LOW-LATENCY, HIGH-BANDWIDTH USE CASES FOR NAHANNI / IVSHMEM

Cam Macdonell, Xiaodi Ke, Adam Wolfe Gordon, Paul Lu
University of Alberta
paullu@cs.ualberta.ca

KVM Forum 2011
August 16, 2011

Contents

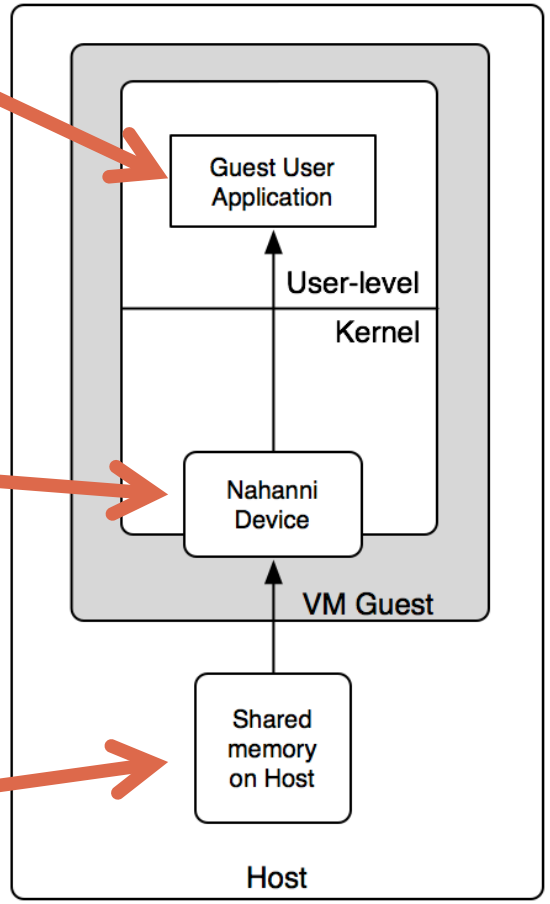
1. For the QEMU/KVM developer
   - Nahanni/ivshmem included as of QEMU 0.13.0, August 2010

2. VMs for Web services
   - Memcached: up to 29% lower inter-VM latencies on workloads

3. VMs for computational science
   - Order-of-magnitude lower latency and higher bandwidth on MPI *microbenchmarks*
   - Up to 30% faster on MPI *application* benchmarks (GAMESS, SPEC MPI2007)

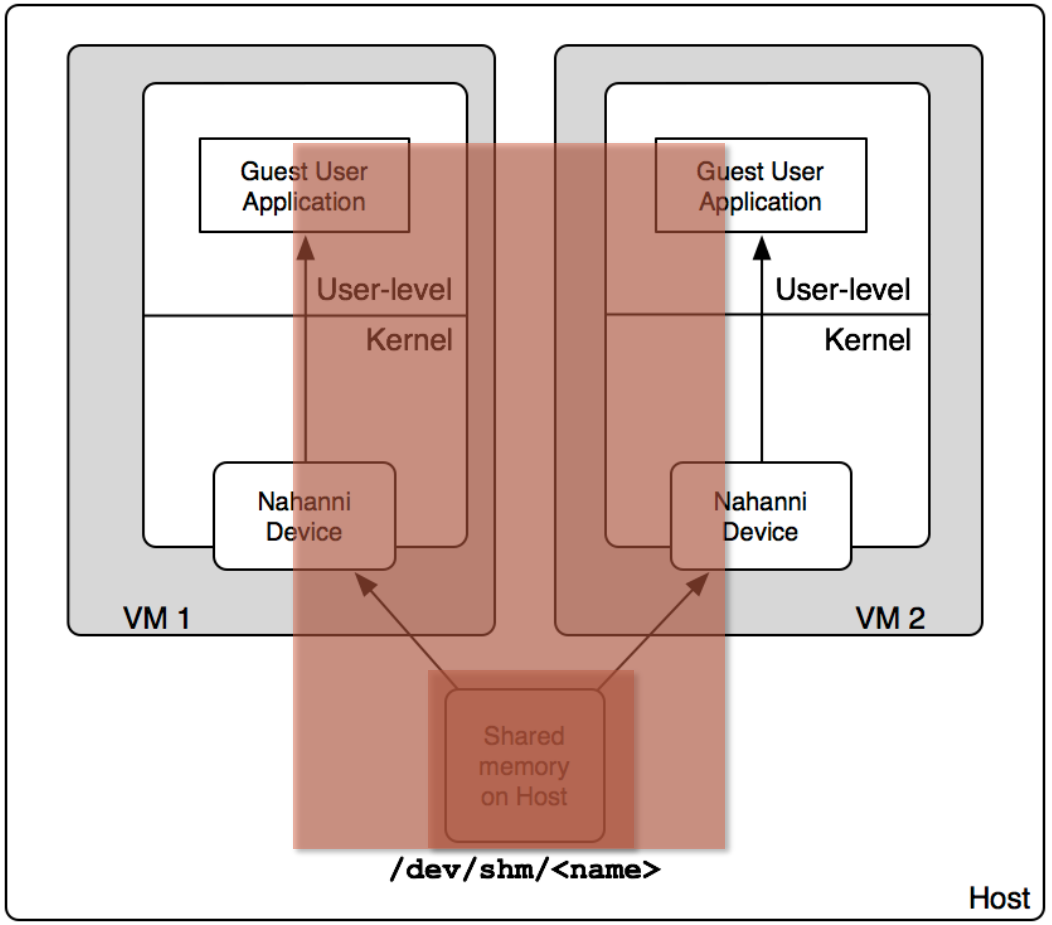
Part 1: What is Nahanni / ivshmem?

- User-level library for *data movement*
  - OS bypass: No guest or host OS involvement
  - Memcached, DDI: pointer-based structured data
  - MPI: message passing, stream data
- Nahanni device looks like a graphics card
  - Guest OS driver (for *initialization only*)
  - No impact on guests that do not load the driver
- New Nahanni virtual PCI device in QEMU
  - `-device ivshmem,shm=<name>,size=1024`
  - Creates POSIX shared memory on host



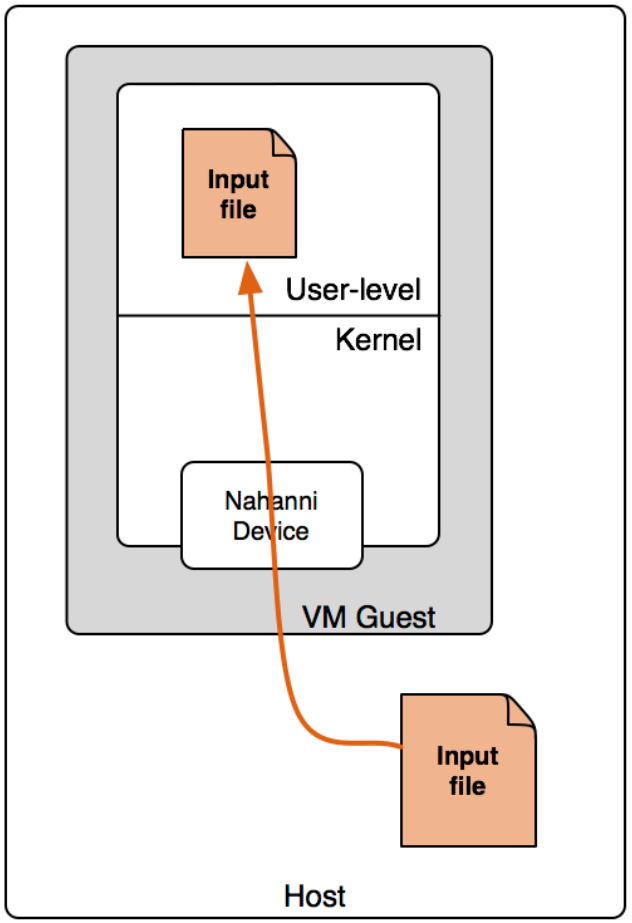
Cam Macdonell's Ph.D.

