



# Benchmarking Clusters with High Core-Count Nodes

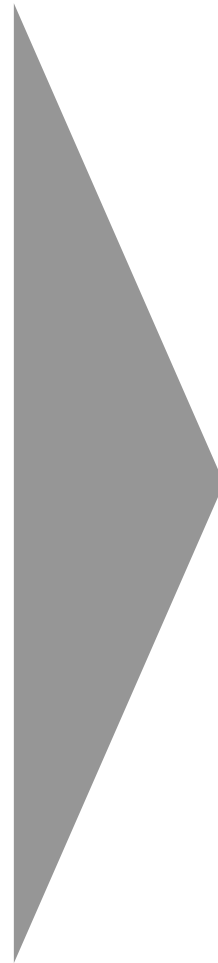
9<sup>th</sup> LCI International Conference on High-  
Performance Clustered Computing  
29 April, 2008

Tom Elken  
Manager, Performance Engineering  
QLogic Corporation

- **The most common interconnect benchmarks exercise one core on each of two nodes**
- **Nodes are becoming larger in terms of core-count.**
  - The interconnect needs to keep pace
  - What benchmarks are appropriate in this environment?

- **QLogic Background**
- **InfiniBand Host Channel Adapters**
- **MPI Benchmarks for  
High core-count nodes & clusters**

# QLogic Acquired Two Leading InfiniBand Companies in 2006





- **Application performance steered development**
  - SPEC HPG member and contributor to MPI2007
  - InfiniBand DDR fabric for bandwidth-intensive apps
  - QLogic InfiniPath HCAs for latency and message-rate sensitive apps
  
- **Two paths to the interconnect**
  1. OpenFabrics driver and libraries interoperate with the InfiniBand ecosystem and high-performance storage
  2. QLogic-specific libraries evolve as a top-down software stack; leverages ASIC capabilities specifically for HPC Apps and parallel programming models.

