Designing Graphics Clusters

Mike Houston, Stanford University

Why clusters?

- Commodity parts
  - Complete graphics pipeline on a single chip
  - Extremely fast product cycle
- Flexibility
  - Configurable building blocks
- Cost
  - Driven by consumer demand
  - Economies of scale
- Availability
  - You can build (small) systems yourself
  - A trip to a local computer shop can bring a node back up
- Upgradeability
  - Graphics
  - Network
  - Processor
- Scalability
  - CPUs
  - Graphics
  - Memory
  - Disk

Why not clusters?

- App rewrites?
- Debugging
- Shared memory requirements
- Massive I/O requirements
- Software solutions
- Support
- Maintenance
- Who’s going to build it?

Design constraints

- Power
- Cooling
- Density
- Performance
- Cost

These all conflict!!!

Power/Cooling/Density

- Power
  - Graphics + processor = huge power draw
    - Intel Nocona = 103W (load)
    - Nvidia 6800 Ultra = 110W
- Cooling
  - Graphics + processor = lots of heat
  - Fans
    - Noise
    - Reliability
  - Liquid
    - Immense courage
- Density
  - How tall?
  - How deep?
  - Cooling?
  - Power?

Performance/Cost

- Primary cluster use
  - Graphics
    - Spend on graphics
  - Graphics + compute
    - Balance choices, but graphics easier to upgrade
  - Compute
    - Spend on processors and memory
- Bigger/Better/Faster => expensive
- Buy at the “knee”

Component choices

- Processor
  - Intel
  - AMD
  - PowerPC
- Interconnect
  - GigE
  - Quadrics
  - Infiniband
  - Myrinet
- Chipset
  - Consumer
  - Workstation
  - Server
- Graphics
  - Vendor
    - 3DLabs
    - ATI
    - Nvidia
  - Market segment
    - Consumer
    - Workstation
- Chassis
  - Desktop case
  - Rackmount
    - Height (1/2/3/4/5U)
    - Depth (Half/Full)

What NOT to do!!!

- Assume a Graphics Cluster is like a Compute Cluster
  - Different cooling, power, performance constraints
  - Different usage scenarios
  - Different bus loads
- Use “riser” boards
  - Signal quality
  - Cooling
  - I’ve seen more issues with this than anything else!!!

What NOT to do!!! continued

- Purchase untested/new chipsets
  - Performance oddities
  - Stability problems
  - Many people bitten by Intel i840/i860 chipsets’ AGP and PCI performance problems
- Buy cheap components
  - Failures
  - Stability
  - Saving $5 off a fan might cost you thousands in hardware failures, down time from instability, and man hours trying to track down the problem

What TO do

- Get help
  - Talk to others who have built these in academia/industry
  - Work with a company that has built these
  - Buy parts/whole thing from a company that has built these
- Testing, Testing, Testing
  - Pound on a few options before you choose, or copy known working solution
  - Processor performance/heat/cooling/stability
  - Graphics performance/heat/cooling/stability
  - Bus performance/stability
  - Network performance/stability
  - Temperature monitoring

What TO do continued

- Maintenance
  - Clean filters
  - Clean/check fans
  - Run memory/processor tests
    - Memtest86 (http://www.memtest.org)
    - CPU Burn-in (http://users.bigpond.net.au/cpuburn)
  - Check disks
    - fsck often
- Monitor your cluster
  - Temperature fluctuations
  - Node stability
  - Checkout Ganglia (http://ganglia.sourceforge.net)

Sources of Bottlenecks

- Sort-First
  - Pack/unpack speed (processor)
  - Primitive distribution (network and bus)
  - Rendering (processor and graphics chip)
- Sort-Last
  - Rendering (graphics chip)
  - Composite (network, bus, and read/draw pixels)

Rendering and the network

- Sort-last
  - Usage patterns
    - All-to-all
    - Pair-swapping, all nodes
  - Switch requirements
    - Provide backplane to support all nodes sending
    - Non-blocking
- Sort-first
  - Usage patterns
    - 1 to N
    - M to N
  - Switch requirements
    - Multicast/broadcast support

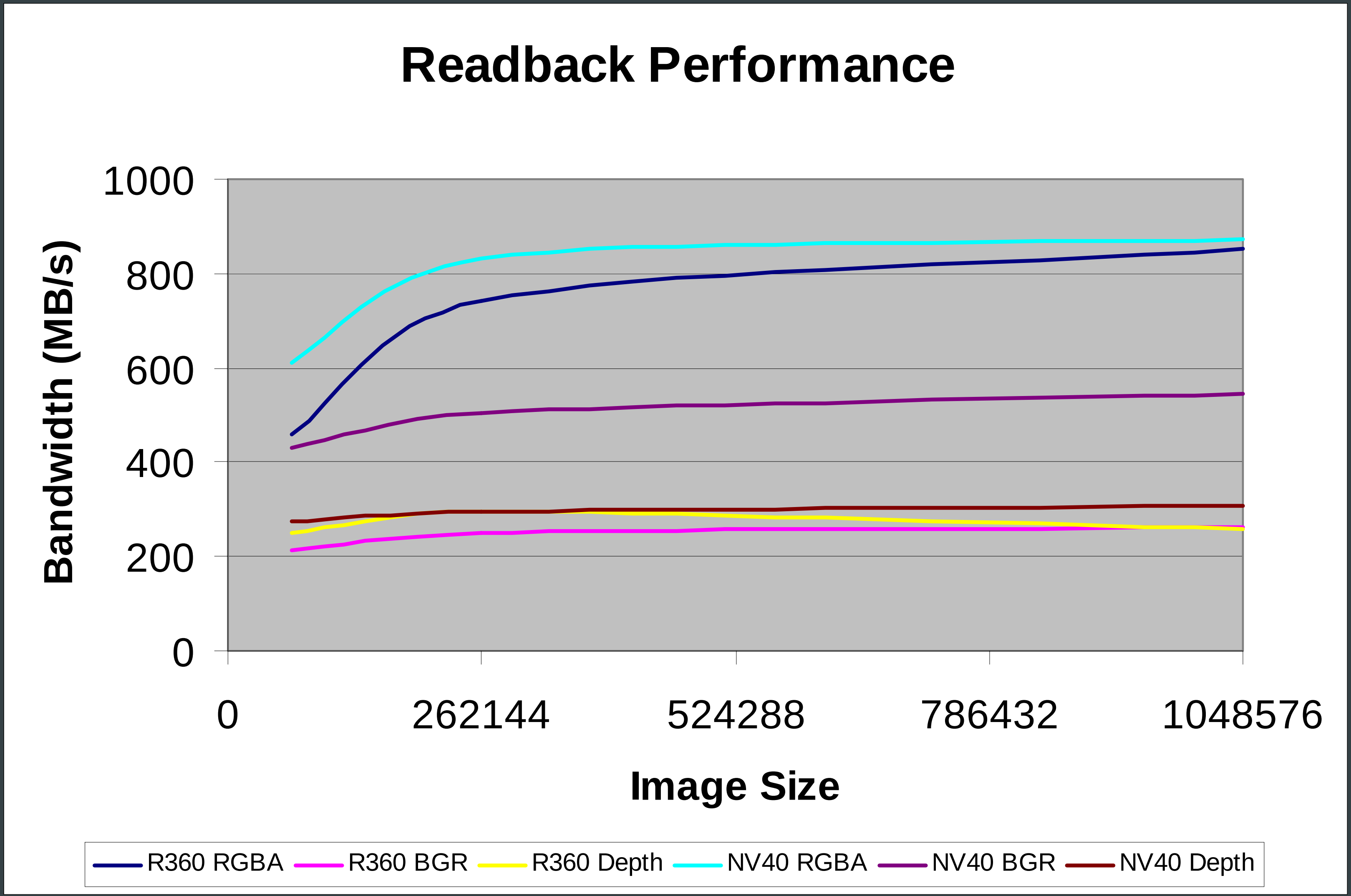
Network interconnects

- GigE
  - Bandwidth: ~90MB/s (large MTU)
  - Latency: 50-100 usec
  - Cost per port: <$100
  - Get chips with a TOE
- Bonded GigE
  - Works great for up to ~4 ports
  - Gets expensive fast, especially for fully connected networks