Host-Guest, Inter-VM Shared Memory



Host-Guest Use Case

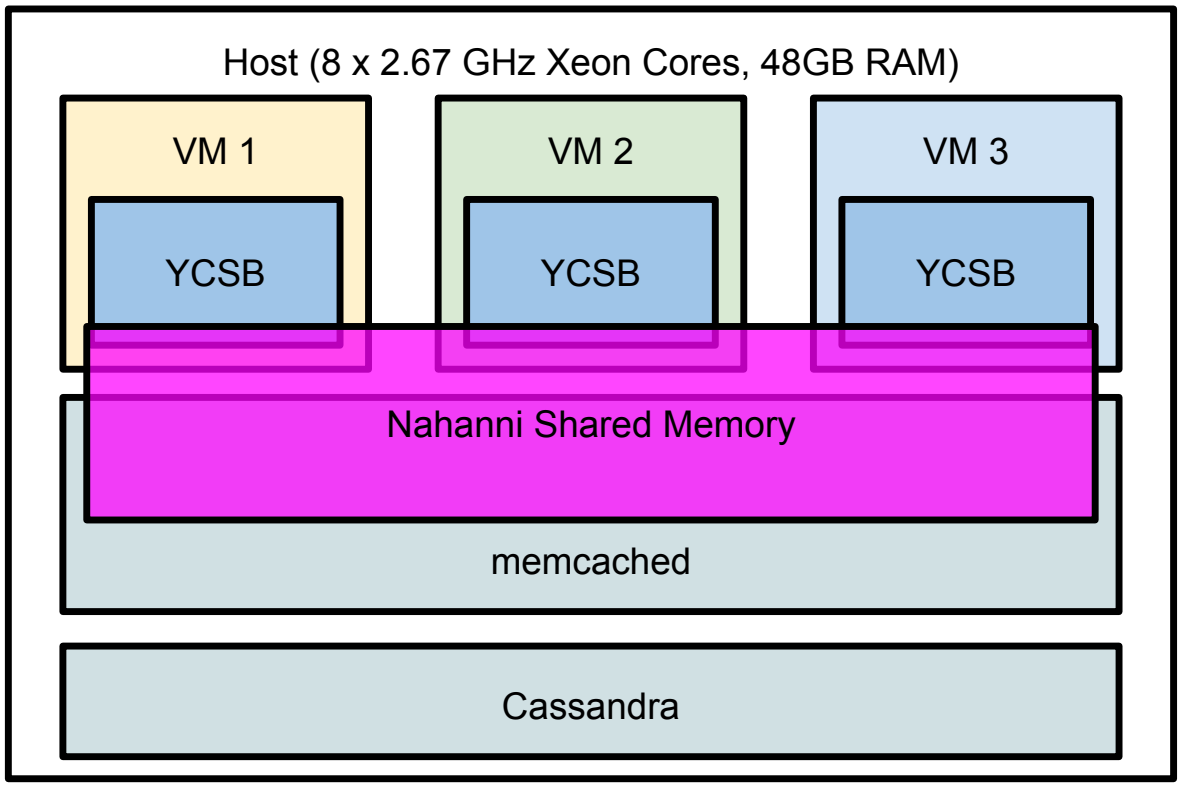
- Copying a file from host into a guest VM
- Nahanni is faster than Netcat, SCP-HPN, 9P



Part 2: Web Services in the Cloud

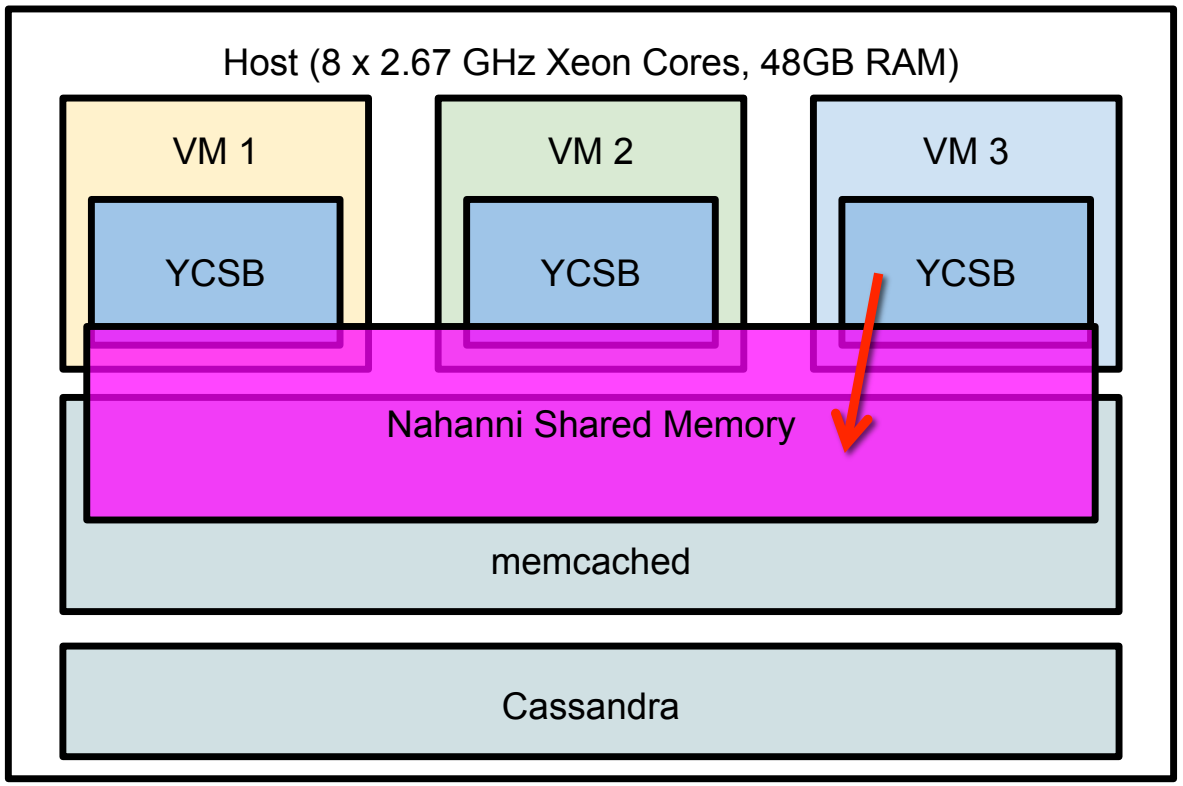
- Many companies use the cloud for their Web servers
  - Reddit and FourSquare are on Amazon EC2
- Memcached is a key-value cache for databases, etc.
  - Used by Facebook, Twitter, others
- **Conclusion: Nahanni reduces look-up latency by 29% on a read-mostly Yahoo Cloud Serving Benchmark (YCSB) workload, for *co-located* VMs**
- M.Sc. thesis and NetDB'11 paper by Adam Wolfe Gordon

Nahanni Memcached + YCSB



- 1 million 1 KB records
- Each VM ran 12 million operations; 36 million in total
- Each VM ran 2,000 operations per second