# QLogic InfiniBand HCAs

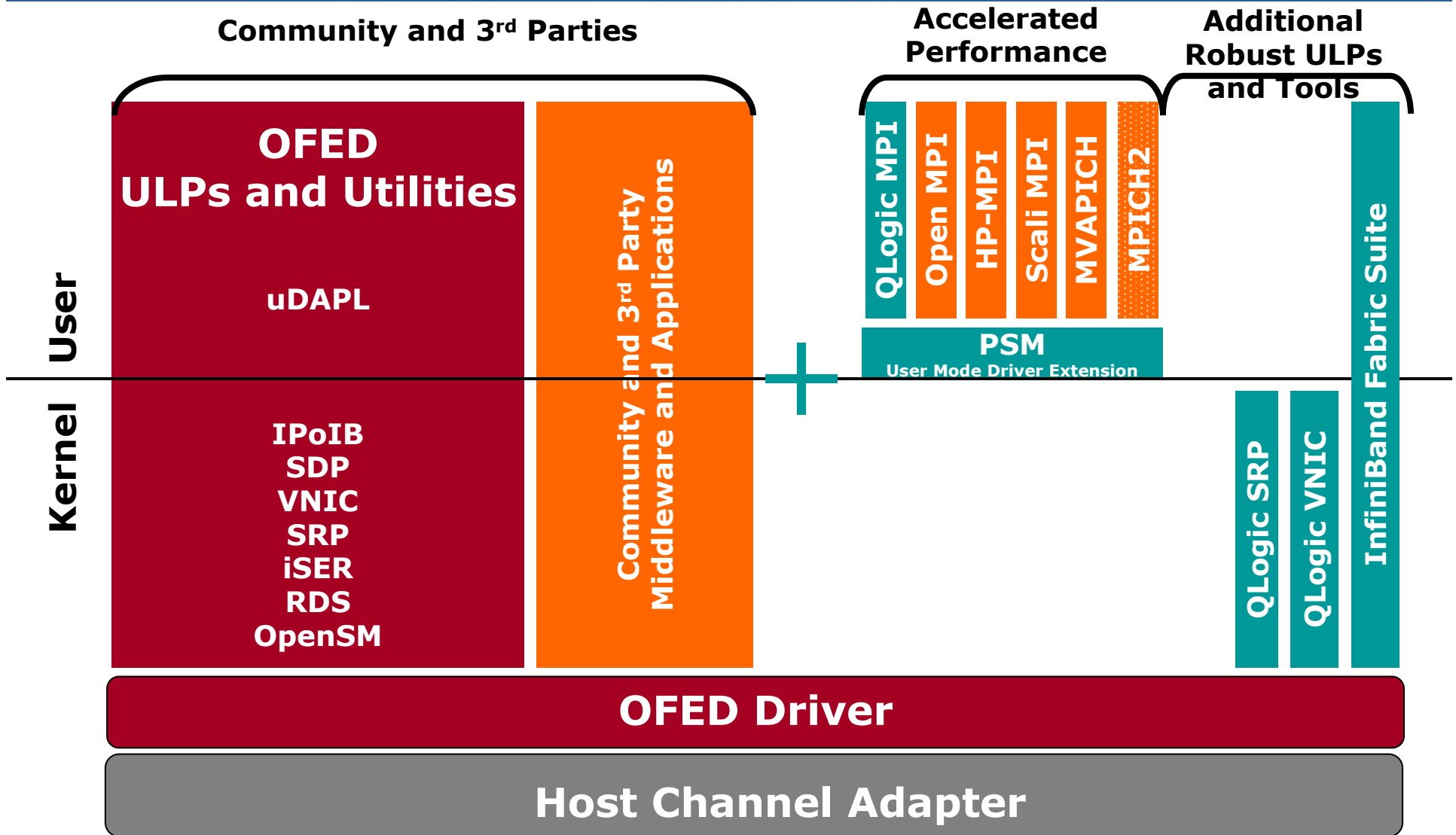


## ■ In this presentation

- QLogic DDR (20 Gbit/s) HCAs
  - QLE7240, PCIe x8
  - QLE7280, PCIe x16
- QLogic SDR (10 Gbit/s) HCA
  - QLE7140, PCIe x8



# OFED+ Provides OpenFabrics Benefits Plus Optional Value Added Capabilities



# What is the spectrum of MPI benchmarks?



- **Microbenchmarks include (there are more):**
  - OSU MPI Benchmarks (OMB)
  - Intel MPI Benchmarks (IMB) formerly Pallas
- **Mid-level Benchmarks**
  - HPC Challenge
  - Linpack, e.g. HPL
  - NAS Parallel Benchmarks (NPB)
- **Application Benchmark Suites**
  - SPEC MPI2007
  - TI-0n (TI-06, TI-07, TI-08) DOD benchmarks
- **Your MPI application**

# OSU MPI Benchmarks applicable to high-core-count nodes



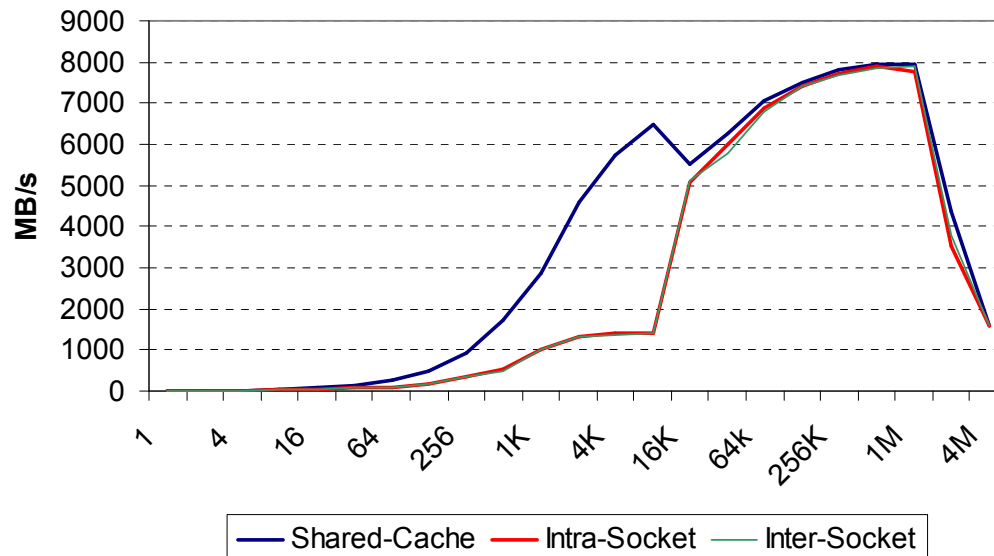
- **The MVAPICH group publishes results on one node**
  - Intra-node MPI performance importance growing as nodes grow their core-counts
  - Pure MPI applications under some competition from more complex hybrid OpenMP – MPI styles of development
- **OMB includes the Multiple Bandwidth, Message Rate test**
- **OMB v3.1 has added a benchmark: Multiple Latency test (osu\_multi\_lat.c)**

# Intra-node MPI Bandwidth measurement



- Most current MPIs use shared-memory copies for intra-node communications – might expect that they all do equally well
- QLogic PSM 2.2 provides a large improvement in intra-node bandwidth; All MPIs it supports (MVAPICH, Open MPI, HP-MPI, Scali, QLogic) get the benefit.
  - Results in a 2% improvement in average applications performance (SPEC MPI2007 on 8x 4-core nodes)

