Red Storm / Cray XT4: A Superior Architecture for Scalability

Mahesh Rajan, Doug Doerfler, Courtenay Vaughan
Sandia National Laboratories, Albuquerque, NM

Cray User Group
Atlanta, GA; May 4-9, 2009

Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

MOTIVATION

- Major trend in HPC system architecture: use of commodity multi-socket multi-core nodes with InfiniBand Interconnect
- DOE under the ASC Tri-Lab Linux Capacity Cluster (TLCC) program has purchased 21 “Scalable Units” (SUs). Each SU consists of 144 four-socket, quad-core AMD Opteron (Barcelona) nodes, using DDR InfiniBand as the high speed interconnect
- Red Storm/Cray XT4 at Sandia was recently upgraded: the 6240 nodes in the ‘center section’ were upgraded to a similar AMD Opteron quad-core processor (Budapest)
- Comparison of performance between Red Storm and TLCC reveals a wealth of information about HPC architectural balance characteristics on application scalability
- The best TLCC performance used for comparisons; Often required *NUMACTL*
- The benefits of the superior architectural features of Cray/XT4 are analyzed through several benchmarks and applications

Presentation Outline

- Overview of current Red Storm/Cray XT4 system at Sandia
- Overview of the Tri-Lab Linux Capacity Cluster -TLCC
- Architectural similarities and differences between the two systems
- Architectural balance ratios
- Micro-benchmarks
    - Memory latency
    - Memory bandwidth
    - MPI Ping-Pong
    - MPI Random and Bucket-Brigade
- Mini-Applications
    - Mantevo-HPCCG
    - Mantevo-phdMesh
- SNL Applications
    - CTH – Shock hydrodynamics
    - SIERRA/Presto – Explicit Lagrangian mechanics with contact
    - SIERRA/Fuego – Implicit multi-physics Eulerian mechanics
    - LAMMPS – Molecular dynamics

Red Storm Architecture

284.16 TeraFLOPs theoretical peak performance

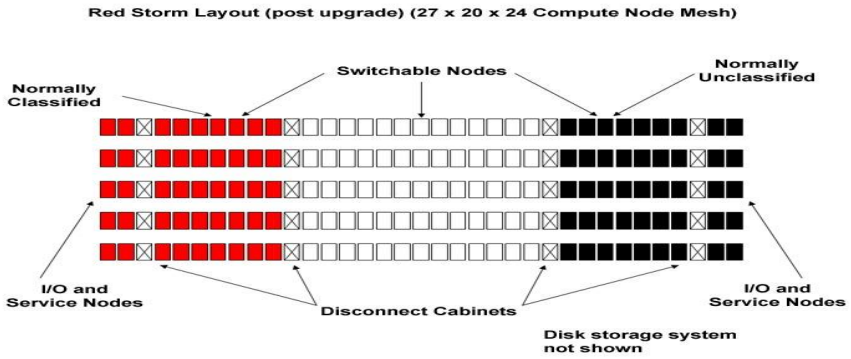
135 compute node cabinets, 20 service and I/O node cabinets, and 20 Red/Black switch cabinets

640 dual-core service and I/O nodes (320 for red, 320 for black)

12,960 compute nodes (dual-core and quad-core nodes) = 38,400 compute cores

6720 Dual-Core nodes with AMD Opteron™ processor 280
- 2.4 GHz
- 4 GB of DDR-400 RAM
- 64 KB L1 instruction and data caches on chip
- 1 MB L2 shared (Data and Instruction) cache on chip
- Integrated Hyper Transport2 Interfaces

6240 Quad-Core nodes with AMD Opteron™ processor Budapest
- 2.2 GHz
- 8 GB of DDR2-800 RAM
- 64 KB L1 instruction and 64KB L1 data caches on chip per core
- 512 KB L2 Cache per core
- 2 MB L3 shared (Data and Instruction) cache on chip
- Integrated Hyper Transport3 Interfaces