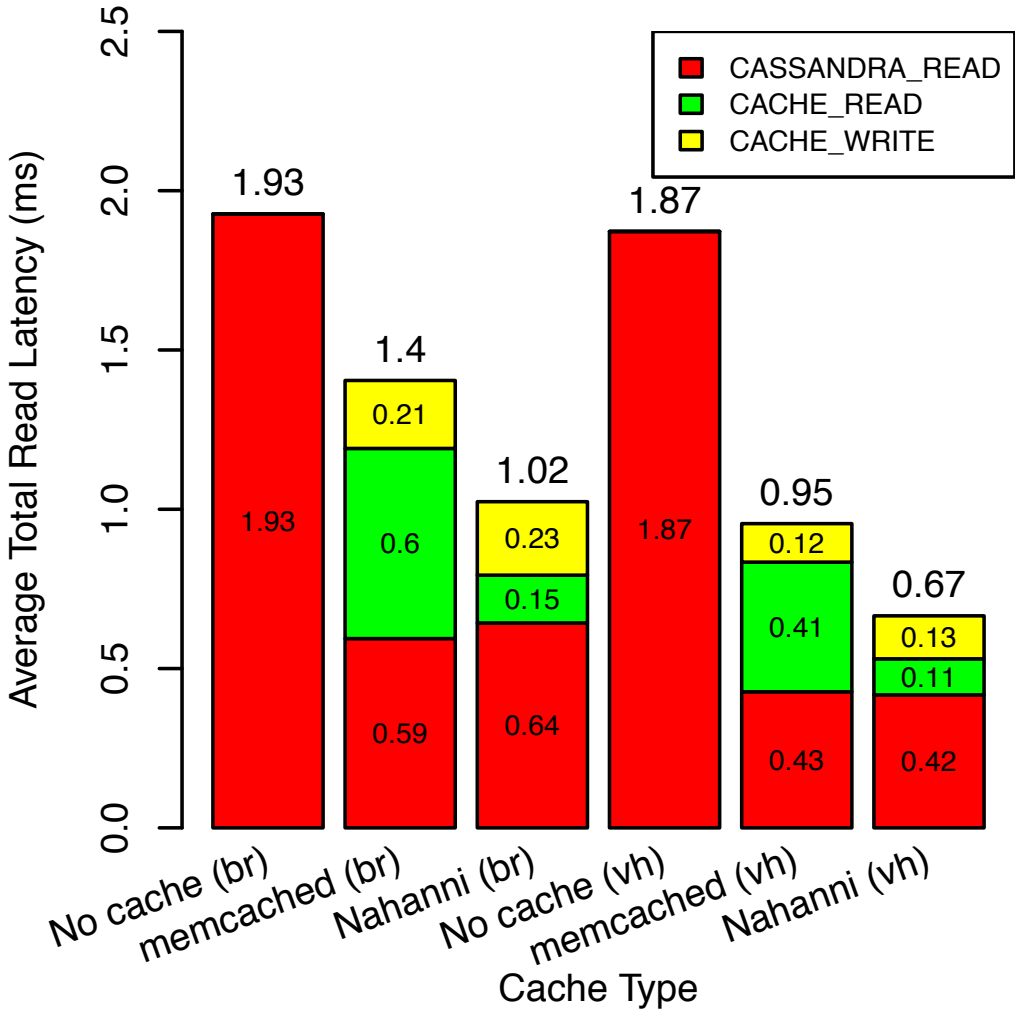
Nahanni Memcached + YCSB



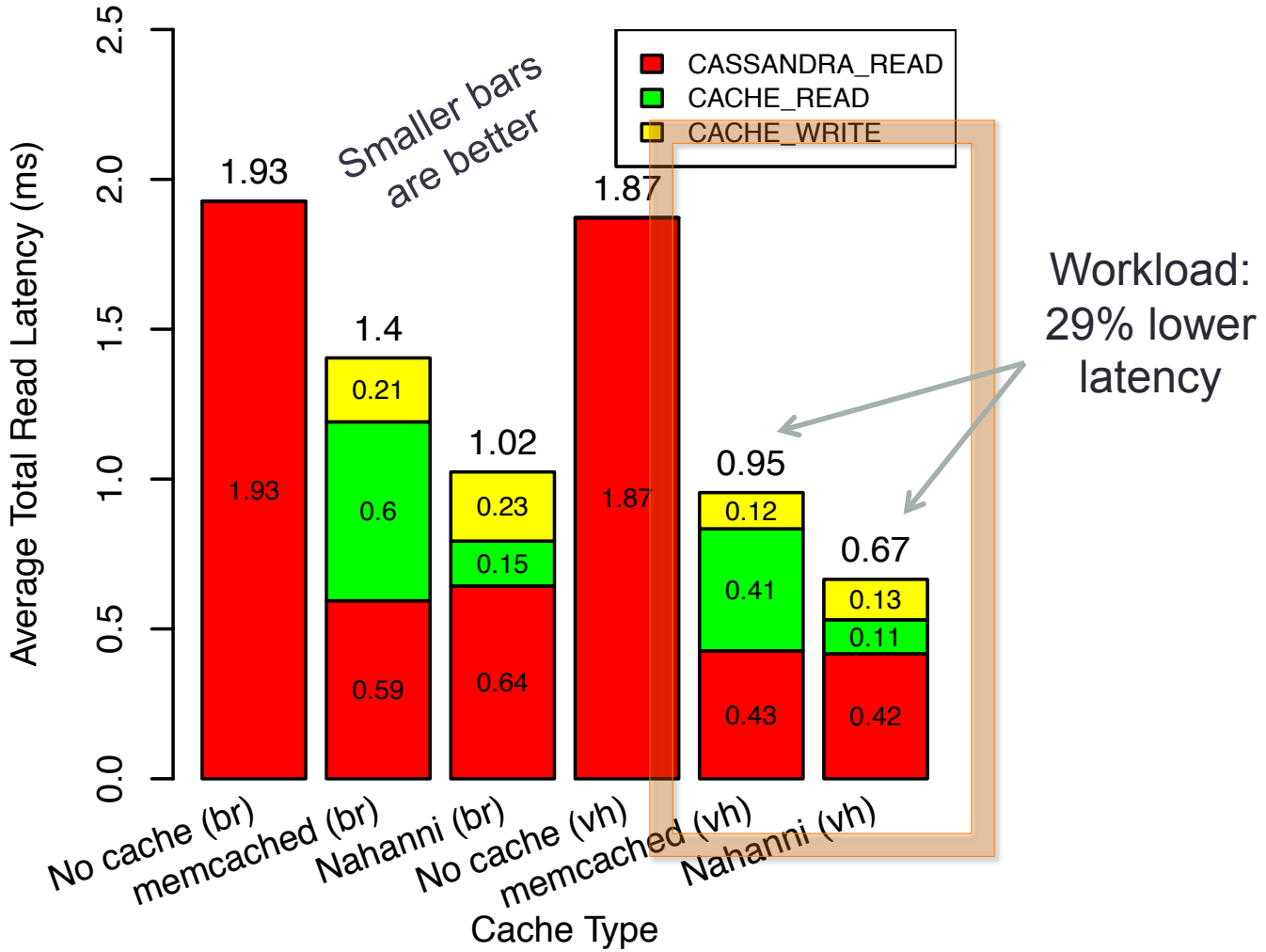
VM 3 can do look-up using pointers and synchronization

- 1 million 1 KB records
- Each VM ran 12 million operations; 36 million in total
- Each VM ran 2,000 operations per second

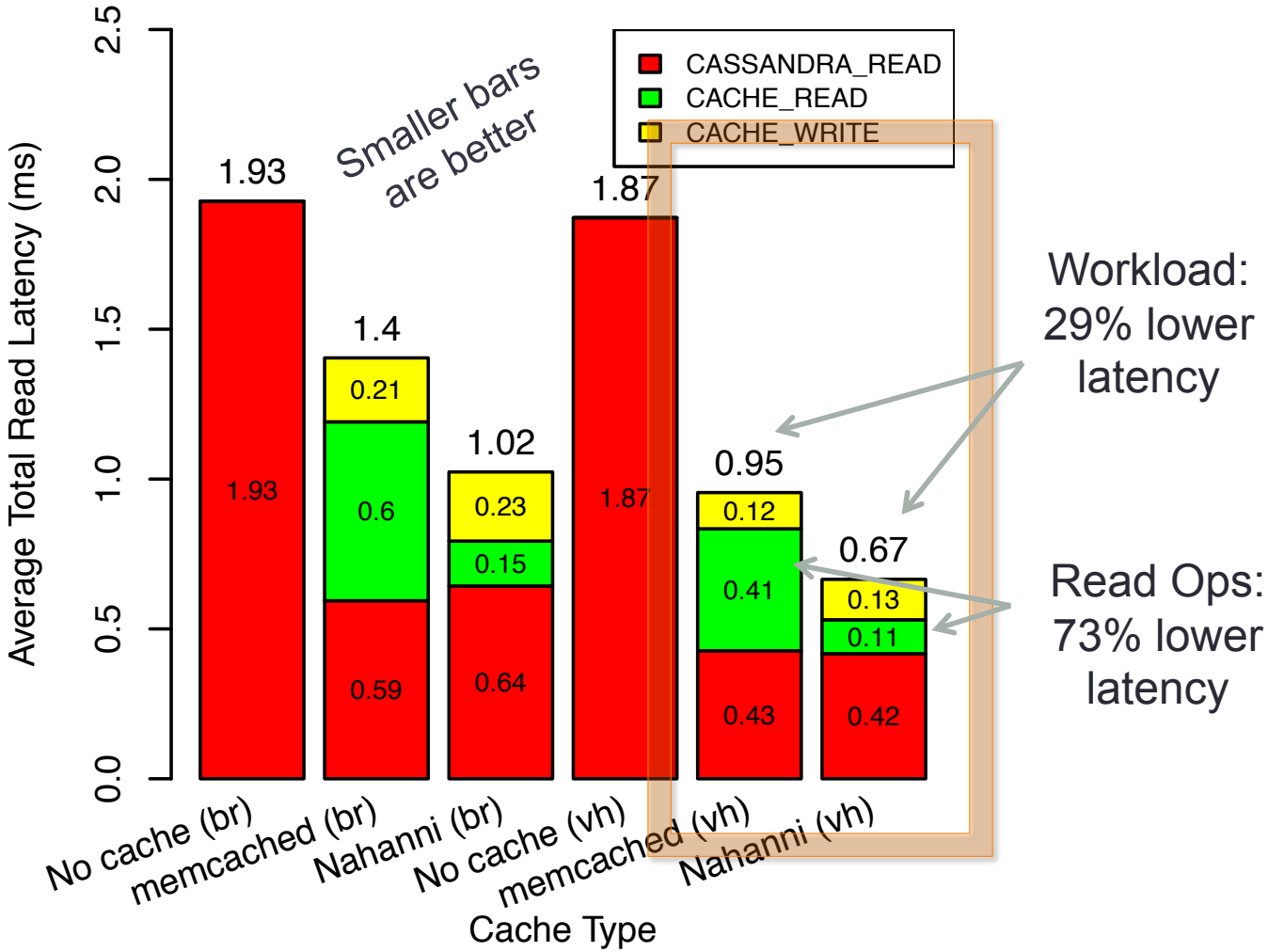
Nahanni vs. virtio/bridging (br) vs. virtio/vhost (vh)



