

# Making Wide-Area, Multi-Site MPI Feasible Using Xen VM

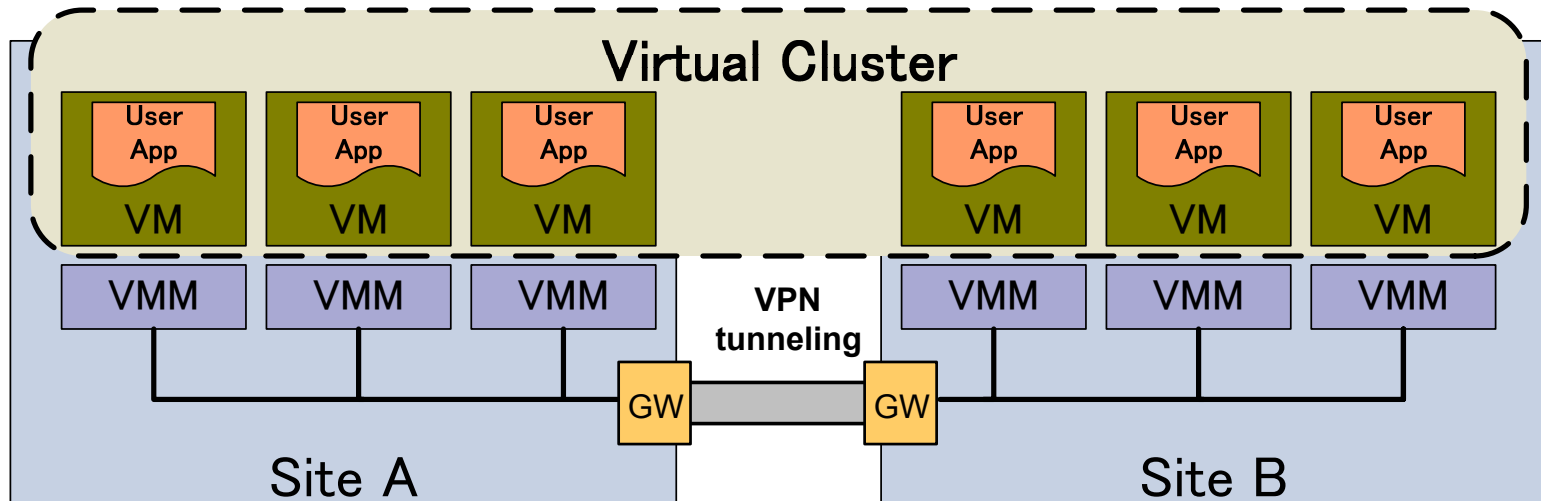
Masaki Tatzono  
Naoya Maruyama  
Satoshi Matsuoka

*Tokyo Institute of Technology*

# Background

- Target environments
  - Computational Grids consisting of multiple clusters distributed over the Internet
    - More computing power than each single cluster
- Goal
  - To optimize the performance of multiple MPI jobs with different characteristics on the computational Grids
- Challenge
  - Exploiting the potentially-available large-scale resources on Grids are considered impractical for MPI programs
    - Software and hardware heterogeneity
    - Narrow inter-site links

# Our Approach



- A virtual cluster for each MPI job
  - Organizes multiple clusters into a single virtual cluster
  - Hides software heterogeneity
- Dynamic migration of MPI programs
  - E.g., starts with globally-distributed VMs; then migrates the VMs to a single physical cluster if the cluster becomes idle

# Issues in MPI on Virtual Clusters

- Heterogeneity in underlying hardware
  - CPU speed, network performance
  - Makes load balancing even harder
- Virtualization overhead
  - Virtual machine monitors
  - Overlay networks
- Inter-site links
  - Higher latency, lower bandwidth
  - Tend to become bottlenecks