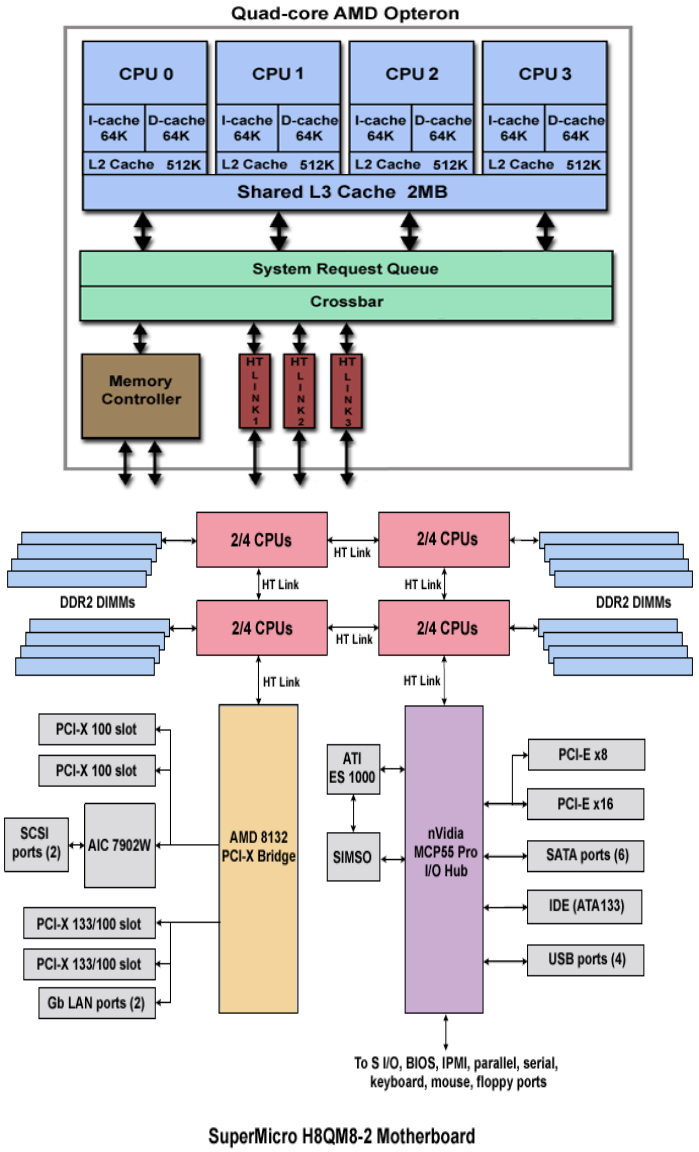


TLCC Overview

SNL 's TLCC

38 TeraFLOPs theoretical peak performance
2 Scalable Units (SUs)
288 total nodes
272 quad-socket, quad-core compute nodes
- 4,352 compute cores
- 2.2 GHz AMD Opteron Barcelona
- 32 GB DDR2 -667 RAM per node
- 9.2 TB total RAM
- 64 KB L1 instruction and 64KB L1 data caches on chip per core
- 512 KB L2 Cache per core
- 2 MB L3 shared (Data and Instruction) cache on chip
- Integrated dual DDR memory controllers
- Integrated Hyper Transport3 Interfaces

Interconnect: InfiniBand with OFED stack
InfiniBand card: Mellanox ConnectX HCA



SuperMicro H8QM8-2 Motherboard

Architectural Comparison

Name	Cores/Node	Networlk/Topology	Total nodes(N)	Clock (GHz)	Mem/core & Speed	MPI Inter Node Latency (usec)	MPI Inter Node BW (GB/s)	Stream BW (GB/s/Node)	Memory Latency (clocks)
Red Storm (dual)	2	Mesh/ Z Torus	6,720	2.4	2GB; DDR-400MHz	4.8	2.04	4.576	119
Red Storm (quad)	4	Mesh/ Z Torus	6, 240	2.2	2GB; DDR2-800MHz	4.8	1.82	8.774	90
TLCC	16	Fat-tree	272	2.4	2GB; DDR2-667 MHz	1.0	1.3	15.1	157