**MPI Intra-node Bandwidth (osu\_bw)**



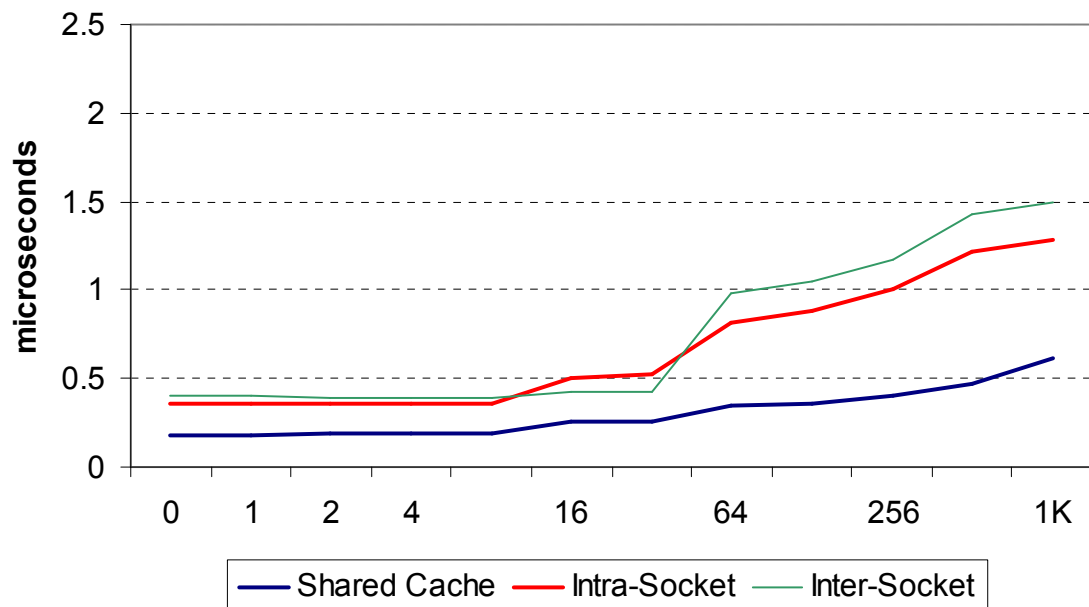
Measured on one node w/ Xeon E5410  
2.33 GHz quad-core CPUs, 8-cores;  
QLogic InfiniPath 2.2 software

# Intra-node MPI Latency measurement



- Most current MPIs use shared-memory copies for intra-node communications – might expect that they all do equally well
- QLogic PSM 2.2 provides a nice improvement in intra-node latency; All MPIs it supports (MVAPICH, Open MPI, HP-MPI, Scali, QLogic) get the benefit.

## MPI Intra-node Latency



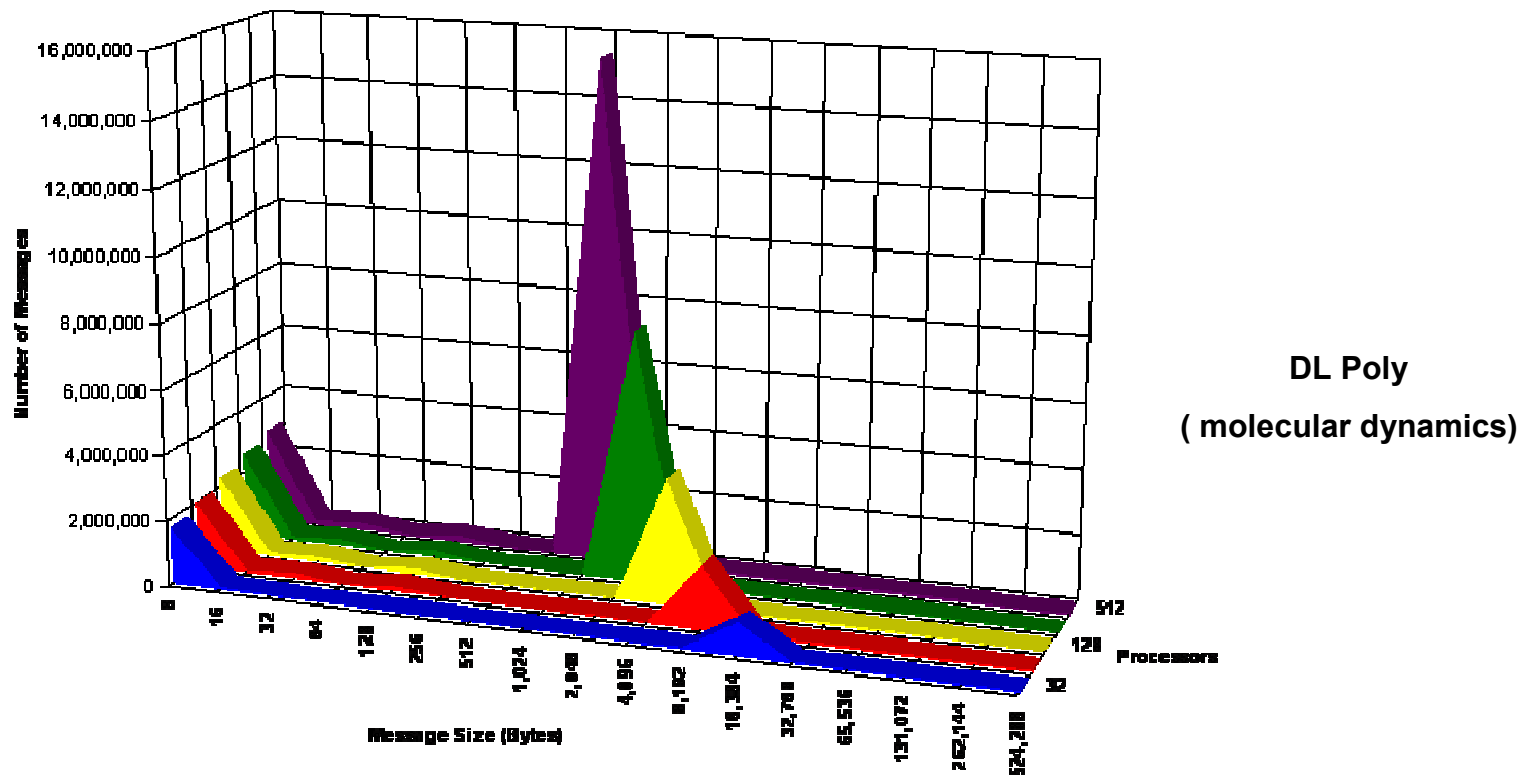
Lower is better

Measured on one node w/ Xeon E5472  
3.0 GHz quad-core CPUs, 8-cores;  
QLogic InfiniPath 2.2 software