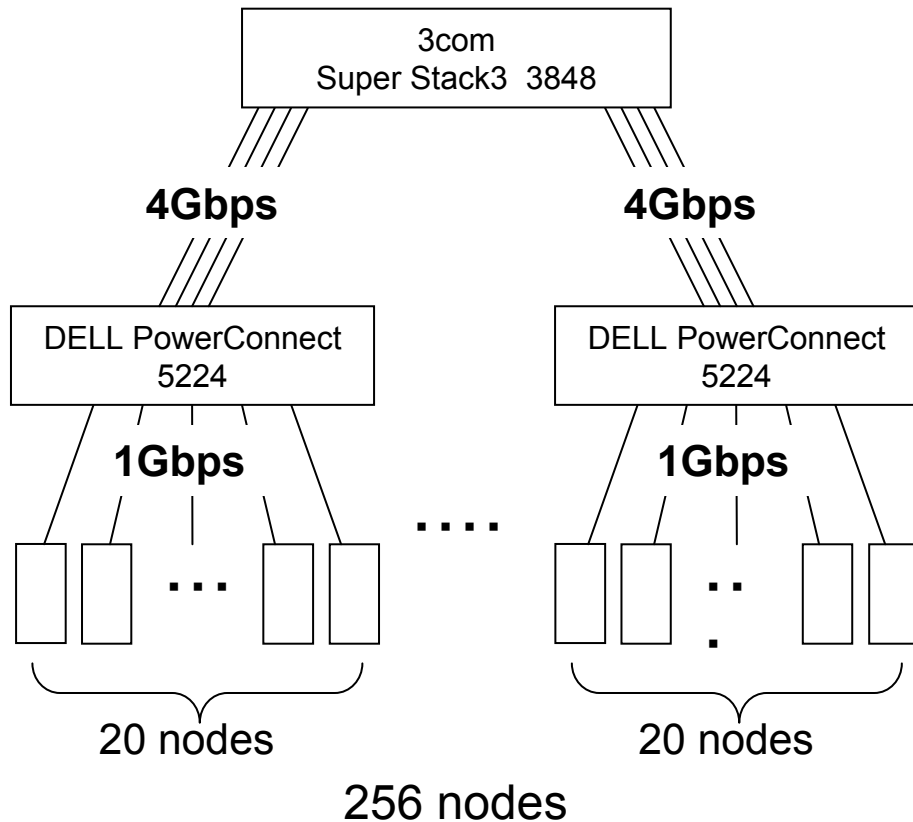
The focus of this work:

**Evaluation of the implications to MPI performance.**

# Evaluation Methodology

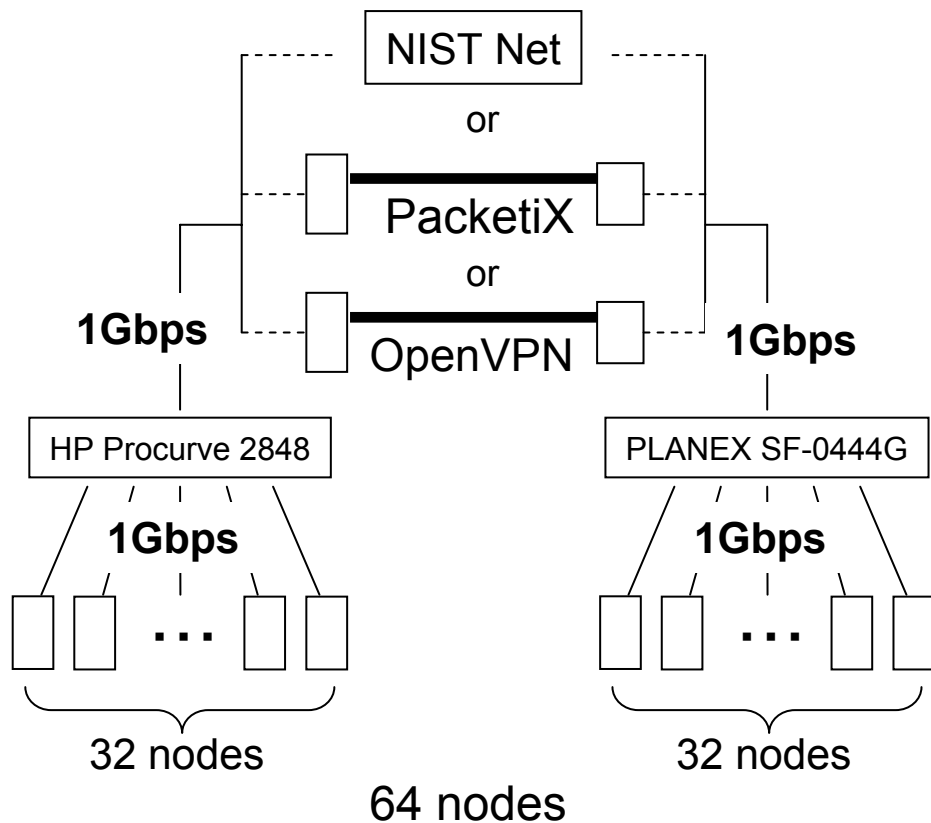
- Purpose
  - To evaluate the network and MPI performance of single-site and multi-site virtual clusters
- Evaluation testbeds
  - Single-site testbed
  - Two-site testbeds
- Evaluation programs
  - Network performance
    - NetPIPE 3.6.2 TCP/IP point-to-point benchmark
  - MPI performance
    - MPICH-1.2.7p1
    - NPB 3.1 Class B

