

Overview of Virtualization in HPC



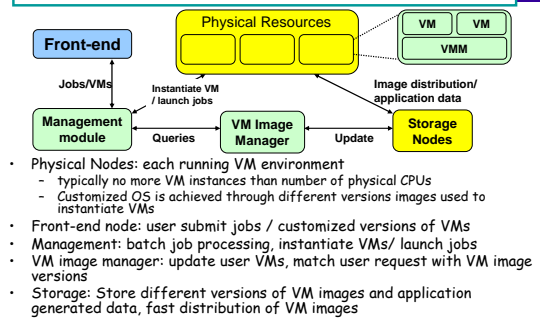
Why Target Virtualization?

- **Ease of management**
 - Virtualized clusters
 - VM migration – deal with system upgrade/failures
- **Customized OS**
 - Light-weight OS: No wide adoption due to management difficulties
 - VM makes these techniques possible
- **System security & productivity**
 - Users can do 'anything' in VM, in the worst case crash a VM, not the whole system

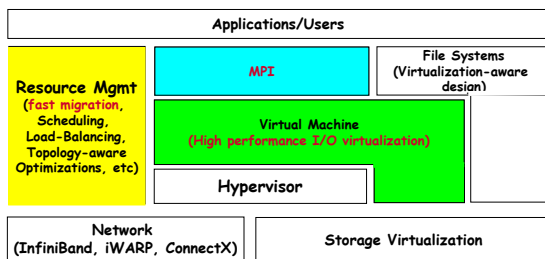
Challenges

- **Performance overhead**
 - CPU and memory
 - HPC applications are highly CPU intensive and spend most of the time in user space
 - Modern VM technologies achieve high performance by executing most instructions natively on host CPUs
 - I/O
 - Bigger problem since the hypervisor lies in the critical path
- **Migration of modern OS-bypass network devices**
- **Management framework to take advantages of VM technology for HPC**

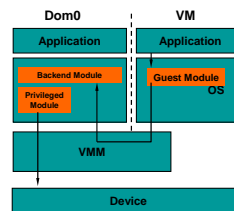
An example for VM-based Computing Environment



Virtual Machine Based HPC: A Roadmap



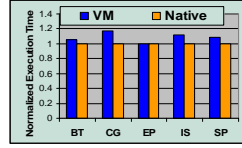
Current I/O Virtualization Approaches



- **I/O in VMM (e.g. VMware ESX Server)**
 - Device drivers are hosted in the VMM
 - I/O operations always trap into the VMM
 - The VMM ensures safe device sharing among VMs
- **I/O in a special VM**
 - Device drivers are hosted in a special (privileged) VM
 - I/O operations always involve the VMM and the special VM
 - Examples: Xen and VMware Workstation

Problem with Current I/O Virtualization

- Performance
 - Every I/O operation involves the VMM and/or another VM
 - VMM may become a performance bottleneck
 - Using a special VM results in expensive context switches between different VMs
 - Undesirable for high end systems, especially those used in high performance computing (HPC)



	Dom0	VMM	DomU
CG	16.6%	10.7%	72.7%
IS	18.1%	13.1%	68.8%
EP	00.6%	00.3%	99.0%
BT	06.1%	04.0%	89.9%
SP	09.7%	06.5%	83.8%

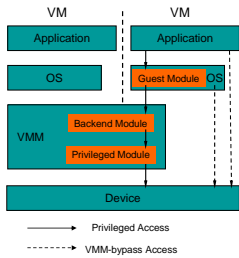
7

VMM-bypass I/O: Basic Ideas

- VMM-bypass
 - Direct HW access for time-critical I/O operations
 - VMM involved for setup and management
- Extending the concept of OS-bypass in the context of VM environments
 - Requires intelligent I/O adapters
- Para-virtualization
 - Does not emulate the same hardware interface in guest VMs
 - But maintains the same high-level interfaces used by Oses and applications in guest VMs

8

From OS-bypass to VMM-bypass



- Guest modules in guest VMs handle setup and management operations (privileged access)
 - Guest modules communicate with backend modules in VMM to get jobs done
 - The original privileged module can be reused
- Once things are setup properly, devices can be accessed directly from guest VMs (VMM-bypass access)
 - Either from the OS kernel or applications
- Backend and privileged modules can also reside in a special VM