# Relationship of applications to micro-benchmarks



- **As the number of processors is increased:**
  - Message size goes down (→ small-message latency)
  - Number of messages goes up (→ message rate)

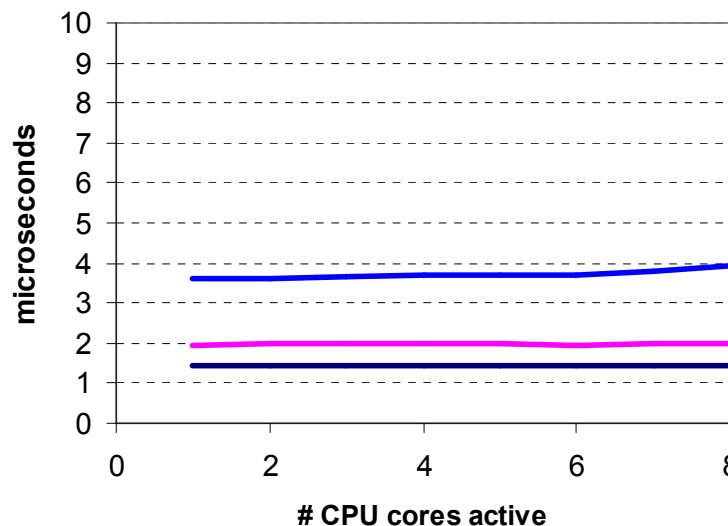


# New OSU Multiple Latency Test

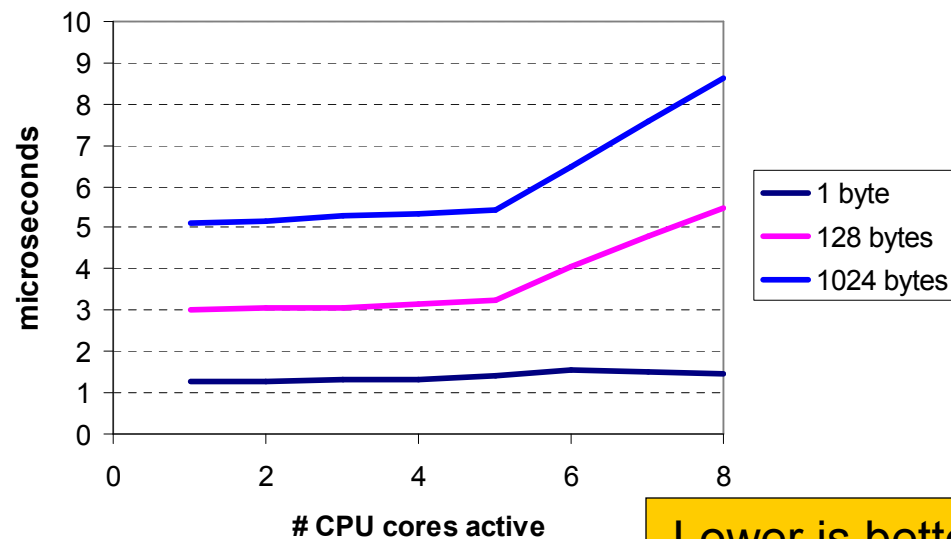


- Measure avg. latency as you add active cores running the latency benchmark in parallel
- Interesting to measure on large core-count nodes, and at multiple small message sizes ...

Average Latency (QLogic IB DDR)



Average Latency (Other IB DDR)



Lower is better

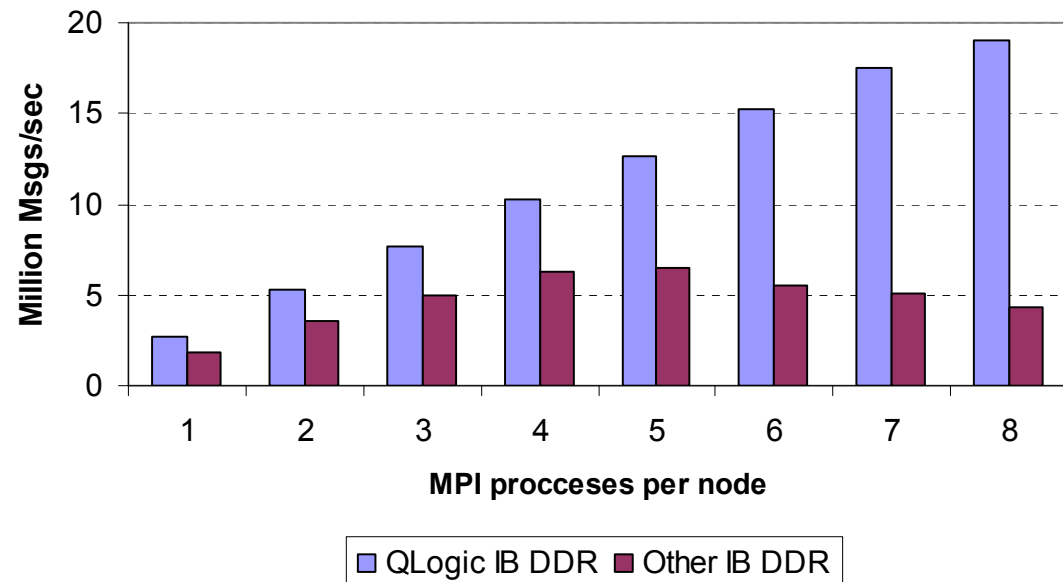
Measured on 2x Xeon E5410 2.33 GHz quad-core CPUs, 8-cores per node; both PCIe x8, DDR InfiniBand Adapters; benchmark is osu\_multilat.c