# The Single-Site Testbed



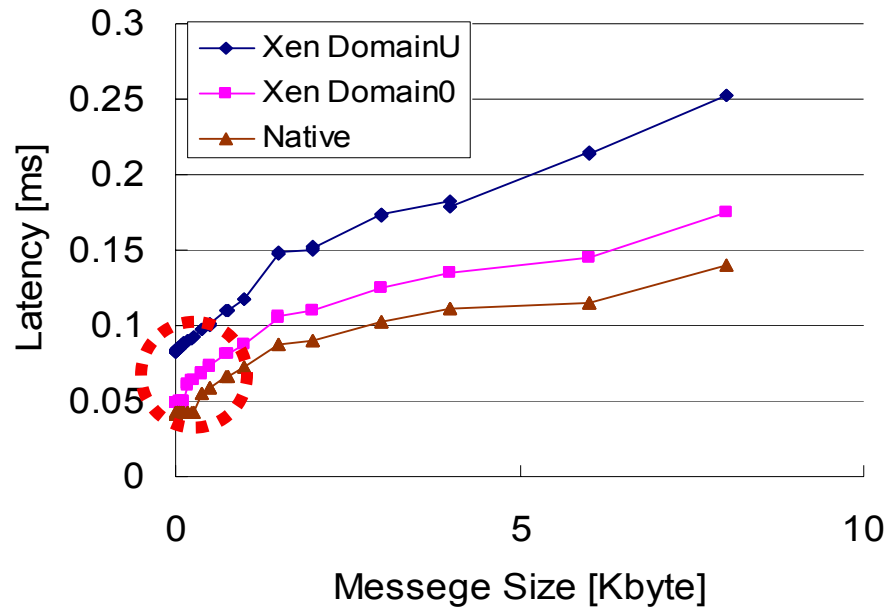
- A Xen v3.0.2-based virtual cluster hosted in a single physical cluster
- Uses 128 nodes from the PrestoIII cluster
  - Dual Opteron 242 (1.6GHz)
  - 2GB of RAM
  - Interconnected with a fat-tree topology of Gigabit Ethernet switches