Node Balance Ratio Comparison

	MAX Bytes-to-FLOPS Memory	MAX Bytes-to-FLOPS Interconnect	MIN Bytes-to-FLOPS Memory	MIN Bytes-to-FLOPS Interconnect
Red Storm (dual)	0.824	0.379	0.477	0.190
Red Storm (quad)	0.756	0.232	0.249	0.058
TLCC	0.508	0.148	0.107	0.009
Ratio: Quad/ TLCC	1.49	1.57	2.33	6.28

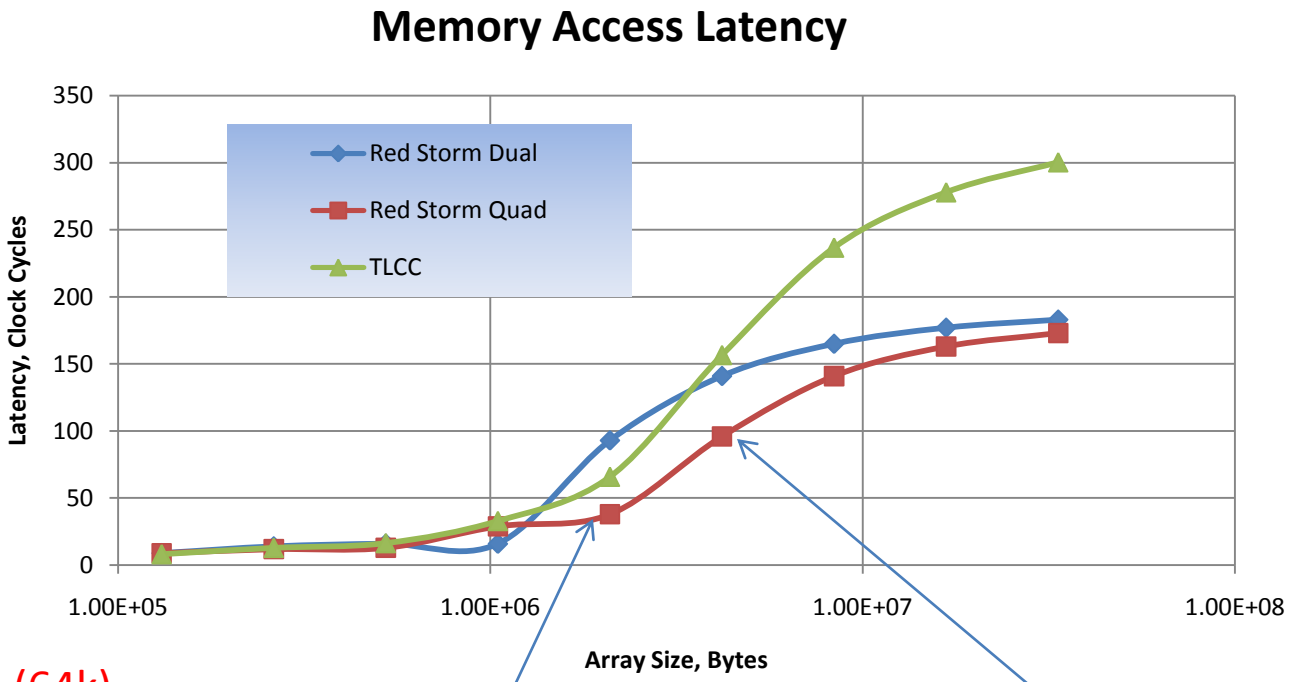
Bytes-to-FLOPS Memory = Stream BW (Mbytes/sec)/ Peak MFLOPS