# MPI Message Rate



- Using `osu_mbw_mr` to measure Message Rate:
  - Measure at several processes per node counts
  - See if results scale with additional processes per node

MPI Message Rate (8 cores per node)



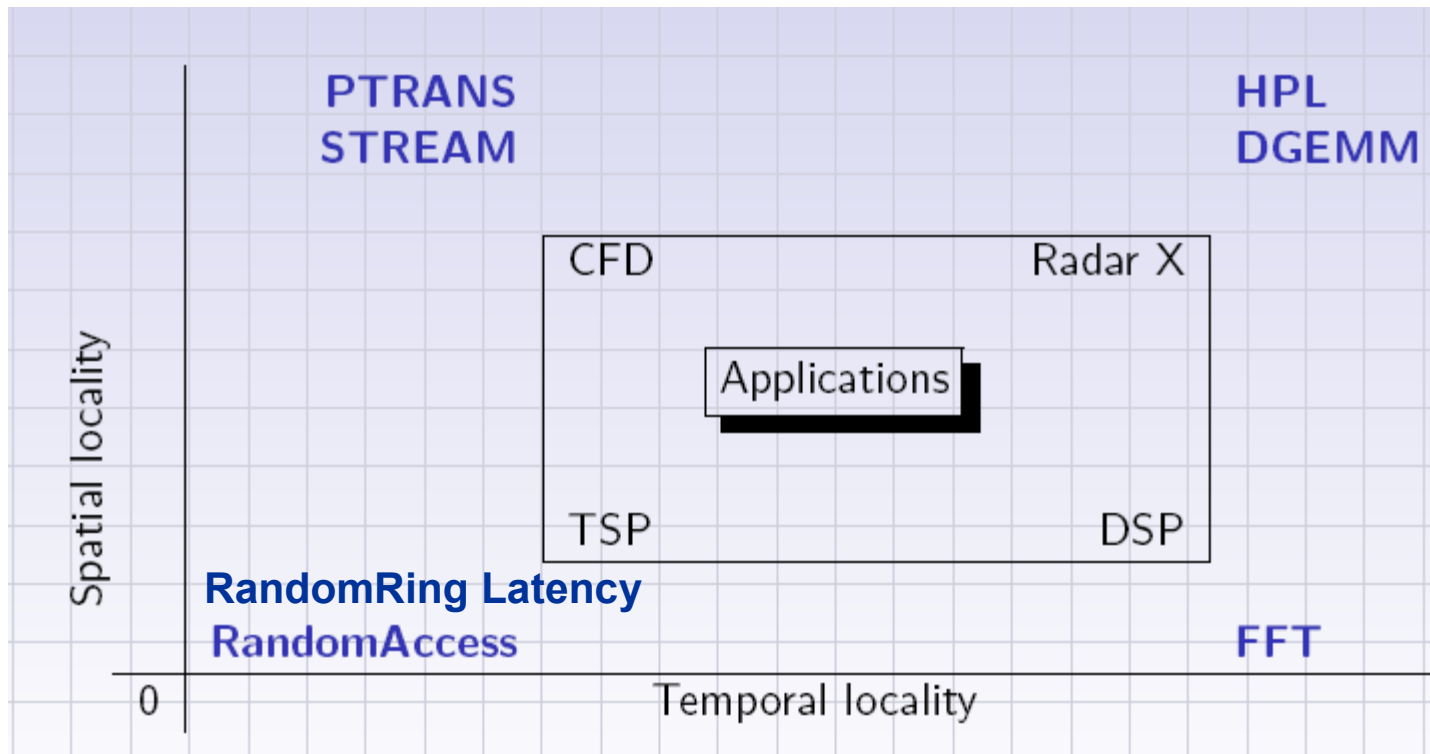
Higher is better

Measured on 2x Xeon E5410 2.33 GHz quad-core CPUs, 8-cores per node;  
both PCIe x8, DDR InfiniBand Adapters

# HPC Challenge Overview



- HPC Challenge component benchmarks are intended to test very different memory access patterns



Source: "HPC Challenge Benchmark," Piotr Luszczek, University of Tennessee Knoxville, SC2004, November 6-12, 2004, Pittsburgh, PA