# The Two-Site Testbeds

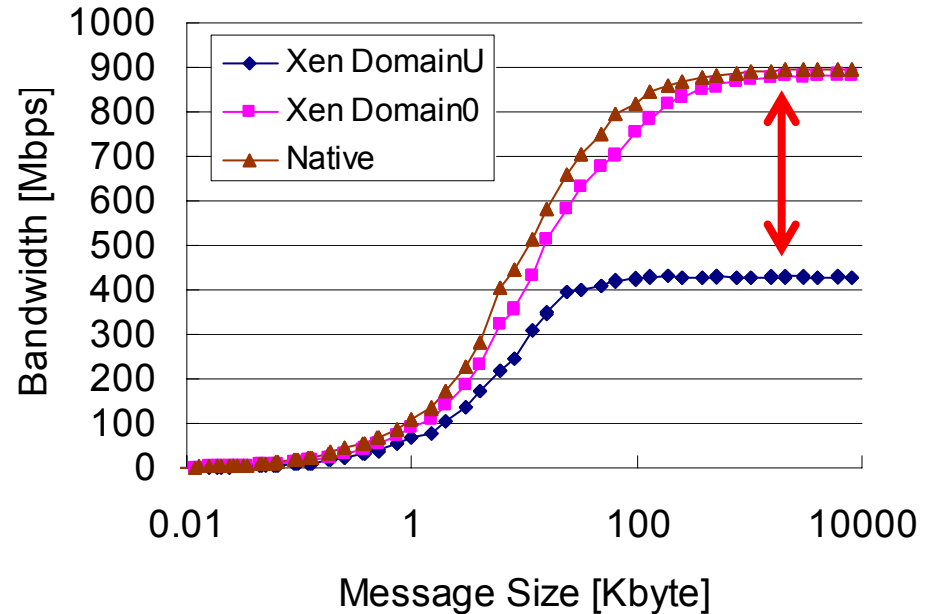


- VPNs consisting of two different networks
  - Uses OpenVPN v2.0 and PacketiX VPN v2.0 as VPN implementations
  - The OpenVPN gateways: Opteron 242 with 2GB of RAM, running Linux 2.6.12.6
  - The PacketiX gateways: Pentium4 662 with 1GB of RAM, running WinXP