High speed interconnects

- Quadrics
  - Bandwidth: 876MB/s (elan4)
  - Latency: 3usec
  - Cost: $1,866 per port
- Myrinet
  - Bandwidth: 500MB/s
  - Latency: 3.5usec
  - Cost: $1,600 per port
- Infiniband 4X
  - Bandwidth: 1450MB/s (PCIe) / 850MB/s (PCI-X)
  - Latency: 4usec / 8usec
  - Cost: <$1,000 per port

Graphics readback (AGP)



PCIe

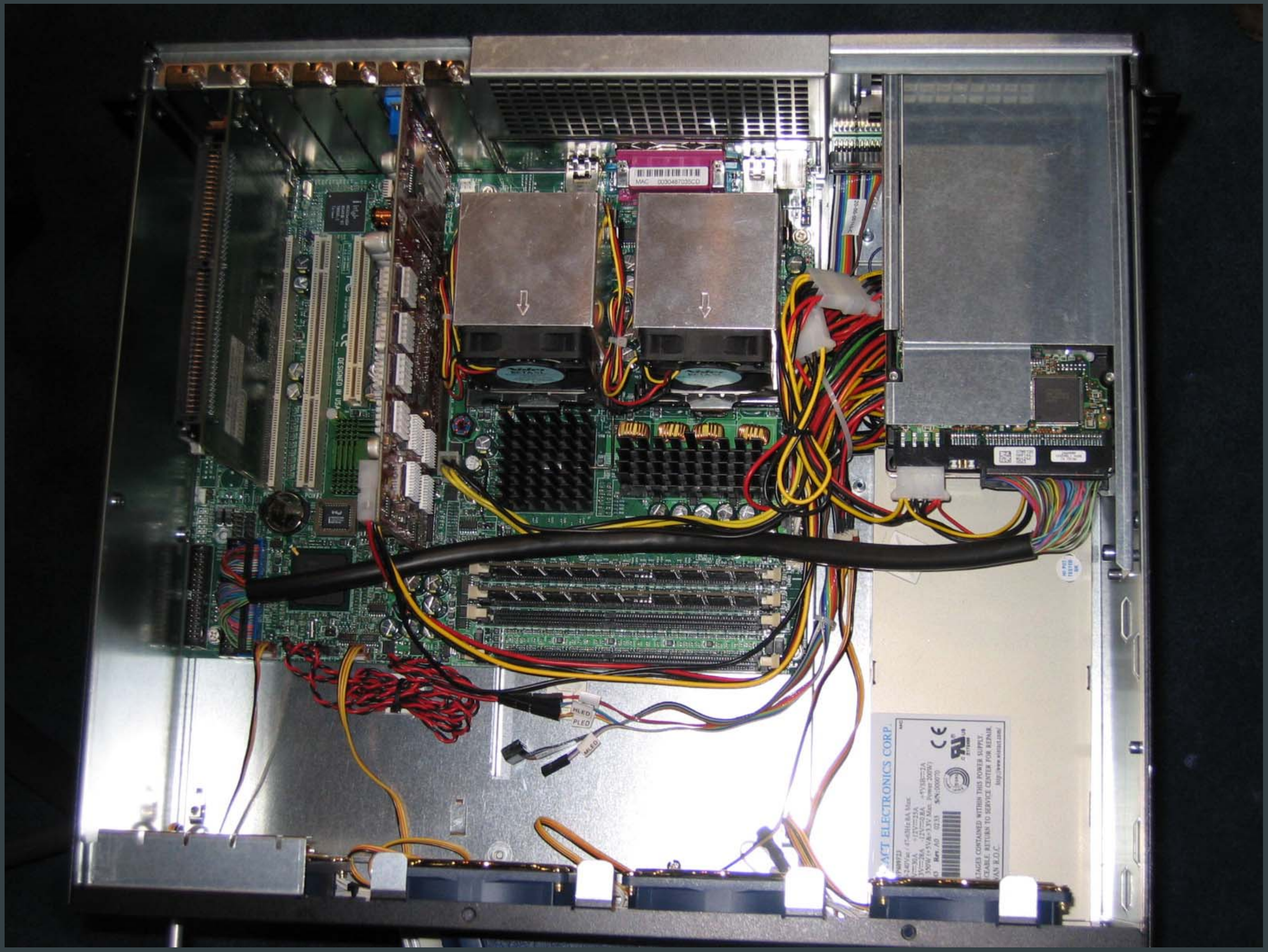
- The promise
  - Graphics readback performance
    - Easier implementation
    - ~2GB/s
  - Network performance
  - Unified standard for graphics, network, I/O
- Problems
  - Limited number of slots
    - 1 x16 + 1 x8 or several x4
  - Stability/Performance
    - Early implementations have “problems”

Case Study - Stanford's SPIRE

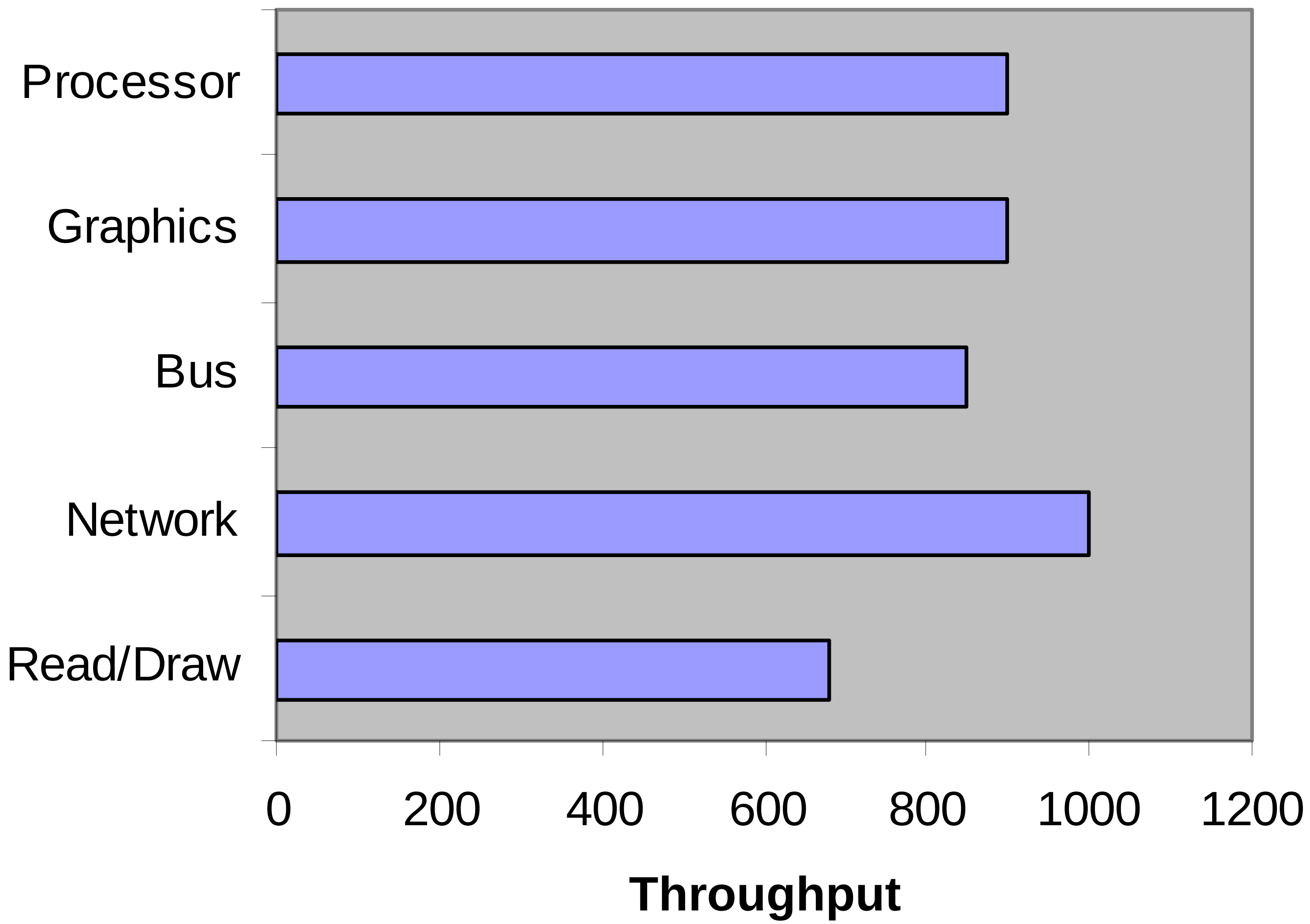
- 16 node cluster
- 32 2.4GHz P4 Xeons
- E7505 chipset (Supermicro)
- 16GB DDR
- 1.2TB IDE storage
- Mellanox Cougar HCA
- Mellanox 16 port InfiniScale switch
- Dlink 24-port GigE switch
- ATI 9800 Pro 256MB (AGP)
- Linux – Fedora Core 2