# Relationship of HPC Challenge to Point-to-Point benchmarks



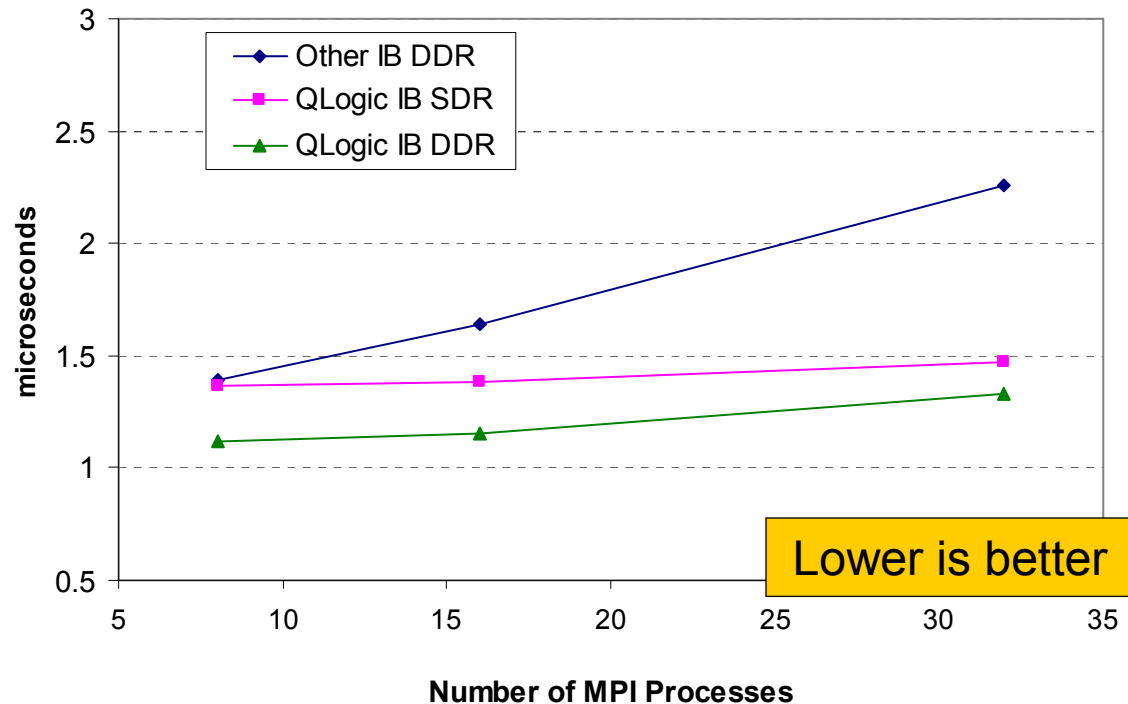
- **Are there benchmarks in HPCC that focus on latency, bandwidth & message rate but involve more of the cluster than 2 cores on 2 nodes, or 16 cores on 2 nodes?**
  - Latency: Random Ring Latency
  - Bandwidth: PTRANS and  
Random Ring Bandwidth
  - Message Rate: MPI Random Access

# Scalable Latency: Number of cores



- QLogic HCAs' MPI latency scales better with larger clusters
- QLogic InfiniBand DDR advantage over Other InfiniBand DDR is
  - 70% at 8 nodes, 32 cores

Comparison of HPCC RandomRing Latency



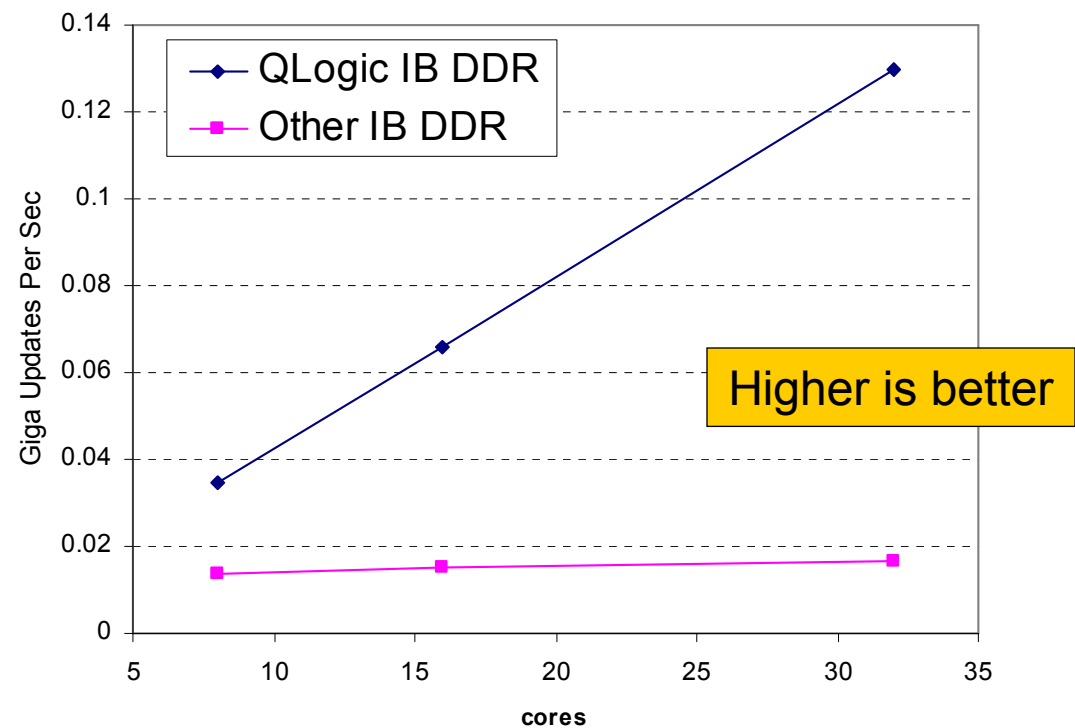
From a QLogic white paper: "Understanding Low and Scalable Message Passing Interface Latency" available at <http://www.qlogic.com/EducationAndResources/WhitePapersResourceLibraryHpc.aspx>