Bytes-to-FLOPS Interconnect = Ping-Pong BW (Mbytes/sec)/ Peak MFLOPS

MAX = using single core on node

MIN = using all cores on a node

Micro-Benchmark: Memory Latency-single thread



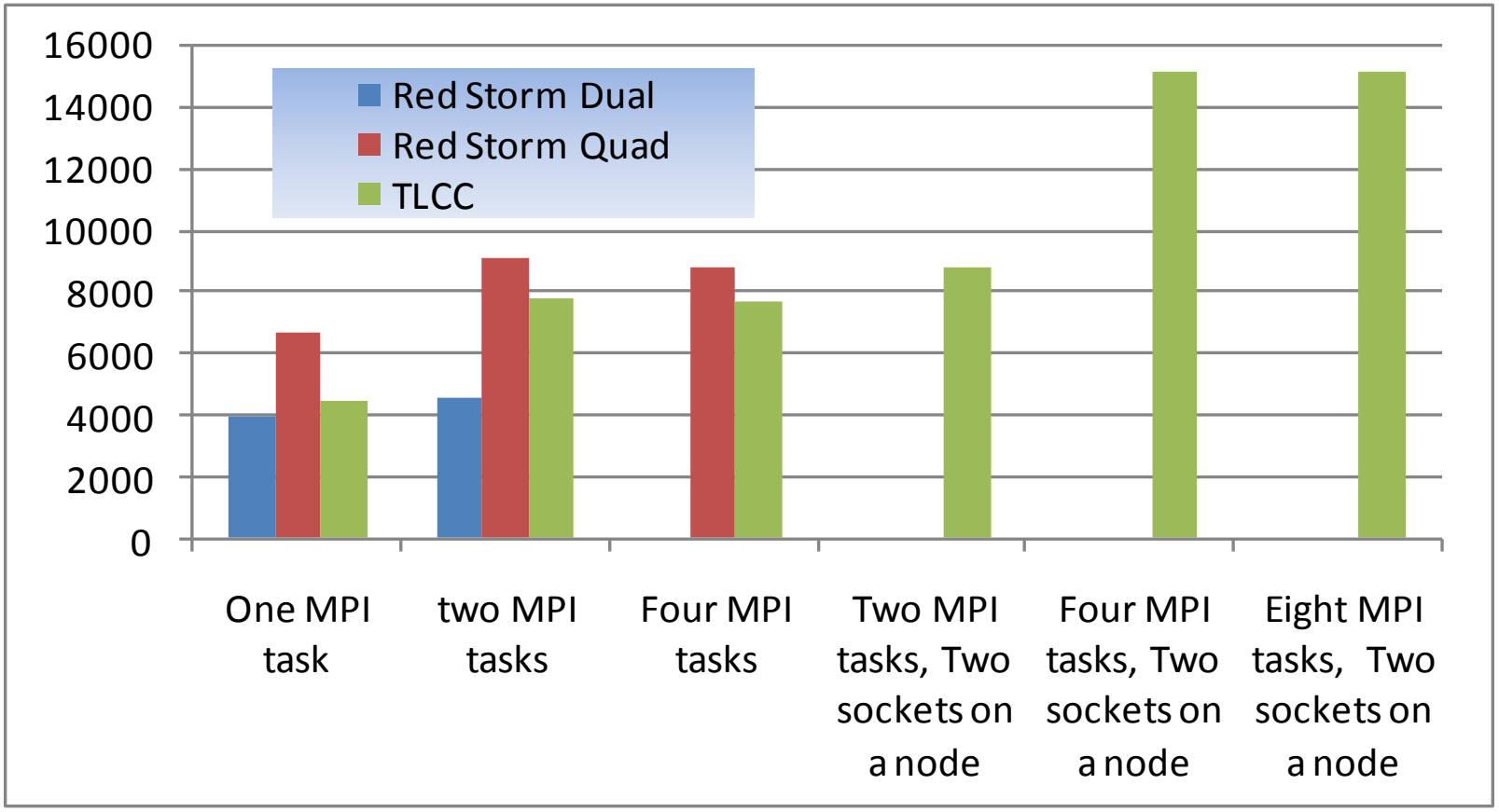
3 cycles L1 (64k)

15 cycles L2(512k)