http://spire.stanford.edu

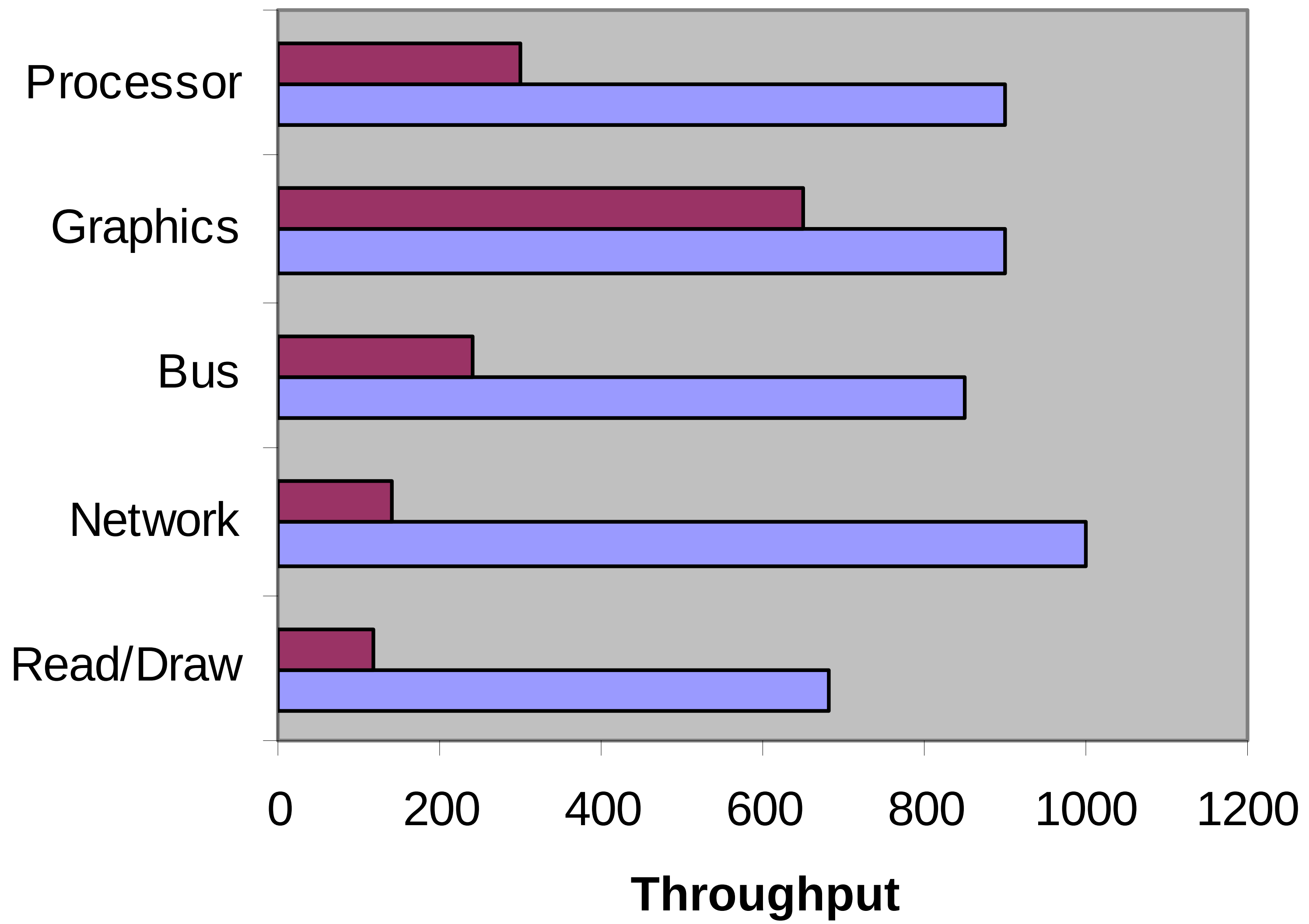
Inside a node



Bottleneck Evaluation – SPIRE



SPIRE vs. Chromium



Compositing Performance (GigE)

- RGBA (1024x1024 image across 16 nodes)
  - Software:
    - 9.5fps
    - 152 MPixel/sec
  - Hardware:
    - 17fps
    - 269 MPixel/sec
- Depth (RGB + Z) (1024x1024 image across 16 nodes)
  - Software:
    - 3.8fps
    - 60 Mpixels/sec
  - Hardware:
    - 7.2fps
    - 116 Mpixels/sec

Compositing Performance (Infiniband)

- RGBA (1024x1024 image across 16 nodes)
  - Software:
    - 14fps
    - 224 MPixel/sec
  - Hardware:
    - 45fps
    - 720 MPixel/sec
- Depth (RGB + Z) (1024x1024 image across 16 nodes)
  - Software:
    - 6fps
    - 96 Mpixels/sec
  - Hardware:
    - 11fps
    - 176 Mpixels/sec

SPIRE

- Performance
  - 8.2 GVox/s (1024^3 @ 8Hz)
  - Quake3 @ 5120x4096: 90fps
  - 790MB/s node to node
  - 10.2GB/s cross sectional bandwidth
- Hardware failures
  - 3 Western Digital drives
  - 4 fans (1 rack, 2 chassis, 1 GPU)
  - 1 CPU

Hardware Suggestions

Low end systems

Characteristics: Single processor, GigE

Cost: <$2,000

- Intel Based
  - Intel P4
  - Intel 925X
    - PCIe
    - GigE onboard
  - 1GB DDR
  - Nvidia 6800GT
- AMD based
  - Athlon 64
  - Nforce3 250Gb
    - AGP
    - GigE onboard
  - 1GB DDR
  - Nvidia 6800GT

Mid range systems

Characteristics: Dual processor, GigE

Cost: <$5,000

- Intel Based
  - Intel Xeon
  - Intel E7525
    - PCIe
    - Dual GigE onboard
  - 4GB DDR
  - Nvidia 6800 Ultra

- AMD based
  - Dual Opteron
  - AMD 85XX
    - AGP
    - Dual GigE onboard
  - 4GB DDR
  - Nvidia 6800 Ultra

High end systems

Characteristics: Dual processor, high speed interconnect

Cost: ~$10k => Arm/leg/first born

- Intel Based
  - Intel Xeon
  - Intel E7525
    - PCIe
    - Dual GigE onboard
  - 16GB DDR
  - Nvidia 6800 Ultra / Quadro 4400/G / SLI
  - Infiniband 4X / Quadrics
    - Single or multirail
- AMD based
  - Dual Opteron
  - AMD 85XX
    - AGP
    - Dual GigE onboard
  - 16GB DDR
  - Nvidia 6800 Ultra / Quadro 4400/G / SLI
  - Infiniband 4X / Quadrics
    - Single or multirail

What would I build next?