Nahanni vs. virtio/bridging (br) vs. virtio/vhost (vh)



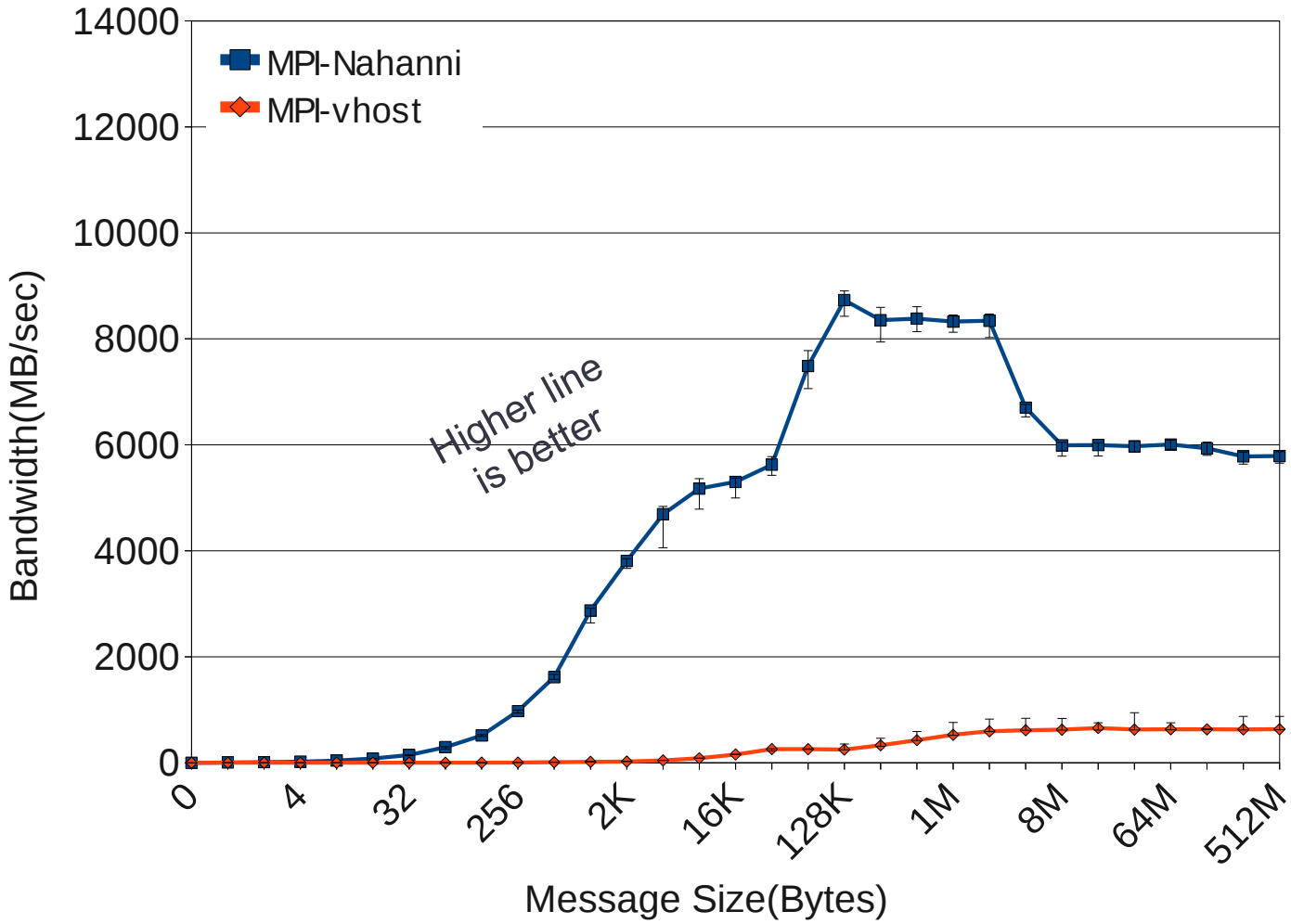
Nahanni vs. virtio/bridging (br) vs. virtio/vhost (vh)



