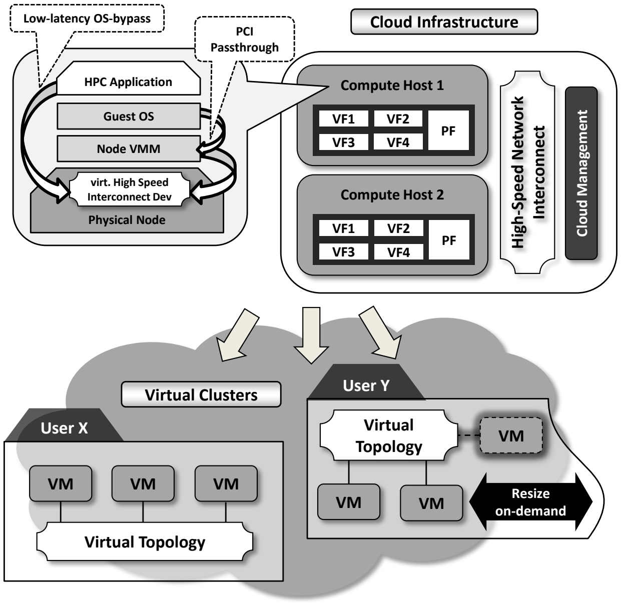

Figure 2. Virtualization of host and the cluster interconnect is managed by a cloud computing framework to provide elastic virtual servers.

2011) with an HPL benchmark of 240TFlops/s. However, the provisioning of VMs with a high-speed interconnect, such as InfiniBand improves performance significantly [9].

InfiniBand (IB) [10] has a substantial performance advantage due to the processing of all network layers within the device hardware. Especially, tightly coupled HPC applications benefit from the very low communication latency in comparison to a traditional network technology such as Ethernet. However, using IB within a virtualized environment is a non-trivial task that can only be partially achieved by software-based approaches. Li et al. [11] have proposed Virtual Machine Monitor (VMM)-bypass I/O, a para-virtualization approach for InfiniBand on Xen. This solution requires ongoing modifications of drivers in host and guest with respect to changes of the underlying hardware and operating system. The concept of Peripheral Component Interconnect (PCI) passthrough grants a VM direct and exclusive access to a dedicated PCI I/O device. It requires an I/O Memory Mapping Unit (IOMMU) to ensure memory protection between different VMs [12] and restricts the number of VMs per host to the number of I/O devices built in. A more suitable solution would be *Single Root - I/O Virtualization (SR-IOV)* [13], which allows a single PCI Express device to appear as multiple, separate devices, called *Virtual Functions (VF)*, a kind of a ”light weighted” PCIe function. Each VM can be allocated to one VF via PCI passthrough. The *Physical Function (PF)* includes the SR-IOV capability and has full configuration resources such as discovery, management and manipulation. It is an anchor for creating VFs and reporting errors and events. Figure 1 provides an overview of the SR-IOV dependencies.

In this paper, we present an architecture for the deployment of multi-tenant virtual clusters, based on the virtualization of the IB interconnect, with acceptable performance and latencies compared to native clusters. In principle, the following implementation design is based on our former research work [3][14]. A detailed evaluation of the architectural approach based on PCI passthrough in combination with OpenNebula has also been worked out by Hillenbrand [15]. With this paper, we extend our approach by using SR-IOV and the deployment of multiple VMs per hosts with IB support within a cloud computing IaaS environment. The next sections provide a fundamental description of the architecture and the prototypic implementation of our solution followed by a basic performance evaluation.

III. Architecture

In relation to the National Institute of Standards and Technology (NIST) definition [1] of cloud computing, two essential characteristics are crucial for HPC IaaS cloud systems. *Resource Pooling* and infrastructure multitenancy is necessary to provide cloud computing services to multiple independent users. It demands the isolation of each cluster network at any time. The user gets the impression to use the interconnect exclusively, albeit with reduced bandwidth. *On-demand self-service* requires automated allocation of virtual HPC resources including the configuration of the cluster network interconnect. Furthermore, *service level agreements* on the minimum network quality have to be guaranteed as HPC tasks may heavily depend on it. Figure 2 illustrates our architectural approach. We extend an existing cloud computing framework with features to manage the provisioning of virtual clusters as well as the configuration of the underlying IB interconnect topology. Available *Virtual Functions* are allocated to new VMs which are able to run HPC applications using the virtual IB device by low-latency bypass.