- A two-network testbed with software-emulated network latencies injected
  - Uses NIST Net v2.0.12 to insert network latency
  - The NIST Net machine: Athlon MP 2000+, 1GB of RAM

# The Single-Site Testbed: Network Performance



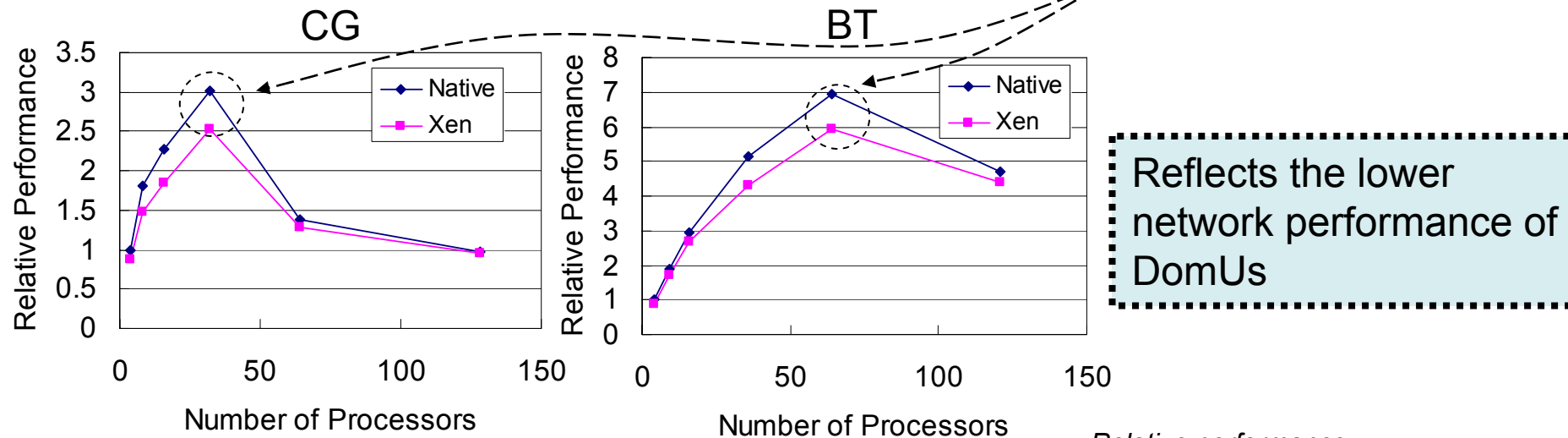
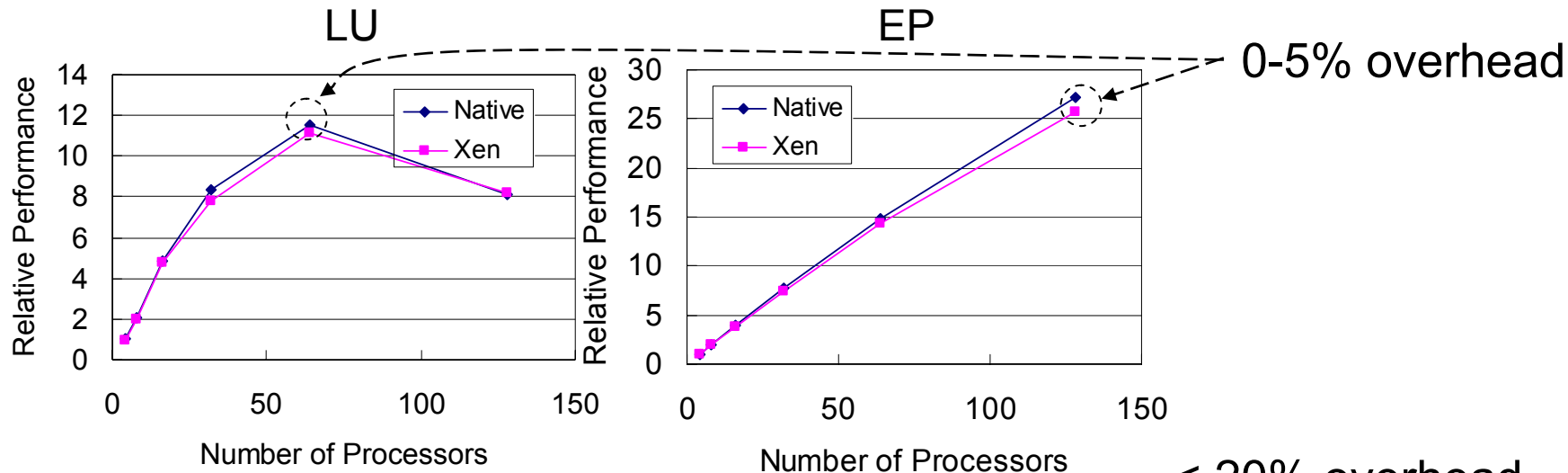
- Minimum latency  
Native: 0.04 ms  
Domain0: 0.05 ms  
DomainU: 0.08 ms