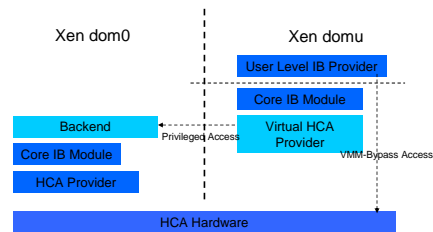
9

Xen-IB: InfiniBand Virtualization Driver for Xen

- Follows Xen split driver model
- Presents virtual HCAs to guest domains
 - Para-virtualization
- Two modes of access:
 - Privileged access
 - OS involved
 - Setup, resource management and memory management
 - OS/VMM-bypass access
 - Directly done in user space/guest VM
 - Maintains high performance of InfiniBand

10

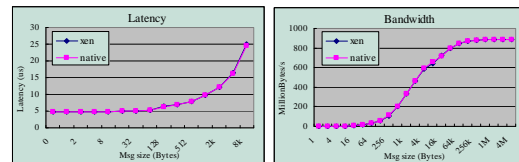
Xen-IB Basic Structure



J. Liu, W. Huang, B. Abali, D. K. Panda, High Performance VMM-Bypass I/O in Virtual Machines, USENIX Annual Technical Conference (USENIX'06), May, 2006

11

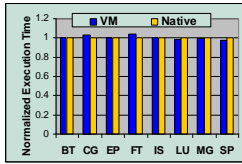
MPI Latency and Bandwidth (MVAPICH)



- Only VMM Bypass operations are used
- Xen-IB performs similar to native InfiniBand
- Numbers taken with MVAPICH

12

HPC Benchmarks (NAS)



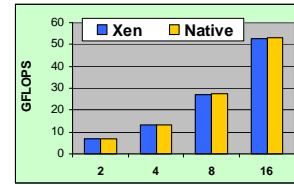
	Dom0	VMM	DomU
BT	0.4%	0.2%	99.4%
CG	0.6%	0.3%	99.0%
EP	0.6%	0.3%	99.3%
FT	1.6%	0.5%	97.9%
IS	3.6%	1.9%	94.5%
LU	0.6%	0.3%	99.0%
MG	1.8%	1.0%	97.3%
SP	0.3%	0.1%	99.6%

- NAS Parallel Benchmarks achieve similar performance in VM and native environment (8x2)

-J. Liu, W. Huang, B. Abali, D. K. Panda. High Performance VMM-Bypass I/O in Virtual Machines, *USENIX Annual Technical Conference (USENIX06)*, May, 2006
 -W. Huang, J. Liu, B. Abali, D. K. Panda. A Case for High Performance Computing with Virtual Machines, *ACM International Conference on Supercomputing (ICS'06)*, June, 2006

13

HPC Benchmarks (HPL)



- HPL: achievable GFLOPS in VM and Native environment is within 1% difference

14

Challenges of Migrating InfiniBand

- Location dependent resources (cannot migrate with VMs):
 - LIDs, QPNs, CQNs
- User level communication:
 - Can be caching handles (memory keys, QPNs, ..) anywhere
 - Hard to suspend communication from kernel
- Hardware managed connection state:
 - Cannot easily achieve reliability during migration

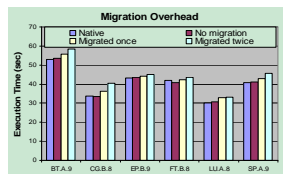
15

Key Ideas of Nomad: Migration support for InfiniBand in VM environment

- Namespace Virtualization:
 - Virtualize all location dependent resources, such as LIDs, QPNs, CQNs, memory keys, etc.
 - Special handling for memory keys to achieve low overhead in critical path
 - Intercept communication calls at libmthca to achieve application transparency
- Coordination:
 - libmthca coordinates during migration to suspend/resume communication
 - Push QPN, LIDs, memory keys updates to connected peers

16

Overhead of Migration

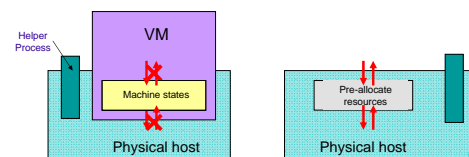


- Each migration costs 0.5 to 3 seconds, depending on the computing and communication patterns
- One process per node (dual processors) to reduce Xen overhead