Using SR-IOV for IB virtualization simplifies many aspects with respect to network isolation, management and security. A VF assigned to the VM is restricted by the physical device hardware compared with an exclusive access to a dedicated physical device via PCI passthrough. First of all, users with administrative privileges within a VM are not able to modify the firmware of the physical PCI device anymore. This is a very important aspect for cloud infrastructure, which is used by multiple users over time. Further restrictions prevent the execution of a *subnet manager* within a VM, which could be used to reconfigure the whole IB network topology.

Users should be able to provide their own VM templates with the corresponding software environment and deploy and resize their VM ensembles on-demand. The underlying cloud computing framework must ensure the network isolation of each user specific VM ensemble at any time. At first glance, the provision of multiple VMs with high performance network interconnect support on single hosts seems to be impractical concerning the usage of HPC workloads. Users would always try to allocate the available hardware with few overhead as possible. However, mixed IaaS environments with common and HPC-capable VMs could utilize the available infrastructure more efficiently. Furthermore, resource over-provisioning could also reduce operation costs. While users would get lower guarantees concerning computing and networking quality, IaaS advantages like flexibility and on-demand provision still can be used compared with the traditional operation of native HPC clusters.

IV. Implementation

The orchestration of HPC resources and the configuration of the network infrastructure with respect to isolated partition is done by a cloud computing framework. We decided to use the Openstack framework, which is currently one of the most popular and promising open-source cloud computing IaaS frameworks on the market. The latest stable version with the codename *Havana*, which we use for our prototypic implementation, already provides the necessary PCI passthrough mechanisms for the assignment of PCI devices to VMs. Openstack supports several hypervisor solutions. We decided to use the Kernel-based Virtual Machine (KVM) hypervisor for node virtualization running on Linux, as Linux can be seen as the de-facto standard operating system for HPC systems [16] and is supported by Mellanox with SR-IOV capable drivers.

The underlying hardware infrastructure to operate our prototypic HPC cloud consists of two *Dell R710* servers. Each of them is equipped with two *Intel Xeon E5620* quadcore 2.66GHz cpus, 64 GB of RAM and a *Mellanox ConnectX-2 InfiniBand Quad Data Rate (QDR) HCA*, which is configured to provide 7 VFs to the host system. Both nodes are connected with Ethernet (1 Gbit/s) and an InfiniBand Doube Data Rate (DDR) switch, which provides a bandwidth of 20Gbit/s. *CentOS 6.4* is used as host and guest operating system together with the *Mellanox OFED 2.0* software stack, which provides drivers and management tools for the integrated IB devices.

To ensure and manage the isolation of virtual clusters with IB partitions at any time, we extend the Openstack framework with the necessary functionality. The IB subnet manager provides a pre-configured *partition key (pkey)*-table to all existing Openstack compute nodes with IB devices. Our Openstack extension registers continuously any changes concerning all available VMs and their associated users. The isolation of an ensemble of logically related VMs is accomplished by assigning a specific pkey to the corresponding VFs. This is done by automated configuring the virtual-to-physical pkey mappings within the host operating systems of the compute nodes. Thereby each virtual IB cluster gets its own pkey and is blocked communicating to other virtual clusters over IB or manipulating IB network topology. This mechanism has similarities with respect to the Virtual Local Area Network (VLAN) technique used by Ethernet network technology. So, our prototypic OpenStack implementation allows users to deploy and resize their VM cluster without having network conflicts with other VMs in the same IB subnet.

TABLE I. HIGH PERFORMANCE LINPACK BENCHMARK

Infrastructure	vCPU config	GFlops	Efficiency
2 nodes (8 Cores)	N/A	156.8	92.1%
4VMs (4 vCores)	dynamic	149.4	87.8%
4VMs (4 vCores)	fixed	152.8	89.8%
8VMs (2 vCores)	dynamic	141.3	83.0%
8VMs (2 vCores)	fixed	143.5	84.3%

Using SR-IOV leads to limitations. Although the SR-IOV specification allows up to 255 VFs per PCI device, the actual usable number is often extremely lower (7–15) because of strong dependencies on the BIOS, chip set and adapter hardware. The capacity of the pkey-table also depends on the provided hardware and is more significant. The above mentioned *Mellanox HCAs* used in our prototypic system are limited to 128 *pkeys*. These circumstances must be taken into account concerning a possible *scale up/out* of the cloud infrastructure.