# Scalable Message Rate: GUPS



- QLogic QLE7200 series HCAs Message Rates scales better
- 8x faster than Other InfiniBand DDR HCA at 32 cores (and advantage climbing)
- GUPS = Giga (Billion) Updates Per Second are a measure of the rate at which messages can be sent to/from random cluster nodes/cores.

HPCC's MPI\_RandomAccess\_GUPS



\* QLogic measurements, one QLogic SilverStorm 9024 DDR switch, 8 Systems: 2 x 2.6 GHz Opteron 2218 CPUs, 8 GB DDR2-667 memory; NVIDIA MCP55 PCIe chipset. Base MPI RandomAccess source code used – no source tweaks to reduce # of messages sent.

# Application Benchmarks



# Application Benchmarks: SPEC MPI2007



- **Suite that measures: CPU, memory, interconnect, MPI, compiler, and file system performance.**
- **SPEC institutes discipline and fairness in benchmarking:**
  - Rigorous run rules
  - All use same source code, or performance-neutral alternate sources
  - Disclosure rules: system, adapter, switch, firmware, driver, compiler optimizations, etc.
  - Peer review of submissions
  - Therefore, more difficult to game

# SPEC MPI2007 Benchmarks 1- 6



<b>Benchmark</b>	<b>Language</b>	<b>Application Area</b>	<b>Brief Description</b>
<b>104.milc</b>	<b>C</b>	<b>Quantum Chromodynamics</b>	A gauge field generating program for lattice gauge theory programs with dynamical quarks
<b>107.leslie3d</b>	<b>Fortran</b>	<b>Computational Fluid Dynamics</b>	CFD using Large-Eddy Simulations with linear-eddy mixing model in 3D.
<b>113.GemsFDTD</b>	<b>Fortran</b>	<b>Computational Electromagnetics</b>	Solves the Maxwell equations in 3D using the finite-difference time-domain (FDTD) method
<b>115.fds4</b>	<b>Fortran</b>	<b>CFD: Fire dynamics simulator</b>	A CFD model of fire-driven fluid flow, with an emphasis on smoke and heat transport from fires
<b>121.pop2</b>	<b>Fortran/C</b>	<b>Climate Modeling</b>	The Parallel Ocean Program (POP) developed at LANL
<b>122.tachyon</b>	<b>C</b>	<b>Graphics: Ray Tracing</b>	A nearly E.P. parallel ray tracing program with low MPI usage

# SPEC MPI2007 Benchmarks 7- 13



Benchmark	Language	Application Area	Brief Description
<b>126.lammps</b>	<b>C++</b>	<b>Molecular Dynamics</b>	a classical molecular dynamics simulation code designed for parallel computers
<b>127.wrf2</b>	<b>C/Fortran</b>	<b>Weather Forecasting</b>	Code is based on the Weather Research and Forecasting (WRF) Model
<b>128.GAPgeofem</b>	<b>C/Fortran</b>	<b>Heat Transfer using FEM</b>	A parallel finite element method (FEM) code for transient thermal conduction with gap radiation
<b>129.tera_tf</b>	<b>Fortran</b>	<b>3D Eulerian Hydrodynamics</b>	Code uses a 2 <sup>nd</sup> order Gudenov scheme and a 3 <sup>rd</sup> order remapping
<b>130.socorro</b>	<b>C/Fortran</b>	<b>Molecular Dynamics</b>	Molecular Dynamics using density-functional theory (DFT)
<b>132.zeusmp2</b>	<b>Fortran</b>	<b>Computational Astrophysics</b>	Performs various hydrodynamic simulations on 1, 2, and 3D grids
<b>137.lu</b>	<b>Fortran</b>	<b>Implicit CFD</b>	Solves a regular sparse block Lower- and Upper-triangular system using SSOR