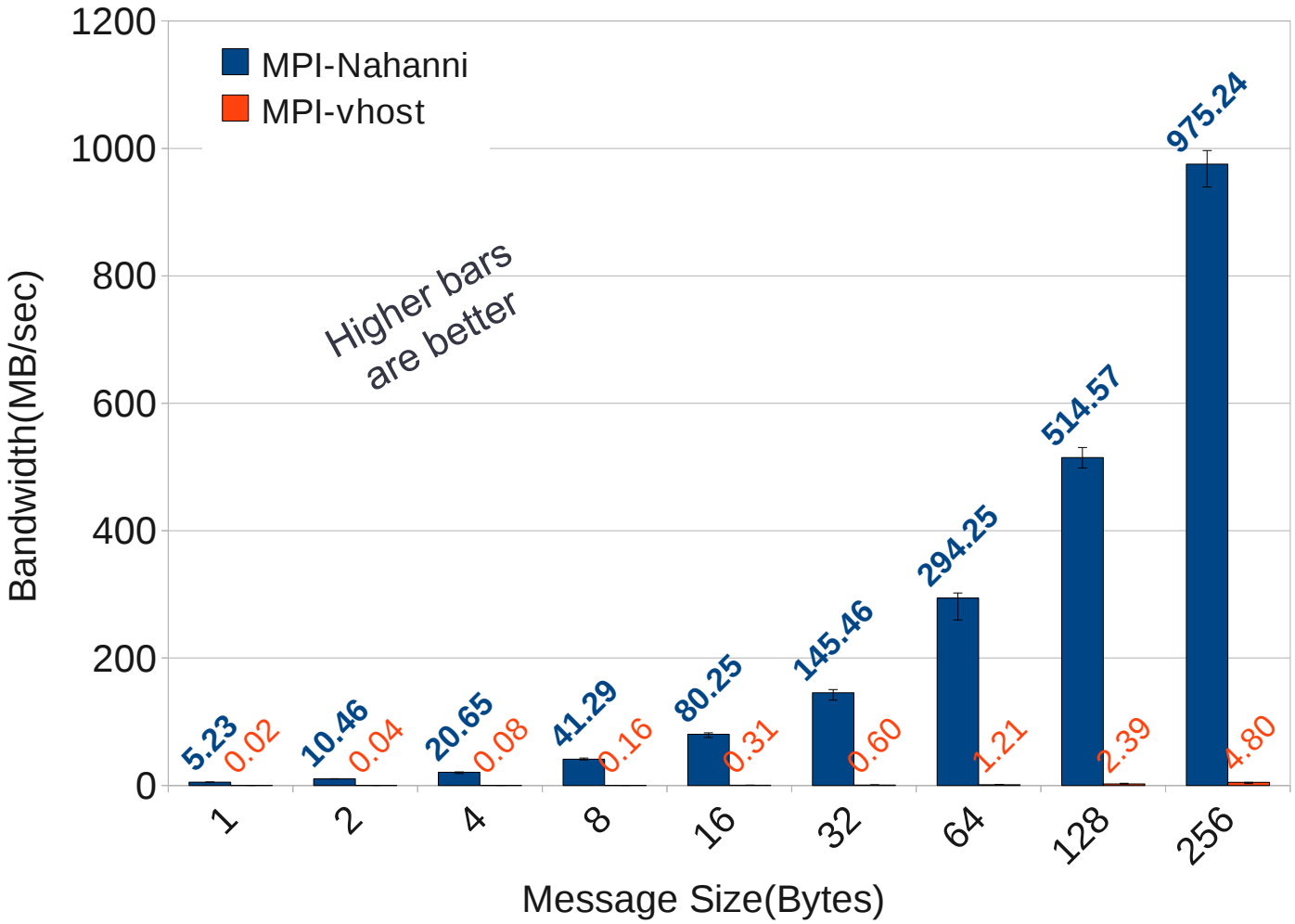
Part 3: Computational Science in VMs

- Your HPC application is likely to have an Message-Passing Interface (MPI) version
- We developed the MPI-Nahanni user-level library
  - Port of MPICH2-Nemesis
  - For co-located VM instances
- **Conclusion: MPI-Nahanni has order-of-magnitude lower latency and higher bandwidth; applications are up to 30% faster**.
- M.Sc. thesis of Xiaodi Ke, Ph.D. thesis of Cam Macdonell

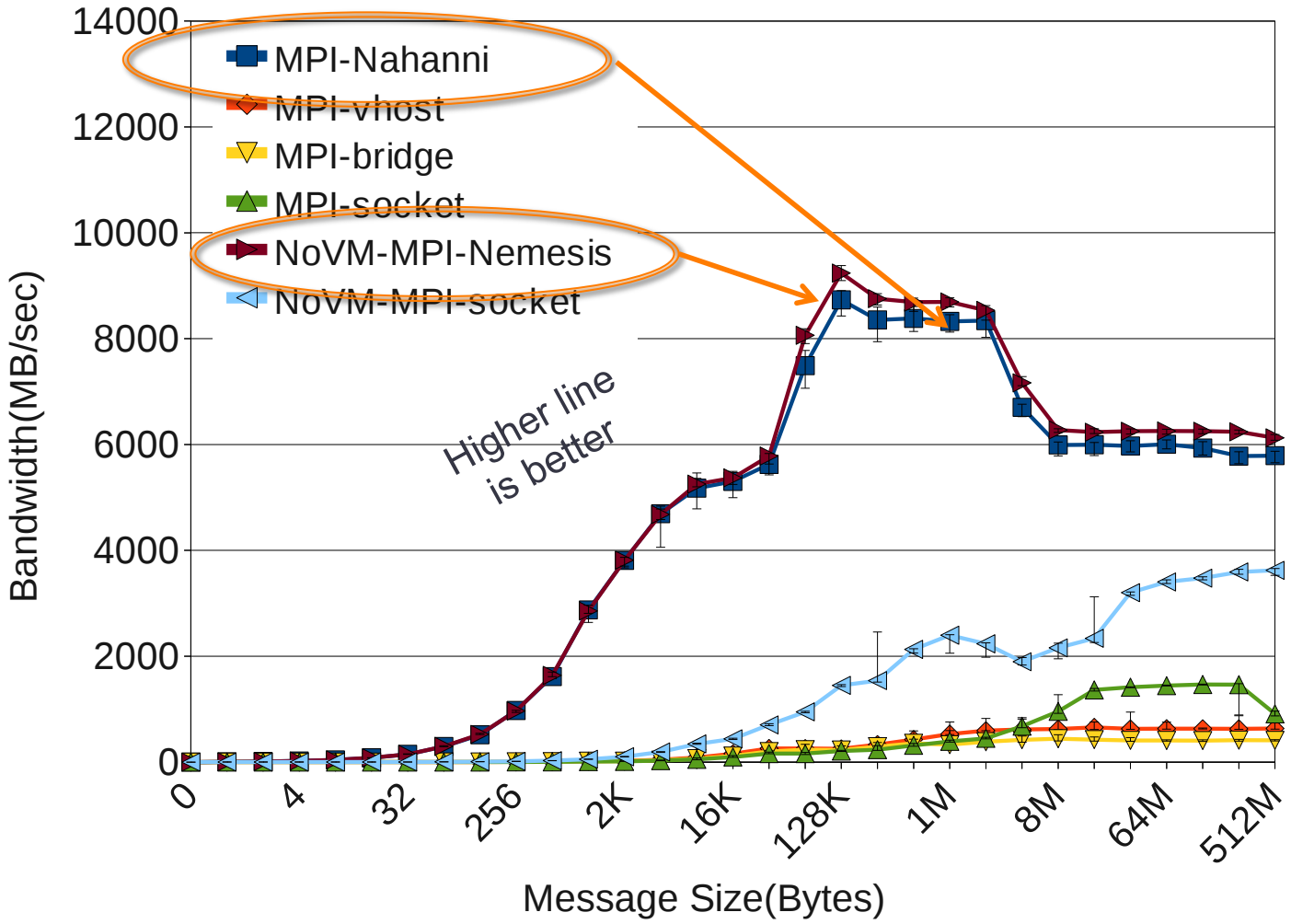
NetPIPE 2-sided Bandwidth (Nahanni vs. vhost)



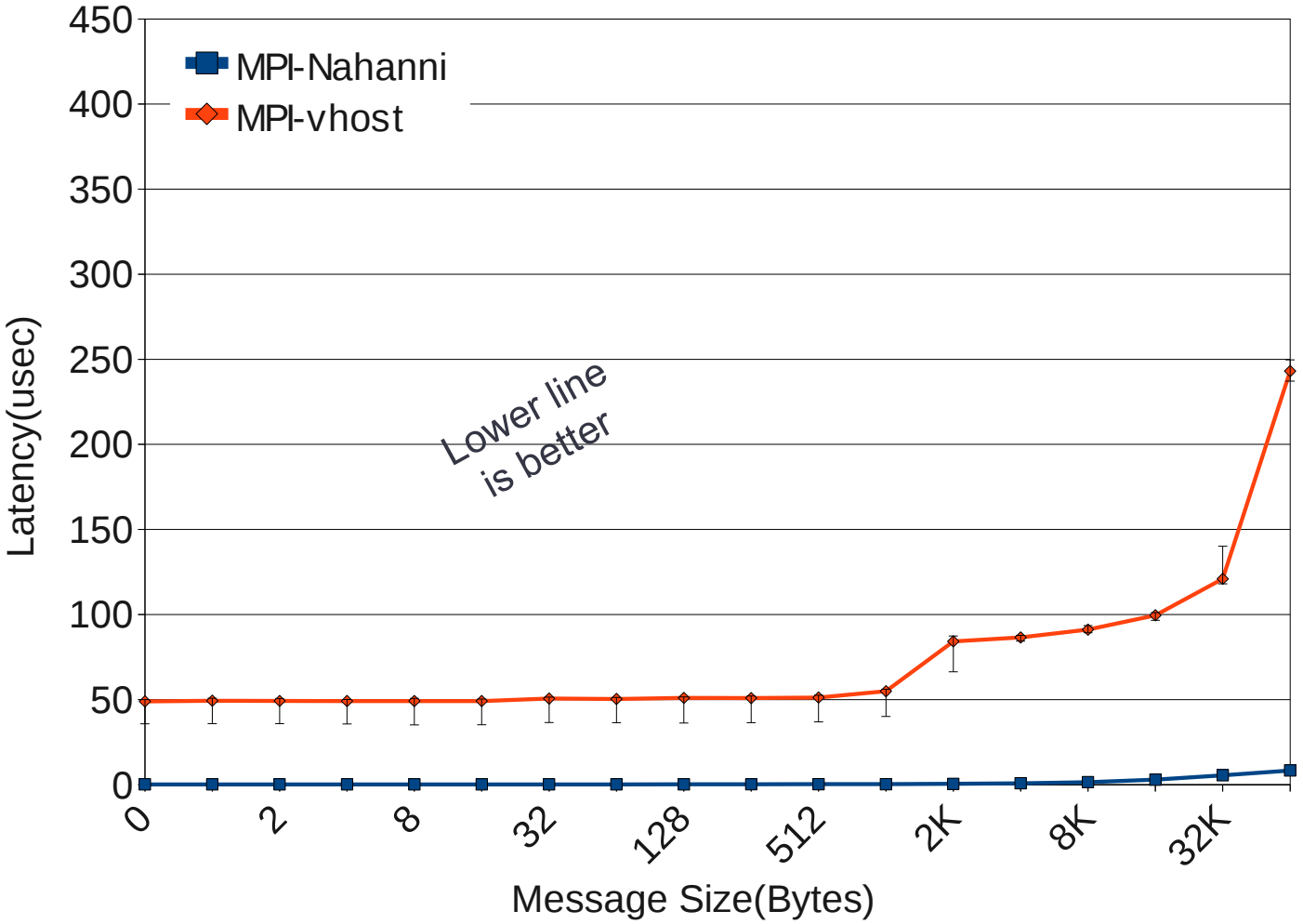
NetPIPE 2-sided Bandwidth (Nahanni vs. vhost)