V. Performance Evaluation

In this section, we present a basic performance evaluation of our early prototypic system. In order to get an impression of the HPC performance, the High Performance Linpack (HPL) [17] benchmark has been executed on several virtual cluster scenarios as well as on the underlying native hardware, see Table I. Therefore, we have used *Intel Optimized LINPACK Benchmark*, which is based on Intel Math Kernel Library (MKL) and the Intel MPI implementation. The corresponding *HPL.dat* tuning parameters for all test scenarios are set as follows: N=100k, NB=168, p=4, q=4. The peak performance of our *Westmere*-based 16-core infrastrucutre is calculated to R_{max}=170.2 Gflops. The test results, we obtained, are looking very promising. As expected, result values provided by virtual clusters are weaker compared to the native environment. When multiple VMs with less virtual cores are combined into virtual clusters, additional virtualization / communication latencies might play a role. On the other hand, an efficiency above 80% is quite noteworthy, considering that native HPC clusters based on Ethernet network technology barely reach an efficiency of 70%. Worth mentioning is also a slight increase in efficiency by a corresponding virtual Central Processing Unit (vCPU) configuration, which ensures that each virtual core is assigned permanently to a fixed physical core.

Especially MPI applications strongly depend on low communication latency. We have run SKaMPI [18], a synthetic MPI benchmark, between two VMs on the same physical node as well as on distributed nodes. In addition, we have performed the same measurement on the native hardware. VMs on the same host are able to communicate with QDR speed directly through the attached *ConnectX-2 HCA*. However, the communication between both hosts is downgraded to DDR speed because of the connection over an IB DDR

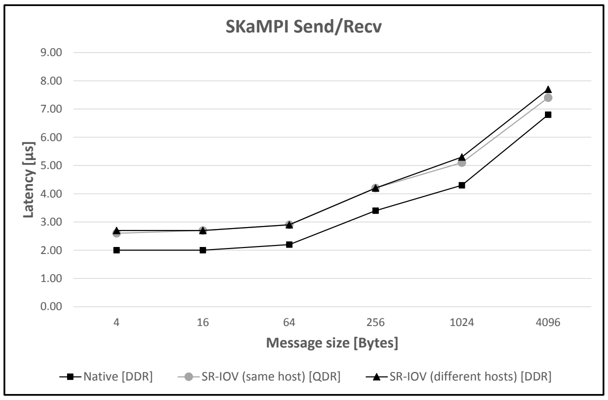

Figure 3. Measured communication latencies in μs.

switch. Figure 3 presents the results. It turns out that the communication latency within virtual cluster increases only slightly compared with the native environment. In summary, it can therefore be said that our prototypic HPC system has the necessary prerequisites to provide an acceptable environment for executing HPC tasks. A more detailed evaluation for SR-IOV performance in combination with the IB interconnect is done by Panda et al. [19].

VI. Conclusion

An IaaS model for HPC based on cloud computing allows users to request elastic virtual HPC clusters as on-demand resources. Compared to native environments, virtual resources are provided to users with administrative privileges and billed according to the *pay-as-you-go* principle. We adapt this architecture model for the operation of an HPC cloud based on the Openstack framework and the IB cluster interconnect. Therefore, we use the SR-IOV specification for PCI devices to manage and utilize the available HPC infrastructure more efficiently. Users get independent isolated clusters with better Linpack performance efficiency and communication latency compared to virtualized and native cluster infrastructure based on Ethernet.

VII. Outlook

Our next steps include extending the infrastructure test bed and extensive testing of typical HPC applications to provide a more detailed evaluation of performance impacts within virtualized HPC environments. Furthermore, we are looking forward to compare our findings with the announced HPC IaaS resources, which will be provided by Microsoft Windows Azure this year. New compute intensive VM instances of type A8/A9 will also support the IB interconnect for running HPC tasks. However, at the moment we have no information about the specific deployment technology.