- Maximum bandwidth  
Native: 896 Mbps  
Domain0: 883 Mbps  
DomainU: 430 Mbps

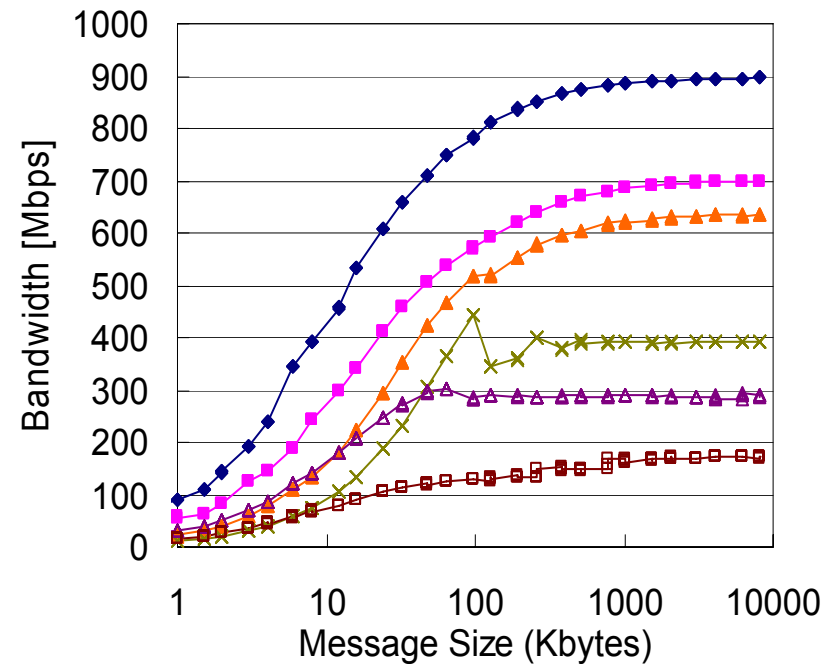
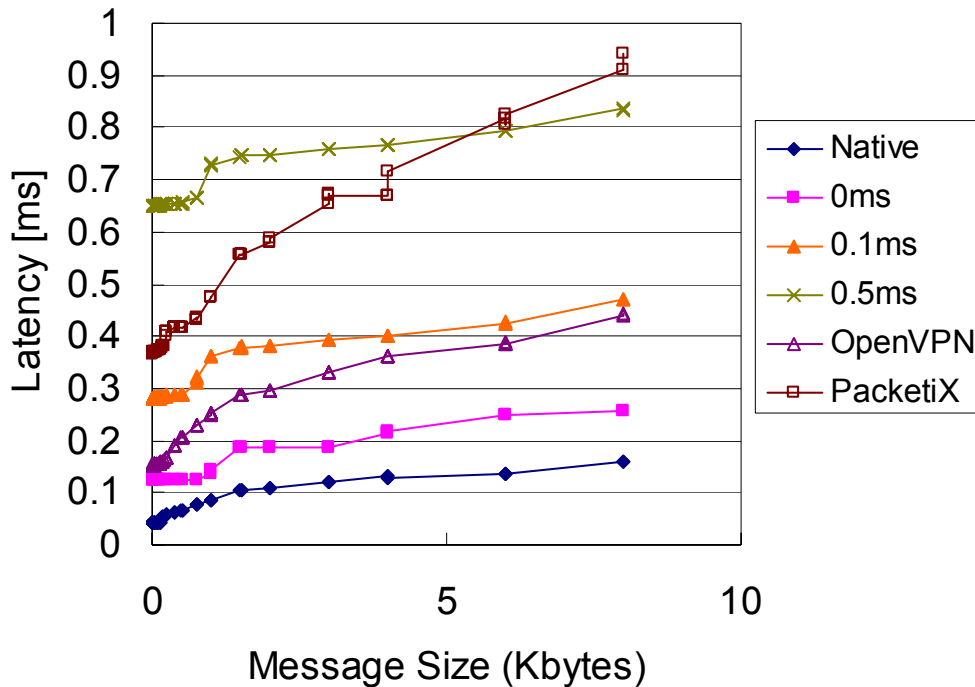
• latency: 200%  
• bandwidth: 50%