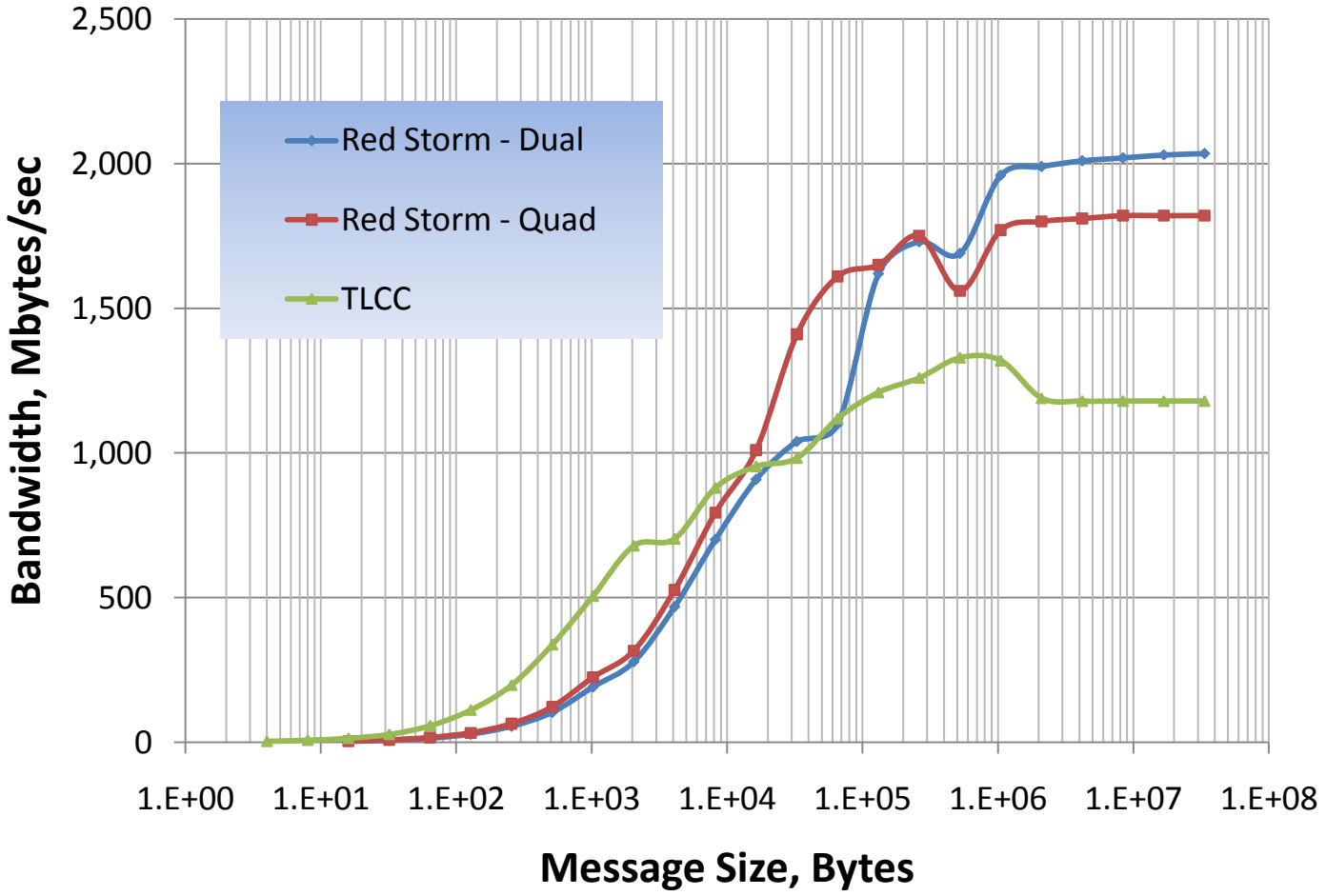
45 cycles shared L3(2MB)