- Intel
  - Dual Nocona
  - E7575 (Tumwater)
  - Infiniband 4X (PCIe)
  - Nvidia NV4X
- Good things
  - Known solution
  - Currently available
- Bad things
  - Heat
  - PCIe issues

- AMD
  - Dual/Quad Opteron
  - PCIe chipset?
    - AMD
    - Nvidia Nforce4
  - Nvidia NV4X
- Good things
  - Heat
  - Dual core chips for upgrade
  - Multiple PCIe x16 slots
- Bad things
  - Unknown performance/stability
  - Not available just yet

Example E7525 prototype

Courtesy GraphStream

List of graphics cluster companies

- ABBA
- GraphStream
- HP
- IBM
- Orad

Questions?

Supplemental

Chromium

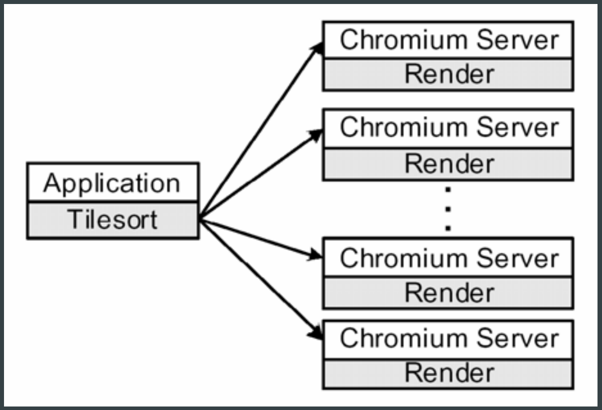
The Chromium Cluster

Chromium Cluster Configuration

- Cluster: 32 graphics nodes + 1 server node
- Computer: Compaq SP750
  - 2 processors (800 MHz PIII Xeon, 133MHz FSB)
  - i840 core logic
    - Simultaneous “fast” graphics and networking
    - Graphics: AGP-4x
  - 256 MB memory
  - 18GB SCSI 160 disk (+ 3*36GB on servers)
- Graphics
  - 16 NVIDIA GeForce3 w/ DVI (64 MB)
  - 16 NVIDIA GeForce4 TI4200 w/ DVI (128 MB)
- Network
  - Myrinet 64-bit, 66 MHz (LANai 7)

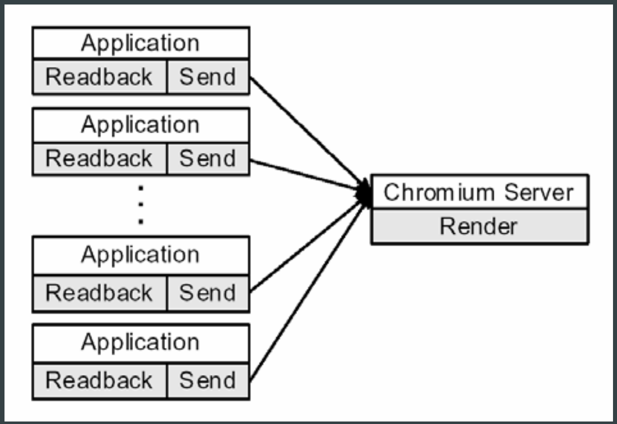
Sort-First Performance

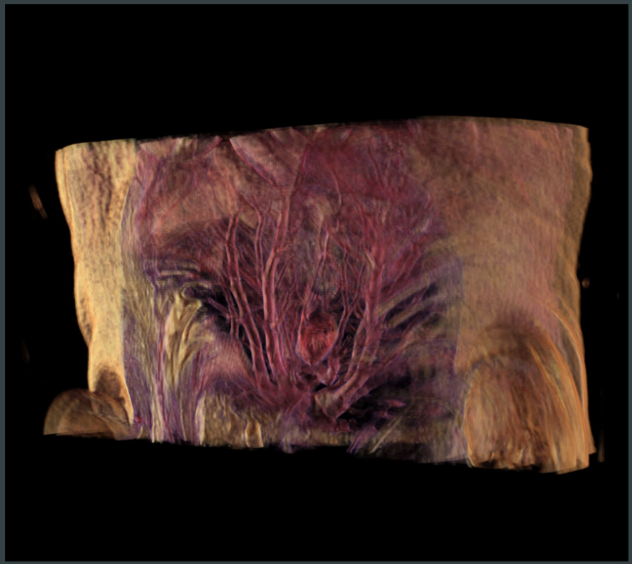
- Configuration
  - Application runs application on client
  - Primitives distributed to servers
- Tiled Display
  - 4x3 @ 1024x768
  - Total resolution: 4096x2304, 9 Megapixel
- Quake 3
  - 50 fps



Sort-Last Performance

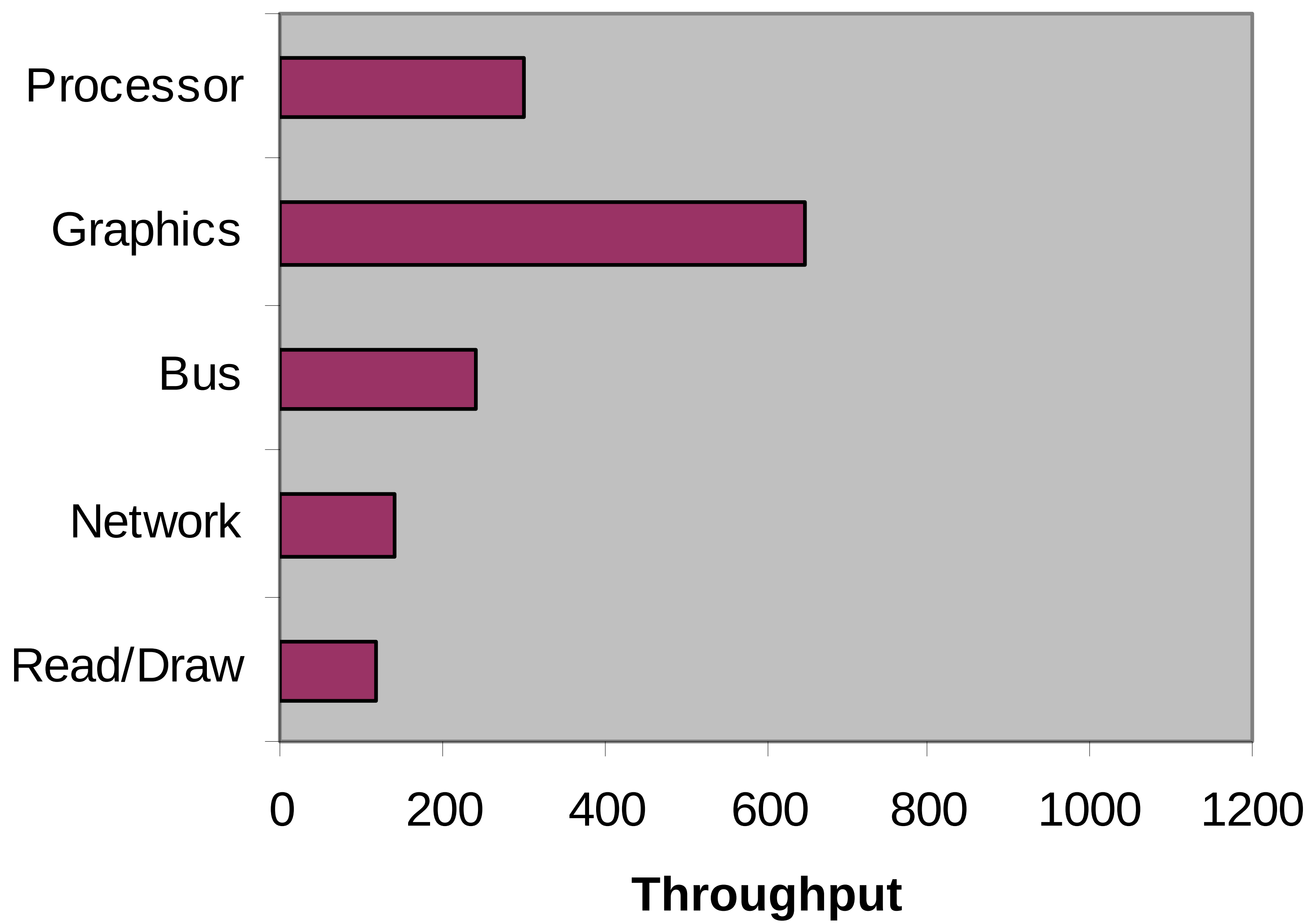
- Configuration
  - Parallel rendering on multiple nodes
  - Composite to final display node
- Volume Rendering on 16 nodes
  - 1.57 GVox/s [Humphreys 02]
  - 1.82 GVox/s (tuned) 9/02
  - 256x256x1024 volume[1] rendered twice