# The Single-Site Testbed: MPI Performance



*Relative performance:*  
performance improvement compared to the 4-PE case

# The Two-Site Testbeds: Network Performance



- Latency

- Native: 0.04 ms
- OpenVPN: 0.15 ms
- PacketiX: 0.36 ms

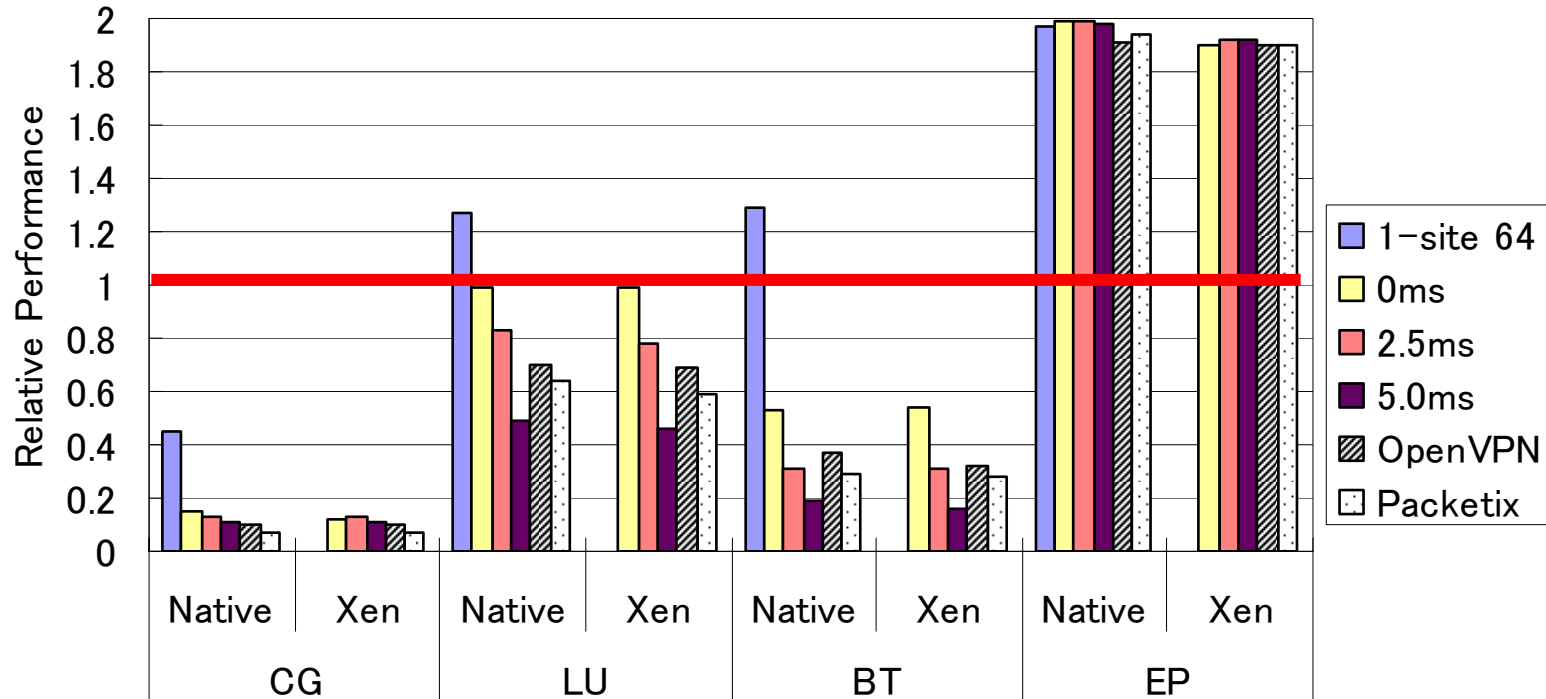
- Maximum bandwidth

- Native: 896 Mbps
- OpenVPN: 290 Mbps
- PacketiX: 170 Mbps

The CPUs of the gateway machines are saturated in large message sizes

# The Two-Site Testbeds: MPI Performance

*Relative performance: improvement compared to the single-site 32-node cluster*



- Compares the performance of two-site 64-node virtual clusters
- Little difference between native and virtual machines (0-10%)
- **Apps like LU or EP could leverage larger number of nodes**
- Less likely to improve network-intensive apps like CG

# Summary of Experiments

- Effects on MPI performance by the Xen VMM
  - Degradation by 0-20%
    - Nearly 0% in a compute intensive benchmark (LU, EP)
    - Less than 20% in the network intensive program (CG, BT)
  - Largely depends on network usage
  - Acceptable in many cases, considering the benefits of virtualization
- MPI performance on multi-site virtual clusters
  - CPU-intensive programs could exploit more resources made available by virtualization

# Future Directions

- Selective use of multiple physical clusters with VM-based dynamic MPI migration
  - Run and profile a job for a short period of time to identify its characteristics
  - CPU-intensive jobs
    - Performance would not be much affected by VM distribution
  - Network-intensive jobs
    - Should be scheduled to a single cluster
- Topology-aware MPI collective communications
  - e.g., [Kielmann et al., '99], GridMPI
- Load-based processor allocation
  - Allocates faster processors to highly-loaded processes
  - Dynamic reallocation with VM migration

# Conclusion

- Multi-site virtual clusters for high-performance MPI execution on Grids
- Preliminary performance studies toward multi-site, wide-area MPI execution
- Network overhead in Xen v3.0.2 is not negligible for MPI
- Possible approaches to high-performance multi-site MPI execution
  - Topology-aware MPI collective operations
  - Selective use of multiple physical clusters
  - Load-based processor allocation