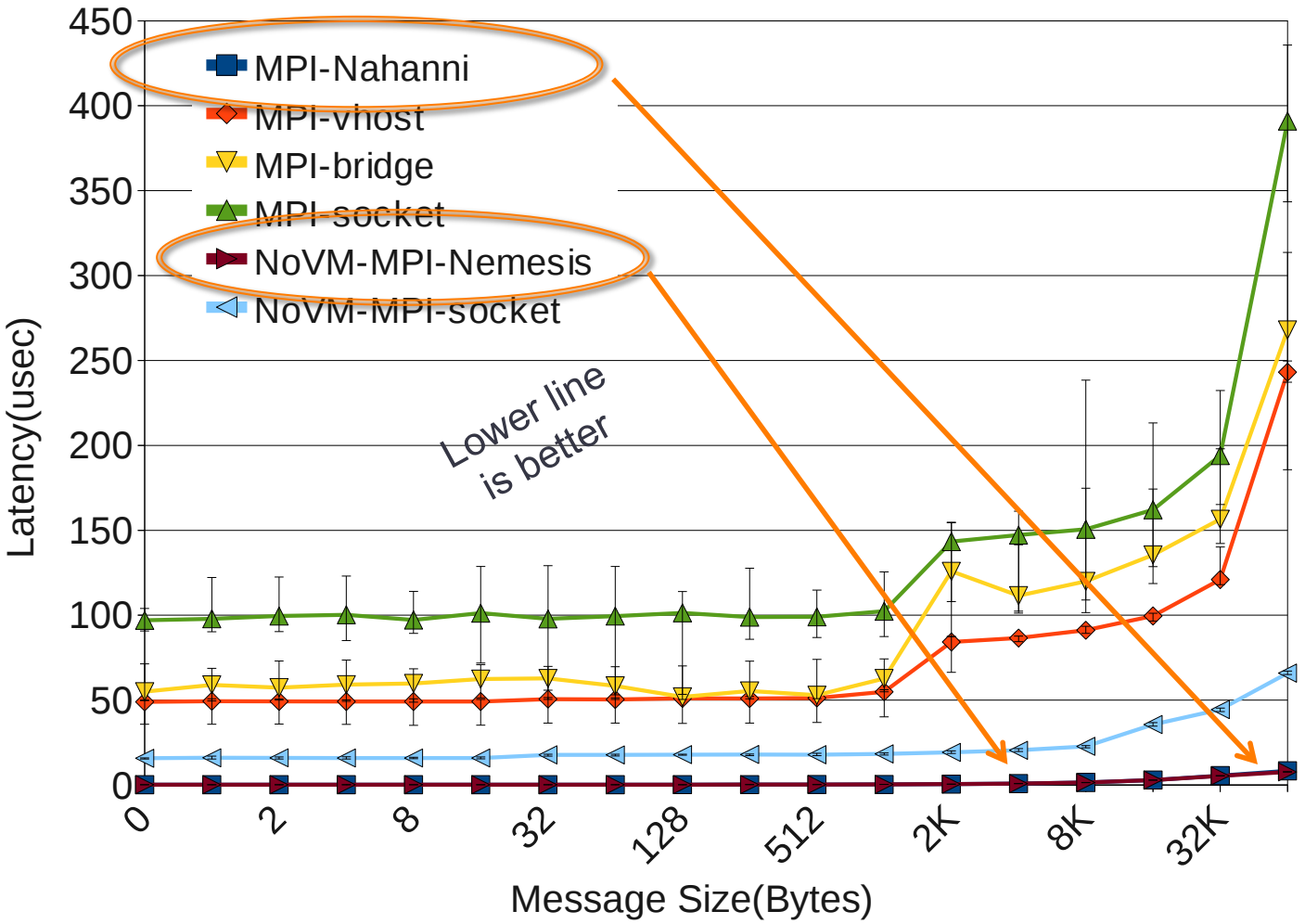
NetPIPE 2-sided Bandwidth (full results)



NetPIPE 2-sided Latency (Nahanni vs. vhost)



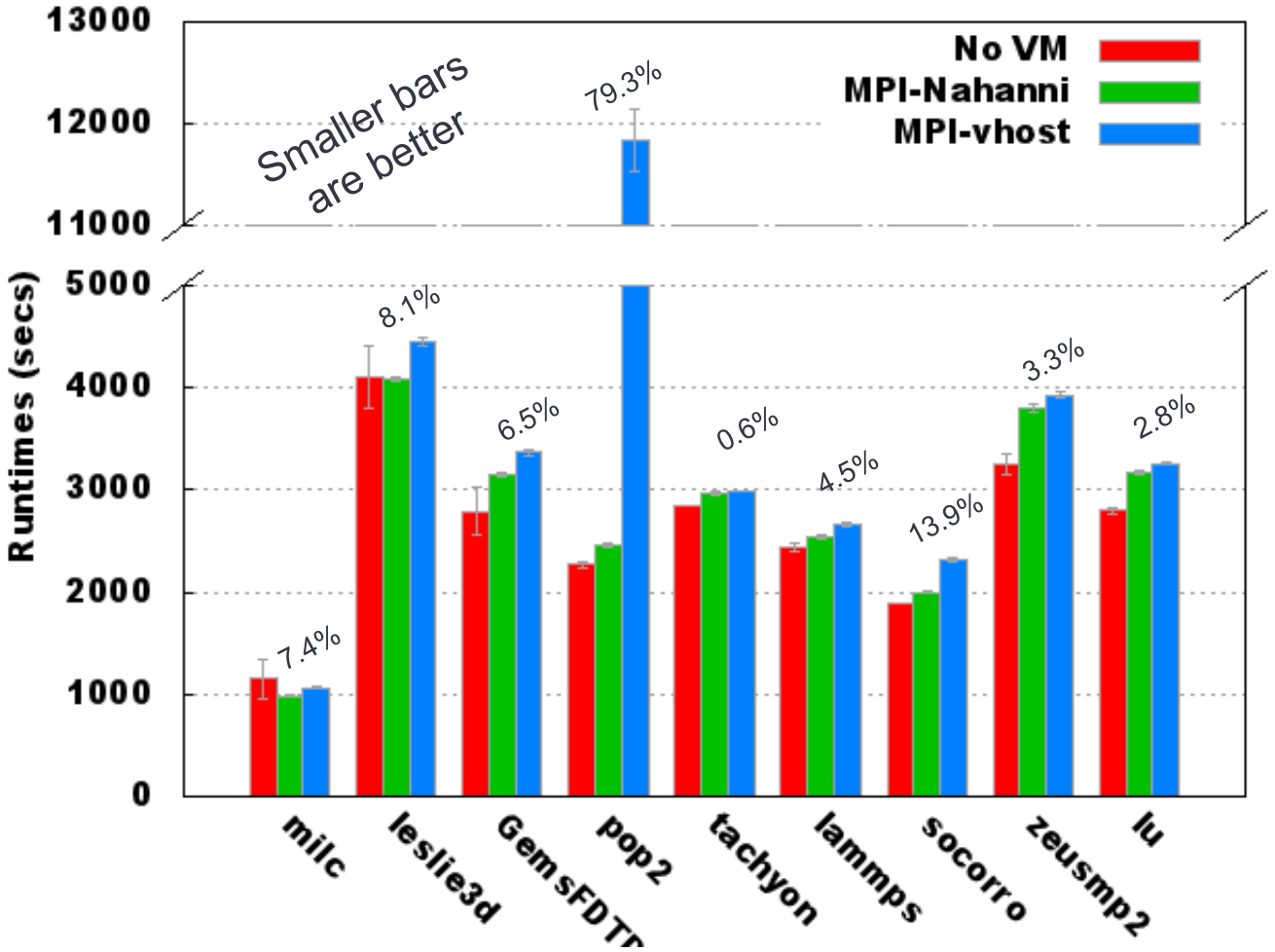
NetPIPE 2-sided Latency (full results)