90+ cycles RAM

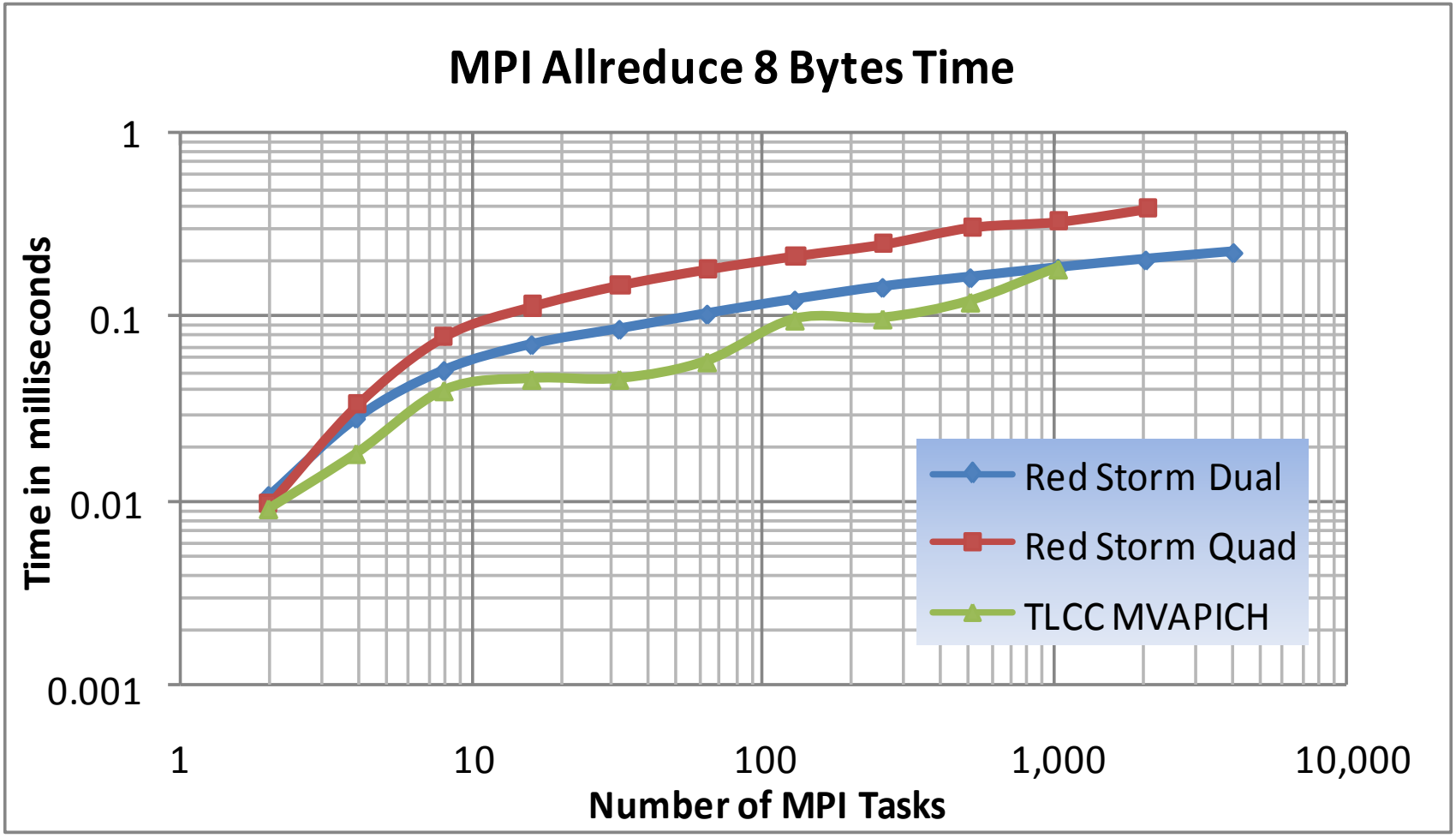
Micro-Benchmark: STREAMS Memory Bandwidth -MBytes/sec



Micro-Benchmark: MPI Ping-Pong



Micro-Benchmark: MPI Ping-Pong



MPI Random and Bucket-Brigade Benchmark Bandwidths in MBytes/sec

Random Message (RM) Sizes = 100 to 1K Bytes; Bucket Brigade Small (BBS) size = 8 bytes; Bucket Brigade Large (BBL) size= 1MB