[1]Data Courtesy of G. A Johnson, G.P.Cofer, S.L Gewalt, and L.W. Hedlund from the Duke Center for In Vivo Microscopy (an NIH/NCRR National Resource)

Bottleneck Evaluation – Chromium

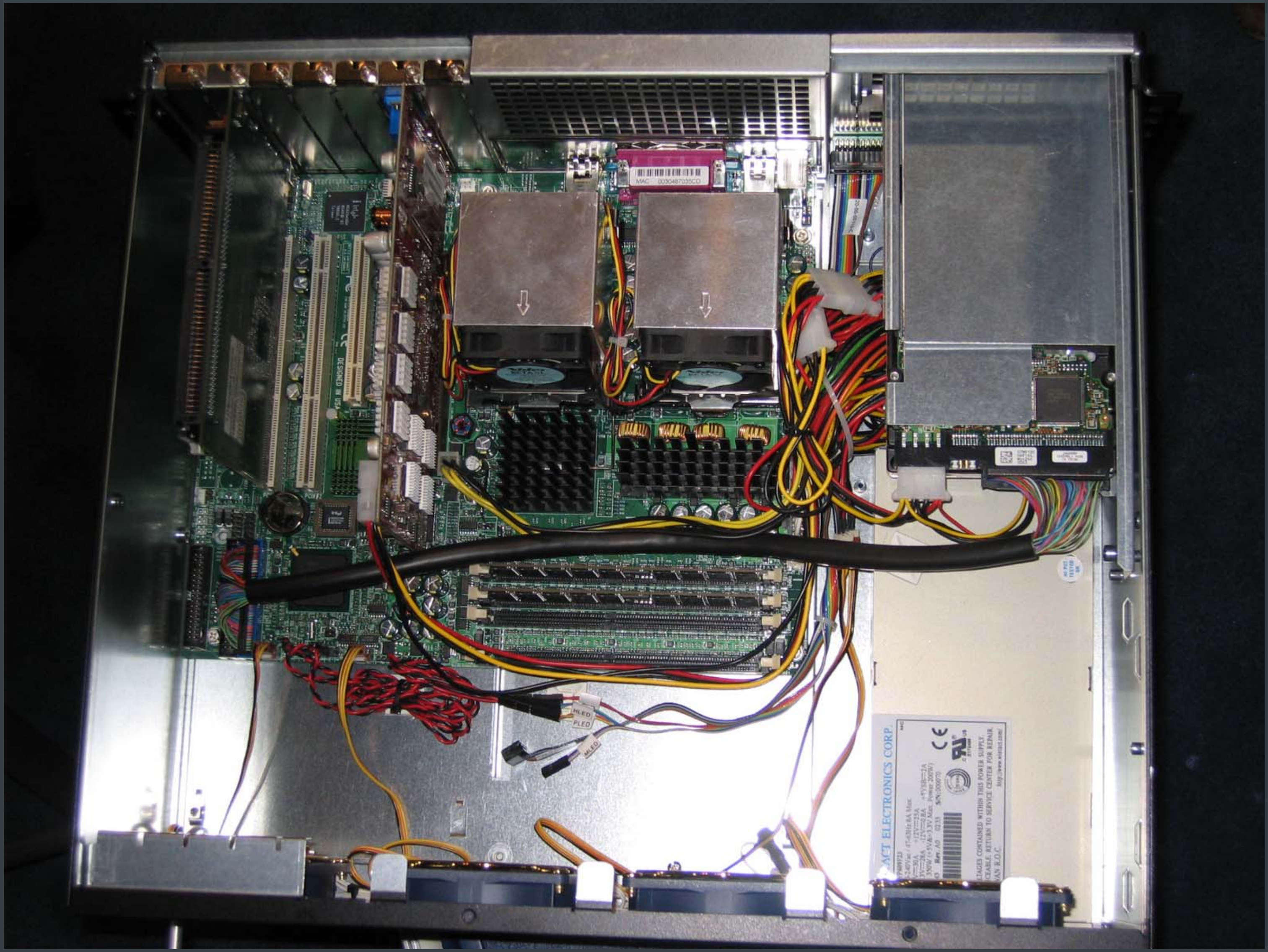


Stanford's SPIRE

Cluster Configuration

- 16 node cluster, 3U half depth
- 2.4GHz P4 Xeon (Dual)
- Intel E7505 chipset1GB DDR (up to 4GB)
- ATI Radeon 9800 Pro 256MB
- Infiniband + GigE
  - Mellanox Cougar HCA
  - Mellanox 6 chip 16-port switch
- 80 GB IDE
- Built by Graphstream
  - We already built one and knew better...
  - Someone else to support hardware failures