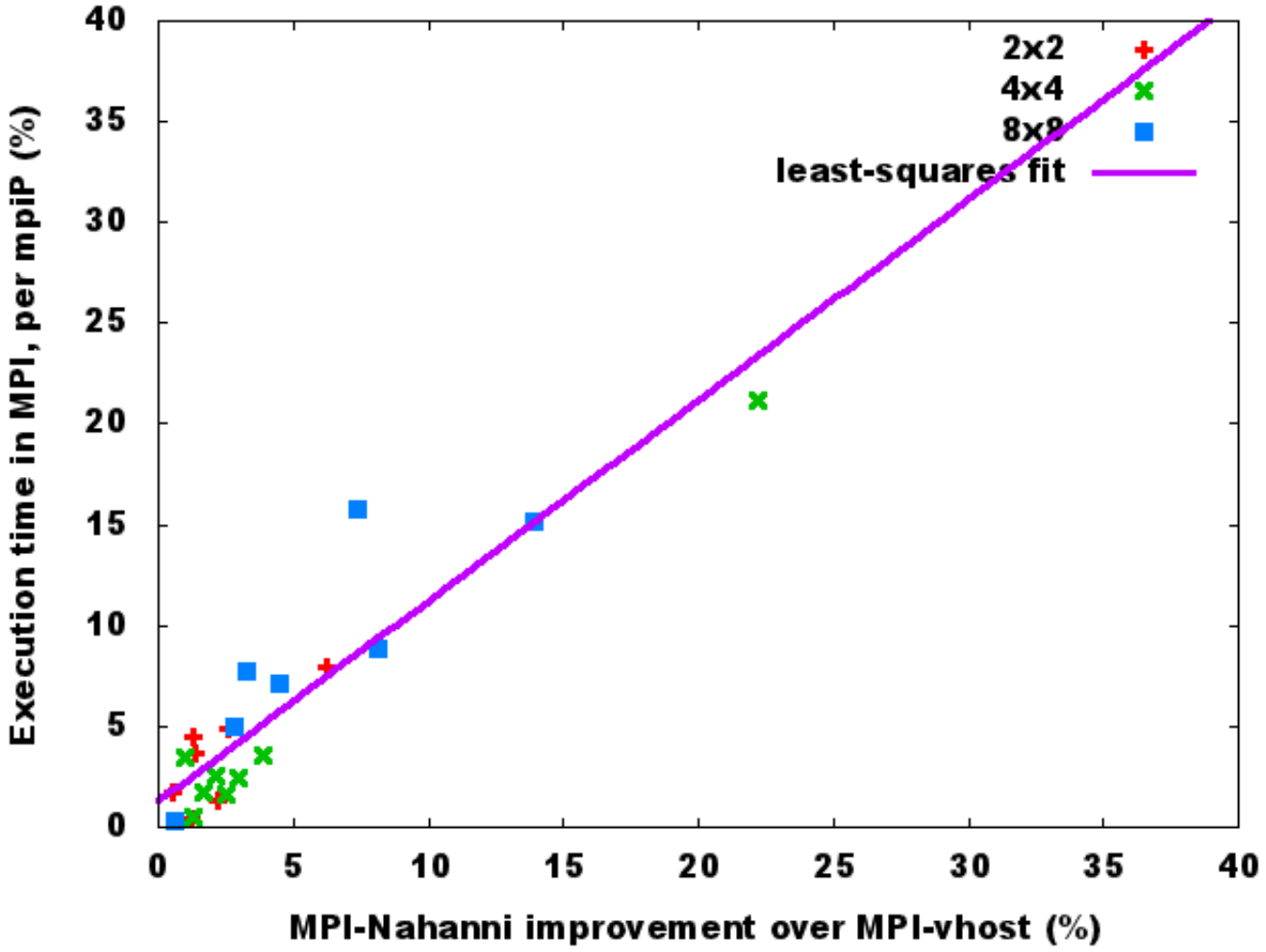
SPEC MPI2007

- SPEC MPI2007 is an industry benchmark for MPI
  - We ran 9 of 13 applications from the medium input set
- Part of Cam Macdonell's Ph.D. thesis
- **Conclusions:**
  - **Performance benefit of MPI-Nahanni grows as the number of processes grows**
  - **Improvement is proportional to time spent in MPI functions**

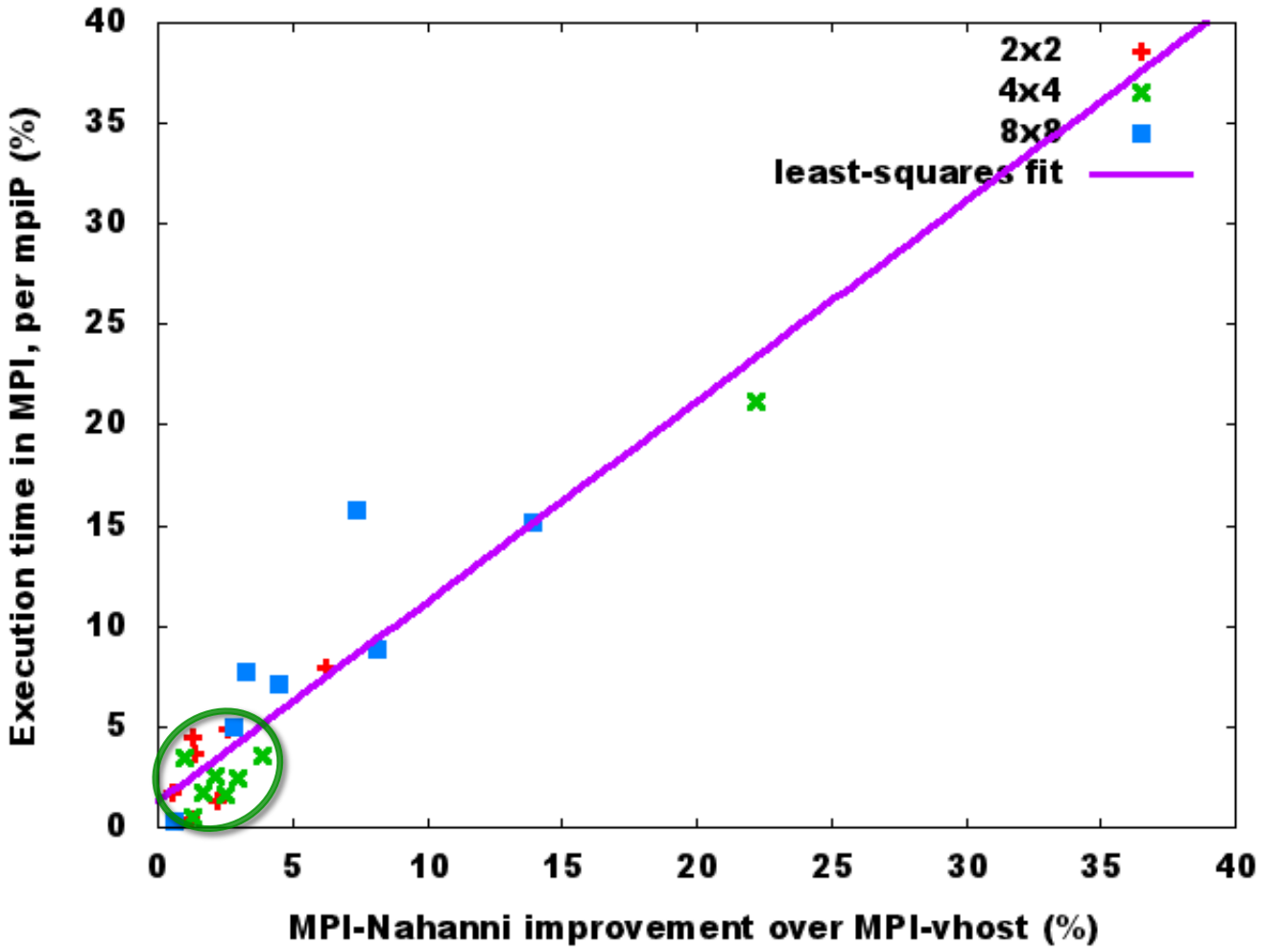
SPEC MPI2007 (8x8)