# SPEC MPI2007 on the web



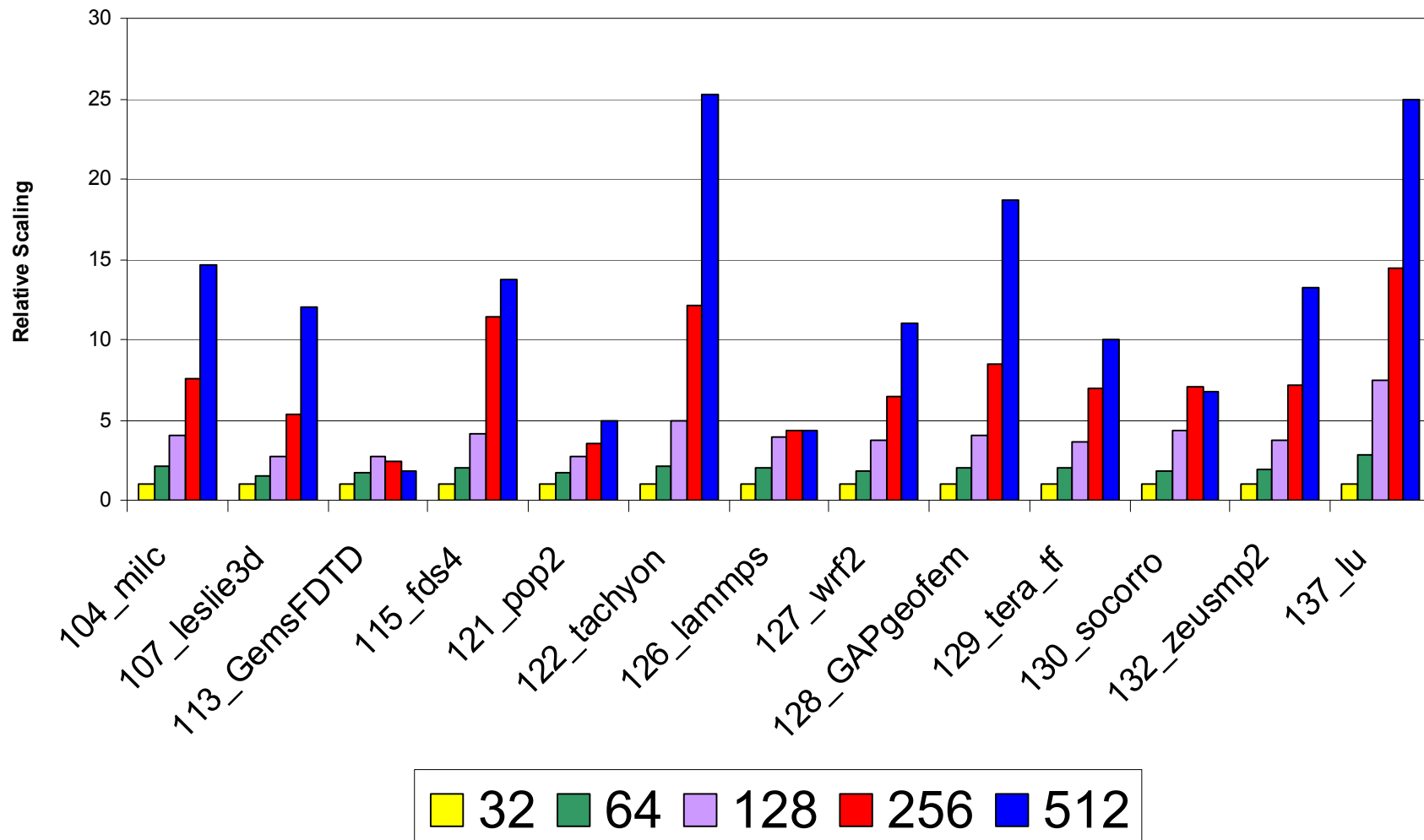
- Result score is an average of ratios for each of 13 codes: the ratio of the run time of a code on your system to the runtime on the reference platform (1<sup>st</sup> listed).

Test Sponsor	System Name	System Configuration				Results	
		MPI Ranks	Compute Threads Used	Compute Nodes Used	Compute Cores Enabled	Base	Peak
Advanced Micro Devices	A2210 ("Serenade") -- Reference Platform <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	16	16	8	16	0.999	0.999
Hewlett-Packard Company	HP Proliant BL460c blade Cluster Platform 3000BL <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	128	128	32	128	11.9	Not Run
Hewlett-Packard Company	HP Proliant BL460c blade Cluster Platform 3000BL <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	256	256	64	256	19.8	Not Run
Hewlett-Packard Company	HP Proliant BL460c blade Cluster Platform 3000BL <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	64	64	16	64	6.39	Not Run
Hewlett-Packard Company	HP Proliant BL460c blade Cluster Platform 3000BL <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	32	32	8	32	3.40	Not Run
Hewlett-Packard Company	HP Proliant BL460c blade Cluster Platform 3000BL <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	16	16	4	16	1.75	Not Run
Intel Corporation	Endeavor <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	256	256	32	256	18.5	Not Run
Intel Corporation	Endeavor <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	32	32	4	32	3.05	Not Run
Intel Corporation	Endeavor <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	64	64	8	64	6.21	Not Run
Intel Corporation	Endeavor <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	128	128	16	128	11.6	Not Run

# Scaling with SPEC MPI2007



## Scaling by application to 512 Cores



- **The best benchmark is “your application,”** particularly since they are usually run on all cores of the nodes used on the job.
- **There is a range of MPI benchmarks because they all have their place:**
  - microbenchmarks are easier, quicker to run and are evolving to test multi-core nodes
  - application benchmarks are a bit more difficult to run, but are a better predictor of performance across a range of applications
- **Benchmarks are evolving to serve the needs of ever-expanding multi-core systems**