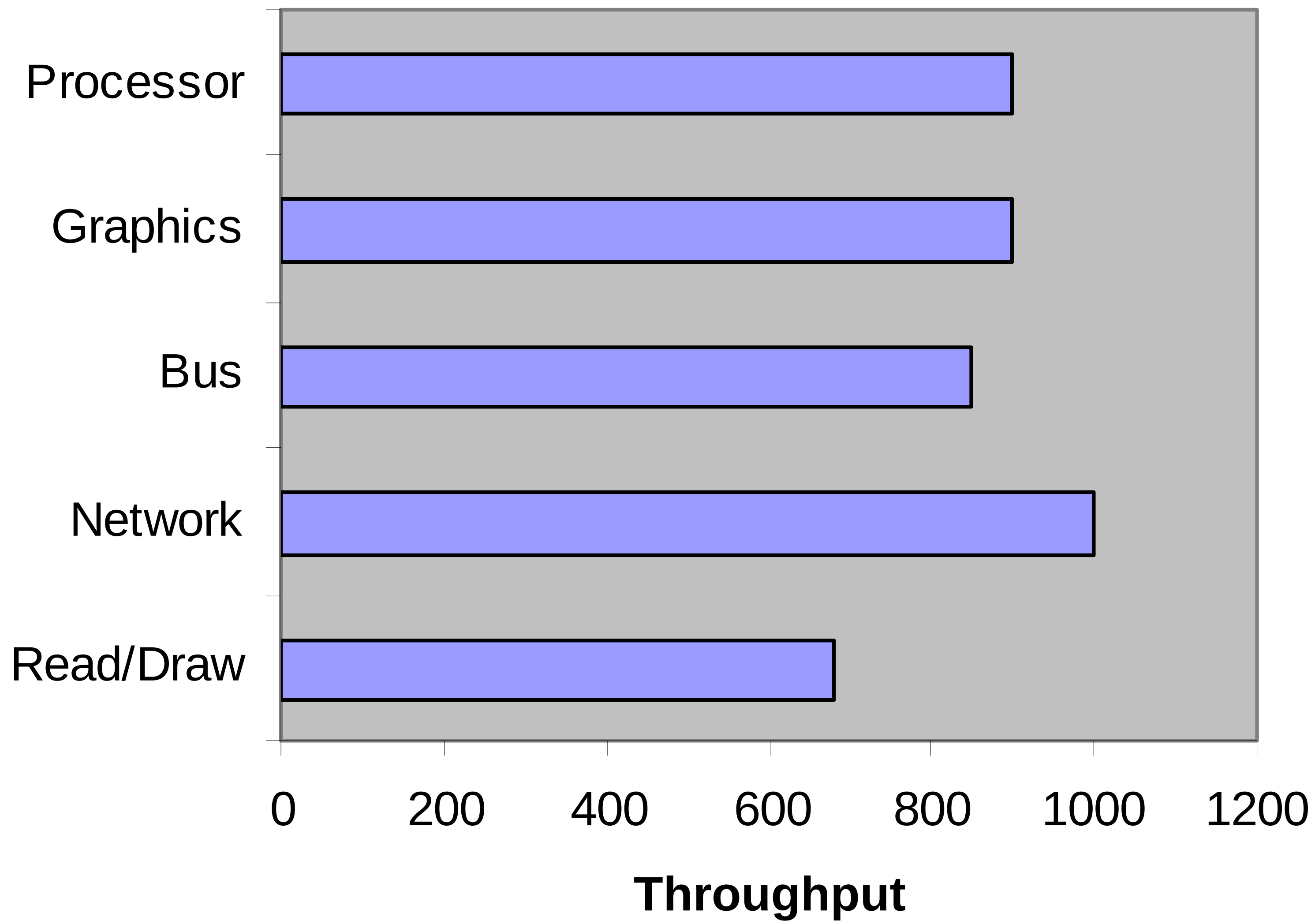
Inside a node



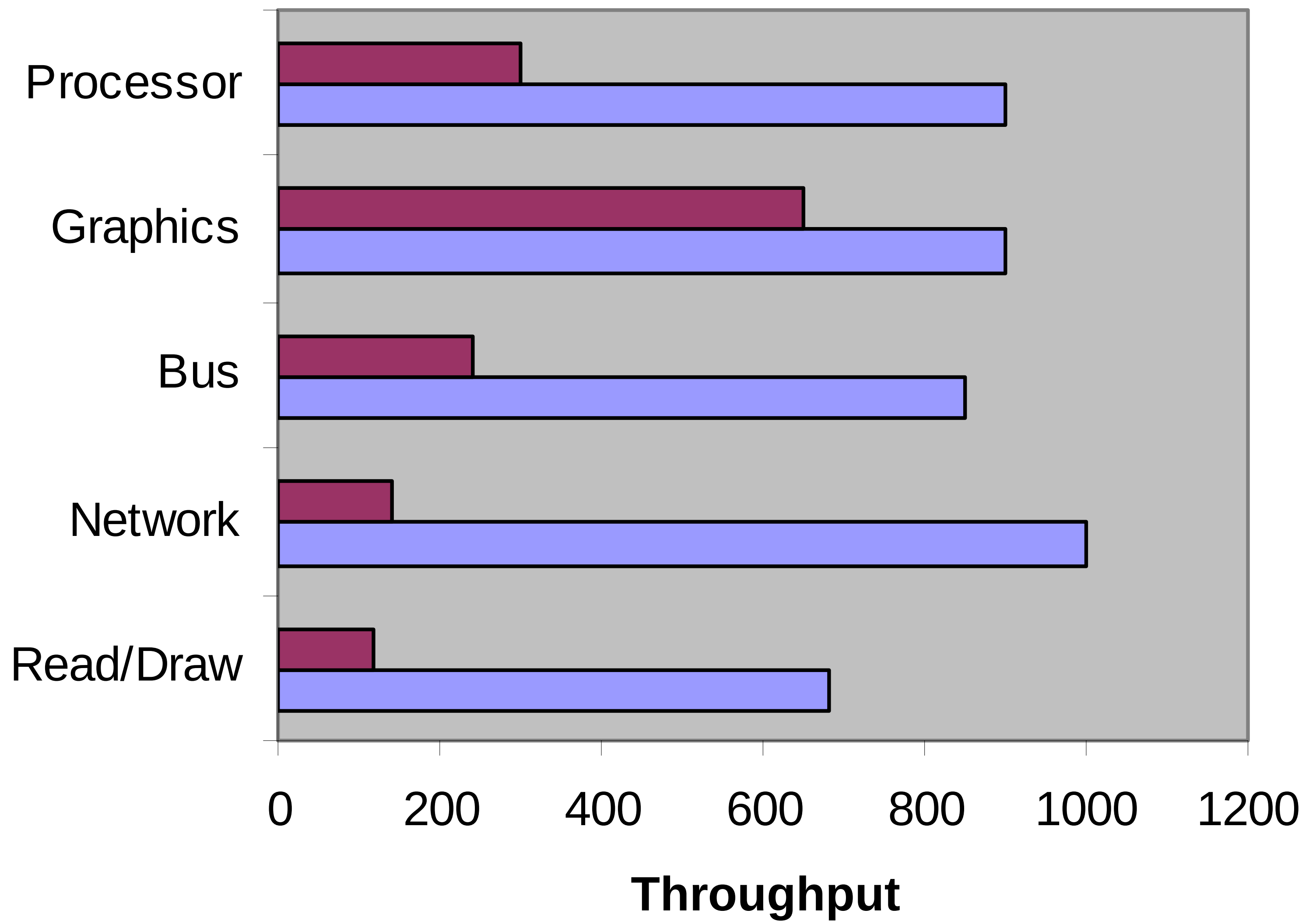
Bleeding edge is painful

- Infiniband
  - 5 months to get IB working and MPI running
    - Driver and firmware issues
  - 3 months to get Chromium VAPI layer running
    - No documentation or support
  - 1 month to get SDP layer working
    - Driver issues
- Linux 2.6 kernel
  - Much better I/O performance
  - Graphics hardware driver issues
    - 4K stack change
    - Register argument passing (REGPARM)
  - Preemptable kernels break many drivers