Currently, live migration is an important mechanism for cloud computing services as it eases cluster management and allows load balancing and fault tolerance. But it is still challenging to achieve with high-speed network interconnects like IB, as these technologies usually maintain their connection state within the network device hardware. Tasoulas [20] presented a first prototypic implementation of live migration over SR-IOV enabled IB devices. We will pursue this research field and try to adjust our architecture if possible.

References

[1] P. Mell and T. Grance, “The nist definition of cloud computing,” National Institute of Standards and Technology, vol. 53, no. 6, 2009, p. 50.

[2] C. Baun, M. Kunze, J. Nimis, and S. Tai, “Cloud computing: Web-based dynamic it services,” 2011.

[3] V. Mauch, M. Kunze, and M. Hillenbrand, “High performance cloud computing,” Future Generation Computer Systems, vol. 29, no. 6, 2013, pp. 1408–1416.

[4] J. Napper and P. Bientinesi, “Can cloud computing reach the top500?” in Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop. ACM, 2009, pp. 17–20.

[5] G. Wang and T. E. Ng, “The impact of virtualization on network performance of amazon ec2 data center,” in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–9.

[6] C. Evangelinos and C. Hill, “Cloud computing for parallel scientific hpc applications: Feasibility of running coupled atmosphere-ocean climate models on amazon’s ec2,” ratio, vol. 2, no. 2.40, 2008, pp. 2–34.

[7] A. Gupta and D. Milojicic, “Evaluation of hpc applications on cloud,” in Open Cirrus Summit (OCS), 2011 Sixth. IEEE, 2011, pp. 22–26.

[8] P. Zaspel and M. Griebel, “Massively parallel fluid simulations on amazon’s hpc cloud,” in Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on. IEEE, 2011, pp. 73–78.

[9] N. Regola and J.-C. Ducom, “Recommendations for virtualization technologies in high performance computing,” in Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on. IEEE, 2010, pp. 409–416.

[10] InfiniBand Architecture Specification Volume 1, Release 1.2.1. InfiniBand Trade Association, 2007.

[11] J. Liu, W. Huang, B. Abali, and D. K. Panda, “High performance vmm-bypass i/o in virtual machines.” in USENIX Annual Technical Conference, General Track, 2006, pp. 29–42.

[12] B.-A. Yassour, M. Ben-Yehuda, and O. Wasserman, “Direct device assignment for untrusted fully-virtualized virtual machines,” Tech. rep., IBM Research Report H-0263, Tech. Rep., 2008.

[13] P. SIG, “Single root i/o virtualization and sharing specification, revision 1.0,” 2008.

[14] M. Hillenbrand, V. Mauch, J. Stoess, K. Miller, and F. Bellosa, “Virtual infiniband clusters for hpc clouds,” in Proceedings of the 2nd International Workshop on Cloud Computing Platforms. ACM, 2012, p. 9.

[15] M. Hillenbrand. Towards virtual infiniband clusters with network and performance isolation. System Architecture Group, Karlsruhe Institute of Technology (KIT), Germany. Last Access: February, 2015. [Online]. Available: http://os.ibds.kit.edu/ [retrieved: June, 2011]

[16] H. W. Meuer, “The top500 project. looking back over 15 years of supercomputing experience,” PIK-Praxis der Informationsverarbeitung und Kommunikation, vol. 31, no. 2, 2008, pp. 122–132.

[17] J. Dongarra, P. Luszczek, and A. Petitet, “The linpack benchmark: past, present and future,” Concurrency and Computation Practice and Experience, vol. 15, no. 9, 2003, pp. 803–820.

[18] R. Reussner, P. Sanders, L. Prechelt, and M. Müller, “Skampi: A detailed, accurate mpi benchmark,” in Recent advances in parallel virtual machine and message passing interface. Springer, 1998, pp. 52–59.

[19] 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, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on. IEEE, 2013, pp. 385–392.

[20] V. Tasoulas. Prototyping live migration with sr-iov supported infiniband hcas. HPC Advisory Council 2013. Last Access: February, 2015. [Online]. Available: http://www.hpcadvisorycouncil.com/events/2013/Spain-Workshop/pdf/4_Simula.pdf [retrieved: September, 2013]