W. Huang, J. Liu, M. Koop, B. Abali, D. K. Panda. Nomad: Migrating OS-bypass Networks in Virtual Machines, *The Third ACM/USENIX Conference on Virtual Execution Environment (VEE'07)*, June, 2007

17

Optimizing VM migration through RDMA

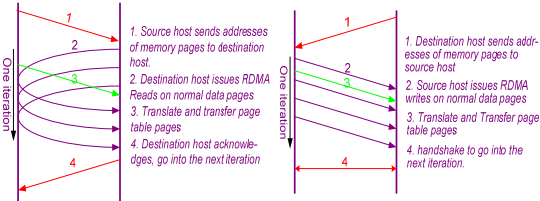


Live VM migration:

- Step 1: Pre-allocate resource on target host
- Step 2: Pre-copy machine states for multiple iterations
- Step 3: Suspend VM and copy the latest updates to machine states
- Step 4: Restart VM on the new host

18

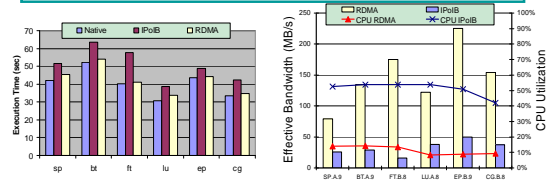
Efficient VM migration using RDMA



- Moving machine states over RDMA:
 - High bandwidth with RDMA capable interconnects
 - Less CPU utilization
 - One sided protocol: only one party involved
 - Migration protocols can be based on either RDMA read or write

19

Fast Migration over RDMA



- Disable one physical CPU on the nodes
- Migration overhead with IPoB drastically increases
- RDMA achieves higher migration performance with less CPU usage

W. Huang, Q. Gao, J. Liu, D. K. Panda. High Performance Virtual Machine Migration with RDMA over Modern Interconnects. *IEEE Conference on Cluster Computing (Cluster'07)*, September 2007 (Selected as a Best Paper)

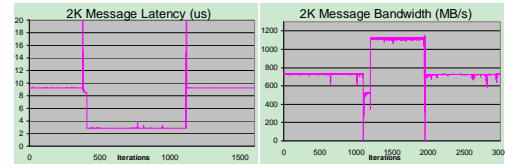
20

MPI in Virtual Machine Environment

- MPI libraries supporting OFA verbs should benefit transparently from VMM-bypass I/O and the migration support
- Extensions: allow efficient inter-VM communication
- Design issues:
 - Shared memory communication for user processes not in the same OS
 - Switch communication method when VM migrates
 - Hide details in MPI library

21

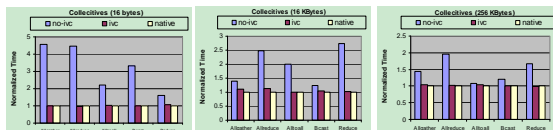
Efficient inter-VM communication through MPI



- MVAPICH2-ivc automatically switches to IVC whenever the target peers are on the same physical nodes
- Above two graphs show decreased latency and increased bandwidth when two processes in separate VMs are migrated to the same physical nodes

22

Efficient inter-VM communication through MPI

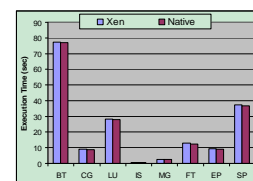


- With inter-VM communication, MVAPICH2-ivc largely closes the gap between native and VM based environments
- Results collected on 8-core systems using Intel MPI Benchmarks (IMB)

More details will be presented on Tuesday at 11am (System Performance Session): W. Huang, M. Koop, Q. Gao, D. K. Panda, Virtual Machine Aware Communication Libraries for High Performance Computing. *Supercomputing (SC07)*, November 2007 (Best Student Paper Finalist)

23

Evaluation on Larger Cluster



- Numbers taken on 64 nodes (dual processor) using NAS class C
- Overhead is marginal in most cases
- Some gap (FT, SP) is due to the optimized SMP performance of MVAPICH2. We will optimize the Xen case in future

24