Bottleneck Evaluation – SPIRE



SPIRE vs. Chromium

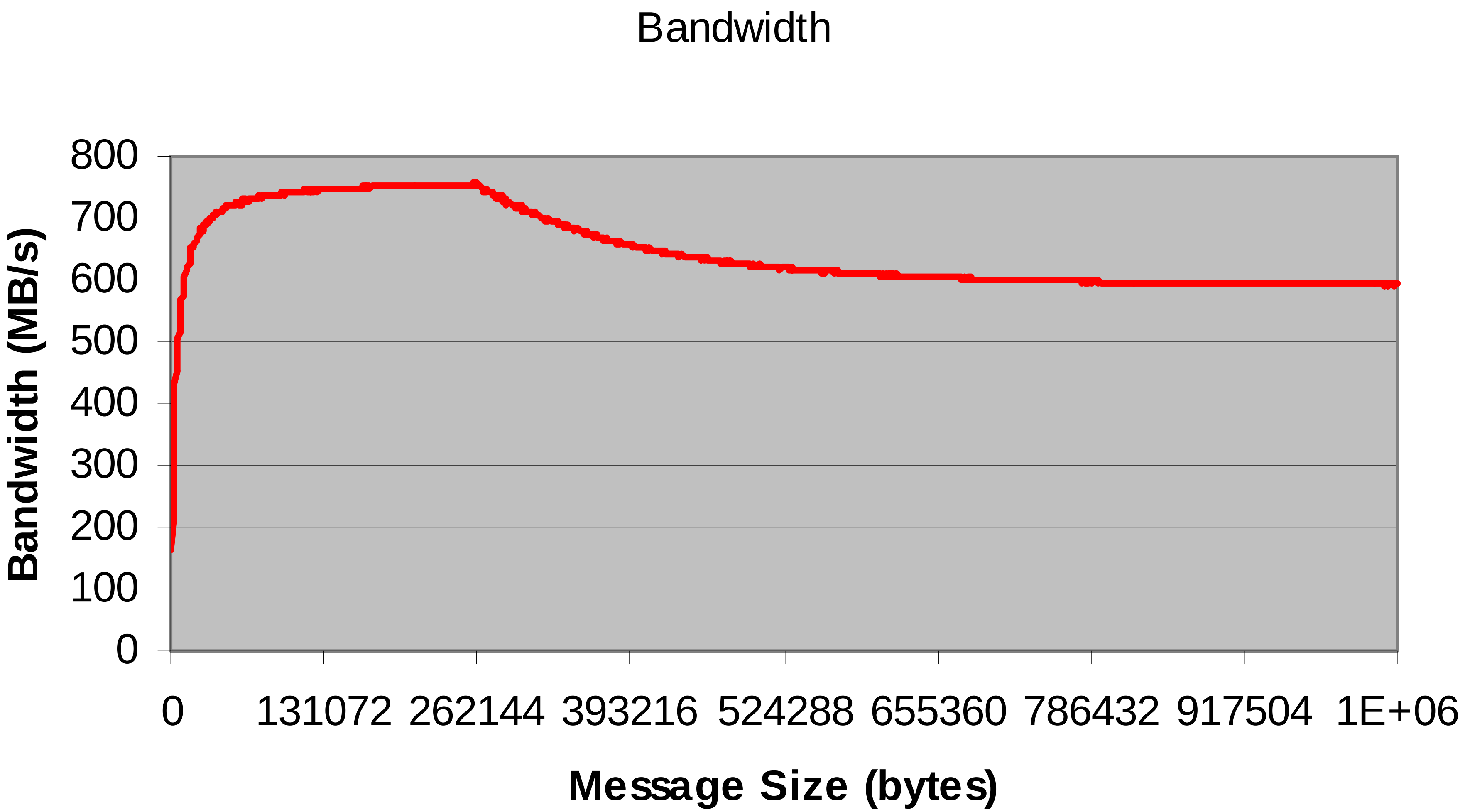


A deeper look into Infiniband

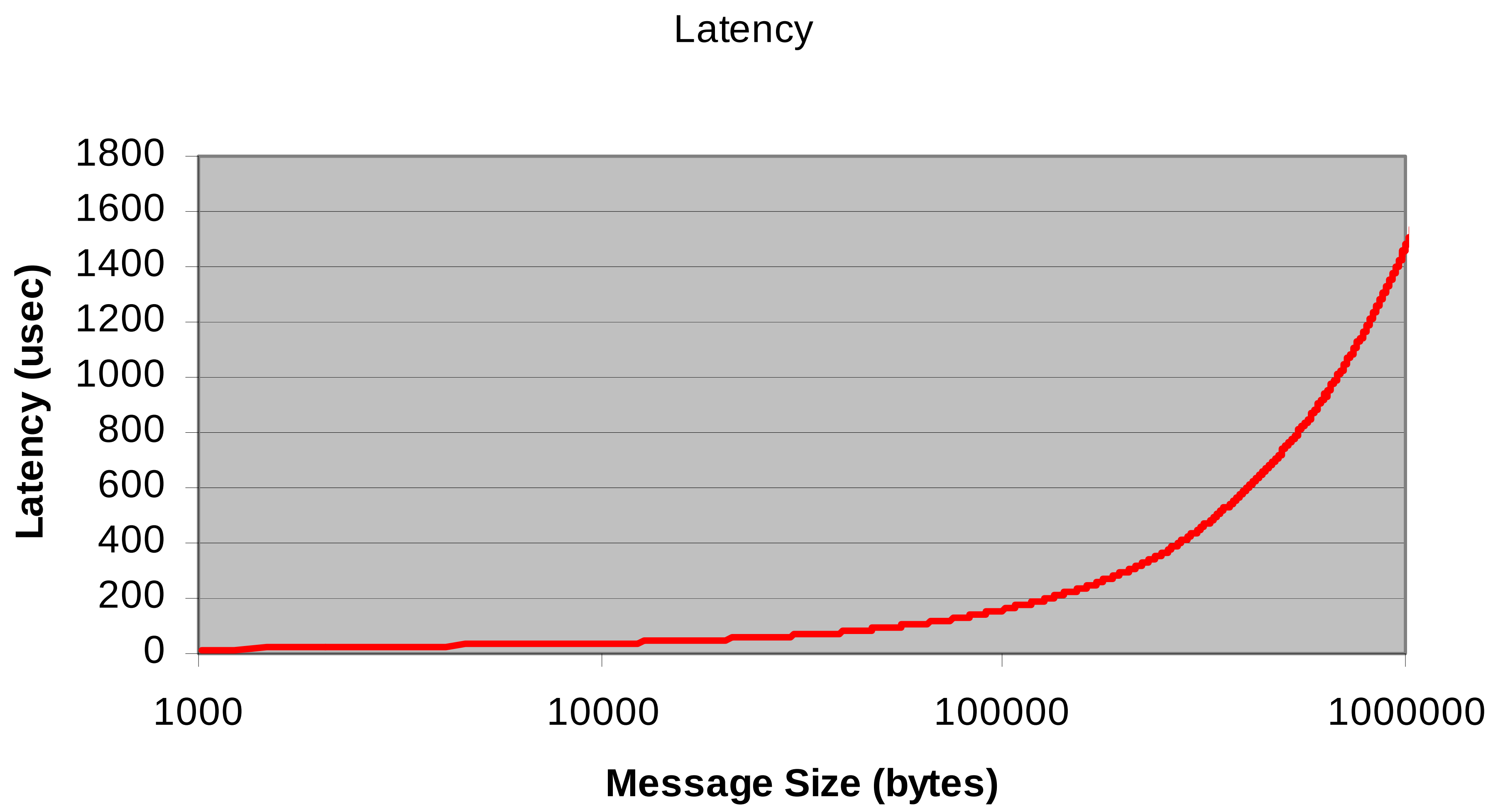
- Point to Point
- 1 to N
- All to All

- Linux kernel 2.8.6
- MVAPICH 0.9.2
- OpenIB gen1 revision 279
- Mellanox Cougar HCAs (PCI-X)
- Mellanox 16 port InifiniScale switch (6-chip)