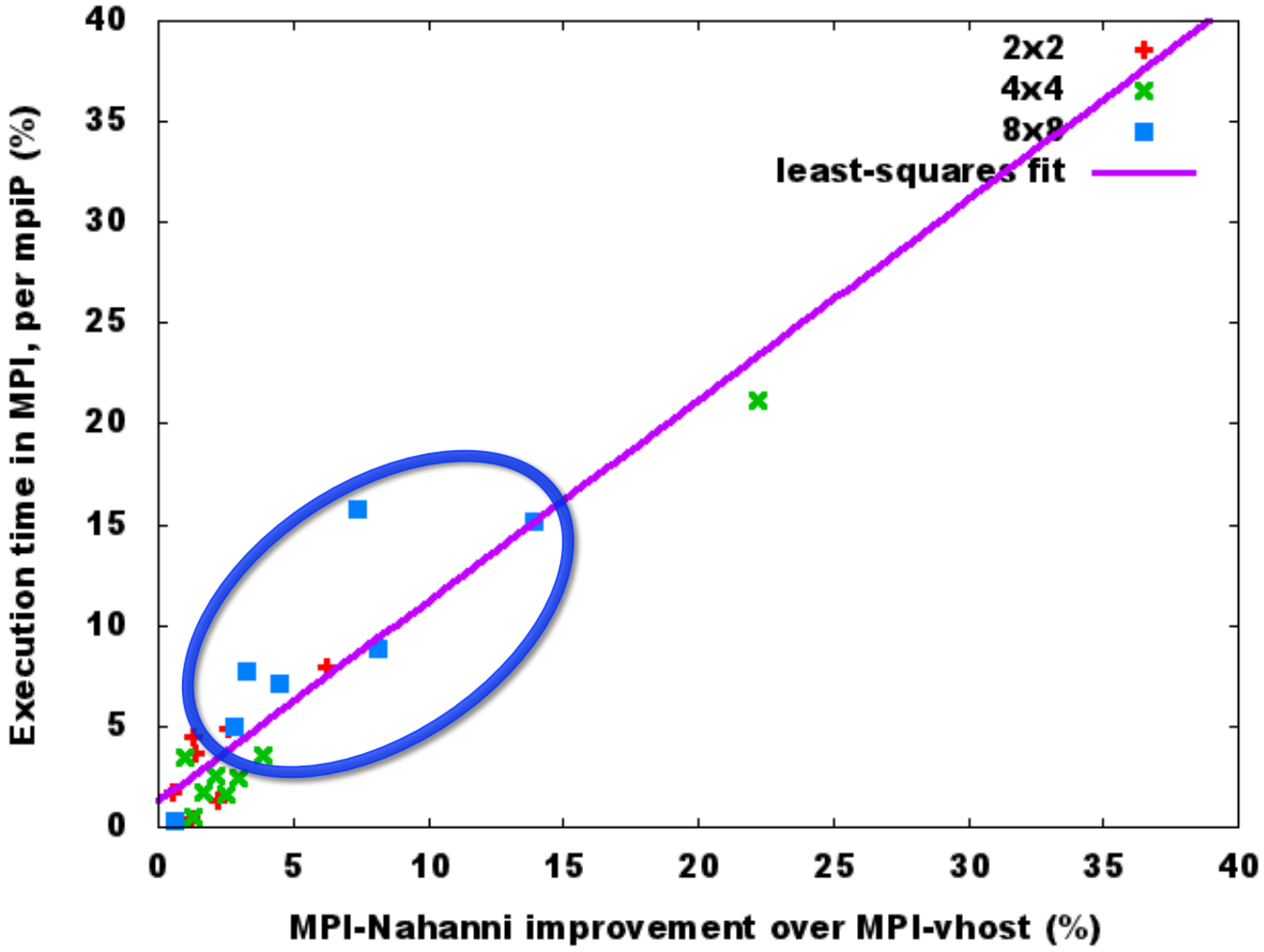
% MPI time (y) vs. % Improvement (x)



% MPI time (y) vs. % Improvement (x)

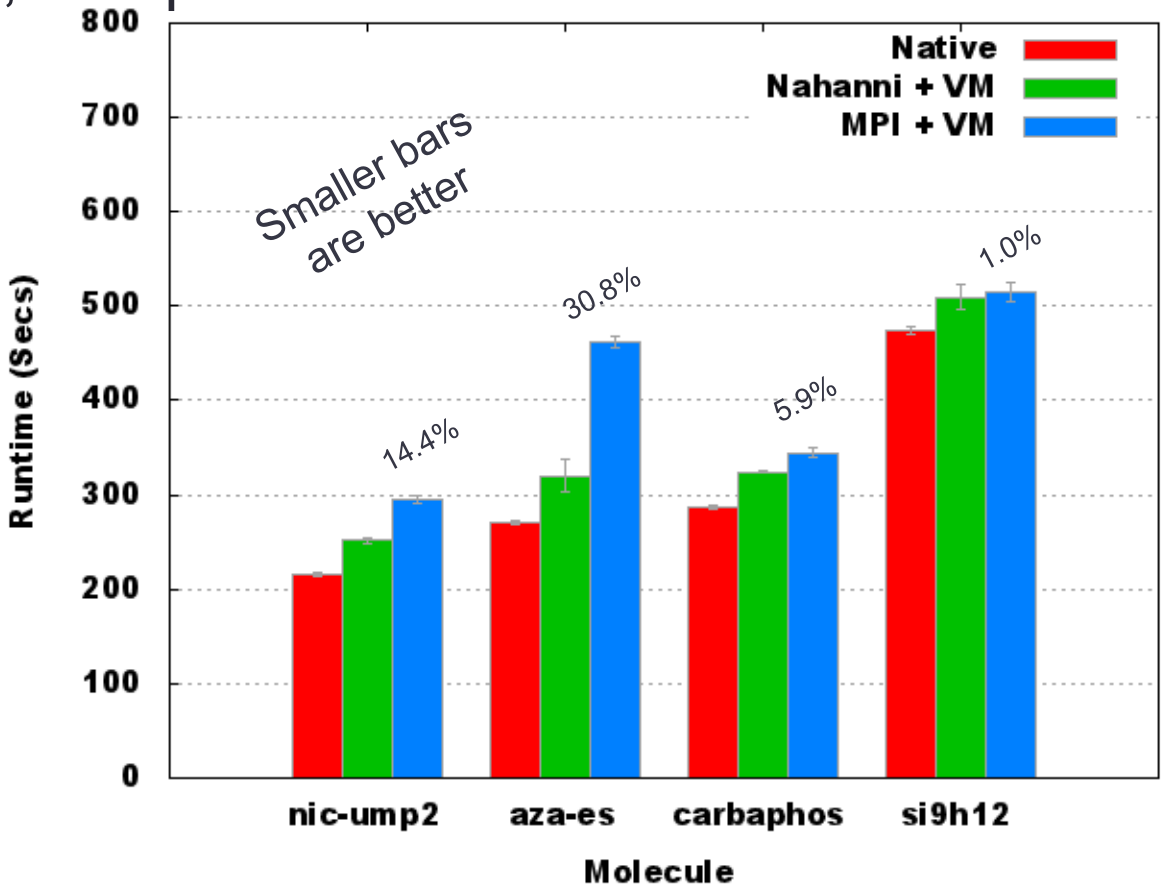


% MPI time (y) vs. % Improvement (x)



GAMESS: Non-MPI Communications

- GAMESS Quantum Chemistry
- Not MPI; DDI ported to Nahanni



Concluding Remarks

- Nahanni / ivshmem is an alternative to other mechanisms (e.g., virtual network, virtio+vhost) for inter-VM, intra-host IPC
  - OS bypass: does not modify or use data movement paths in host or guest VM
  - Supports pointers, non-stream data too
- Web: low latency for client-server, structured data
  - 29% lower on a YCSB + Nahanni memcached workload
- Computational science: low latency and high bandwidth for message-passing applications
  - No changes to MPI code. Up to 30% faster on full applications.

paullu@cs.ualberta.ca