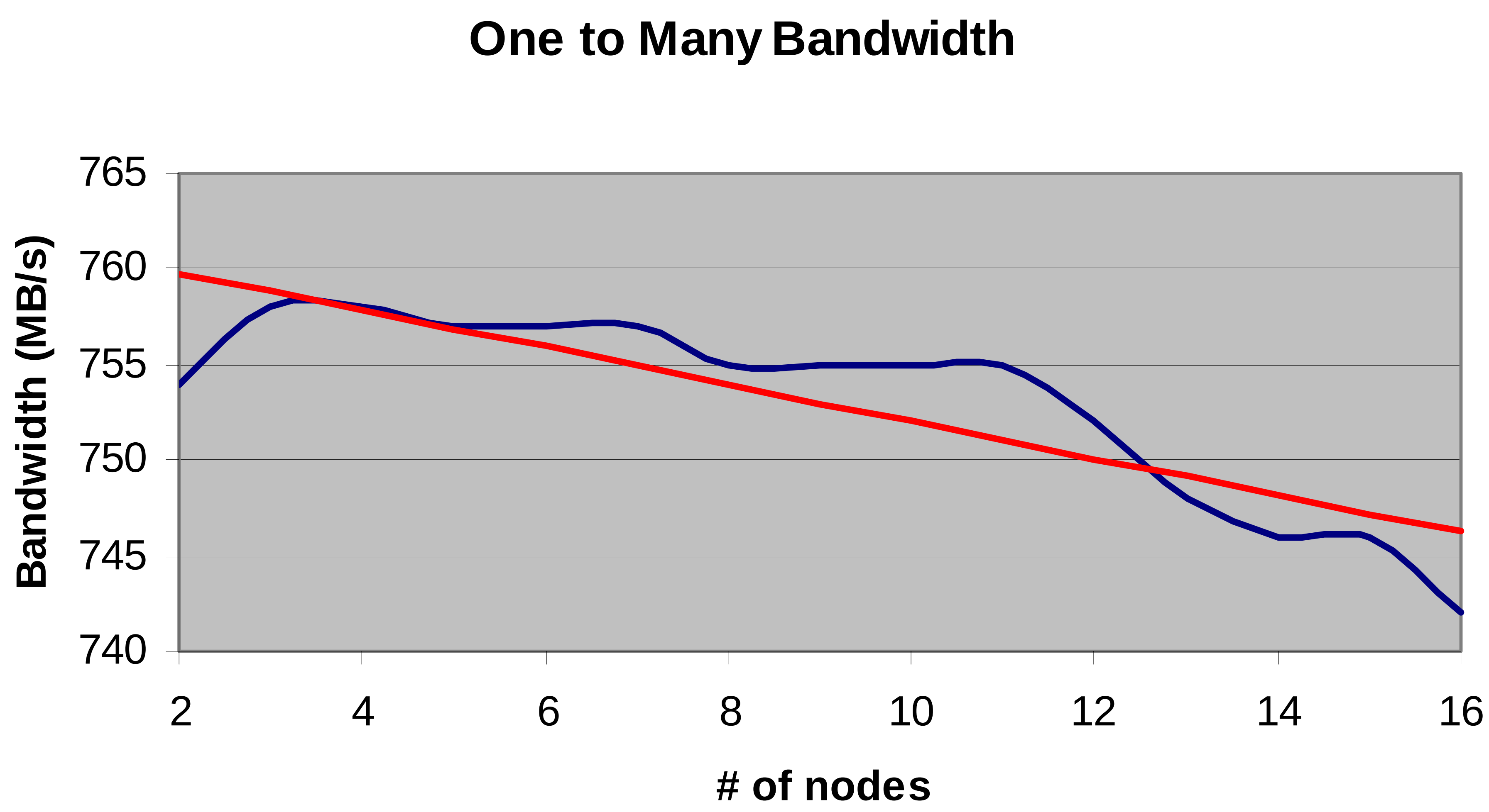
Point to Point



Point to Point

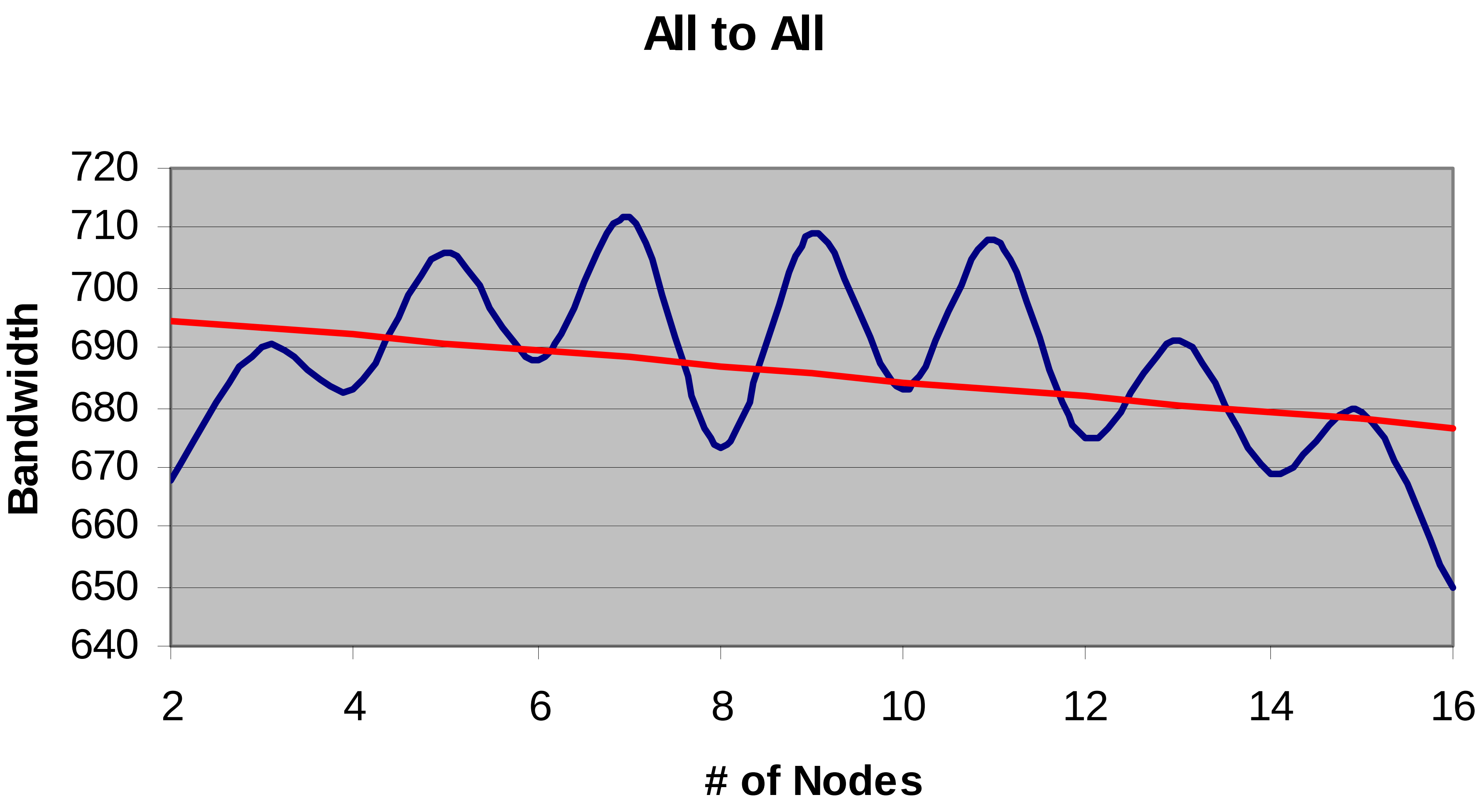


One to Many



One to Many Bandwidth
765
760
755
750
745
740
Bandwidth (MB/s)
2
4
6
8
10
12
14
16
# of nodes

All to All


All to All
720
710
700
690
680
670
660
650
640
Bandwidth
2
4
6
8
10
12
14
16
# of Nodes

Network Summary

- Peak Bandwidth (point to point)
  - 755MB/s
  - 256KB message size
- Cross sectional bandwidth
  - 10.8 GB/s
- Latency
  - 12usec for small messages (we've seen as good as 6usec)
  - Linear scaling for larger messages
- What's causing the falloff?
  - Driver issues
  - Kernel issues
  - Firmware issues

OpenIB – The Good

- Open source
- Lots of users
- Gen1 performance on par with Mellanox stack
- Gen2 looking good
- MPI works great with gen1! (OSU MVAPICH)
- SDP
  - Fast porting of TCP based code
- Vendor “independent”
- Aiming for Linux kernel inclusion

OpenIB – The Bad

- It's yet to be generally stable/usable
- OpenSM
  - Too hard to use
  - Make sure to get a switch with an SM
- Connection management
- CPU overhead
- Aiming for Linux kernel inclusion
- VAPI
  - No documentation
  - EVAPI vs. VAPI
  - Multiple connection issues

OpenIB – And The Ugly…

- Vendor agendas
  - Topspin vs. Voltaire vs. Infinicon
  - Lots of bickering
- Roadmap
  - Too many tangents
  - Doesn't match up with distributions
    - RHEL?
    - FC3?
- Why can't we just get the basics working before we move on?!?!
  - Gen2 will hopefully do this
  - SDP missing…