	RM - 1024	RM-256	RM-64	BBS-1024	BBS-256	BBS-64	BBL-1024	BBL-256	BBL-64
Red Storm Dual	67.7	71.9	75.3	1.19	1.19	1.20	1100.2	1116.1	1132.4
Red Storm Quad	41.6	45.0	46.9	0.86	0.86	0.86	654.7	647.6	632.2
TLCC	0.43	1.59	3.64	1.77	3.37	3.42	275.47	314.3	344.1

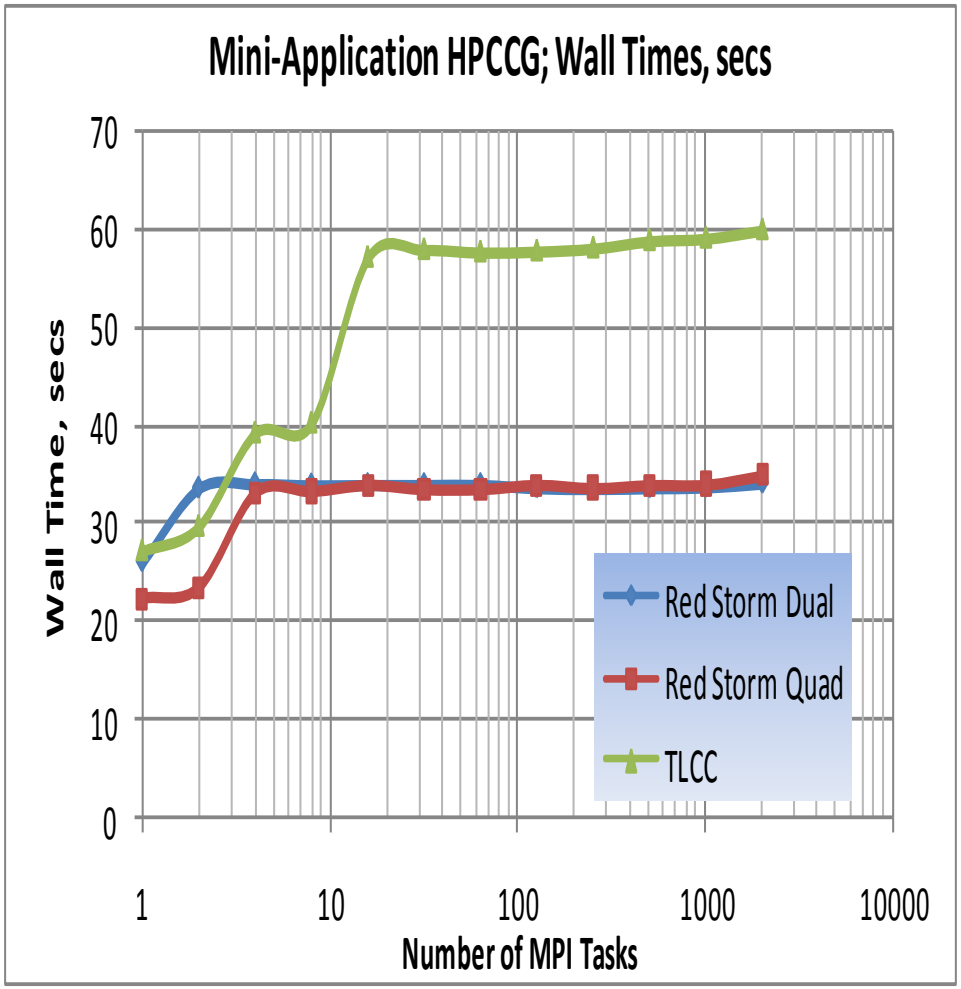
Note Big Difference between Red Storm and TLCC for Random Messaging Benchmark

Random BW Ratio Quad/TLCC @1024 = 97; Random BW Ratio Quad/TLCC @64 = 13

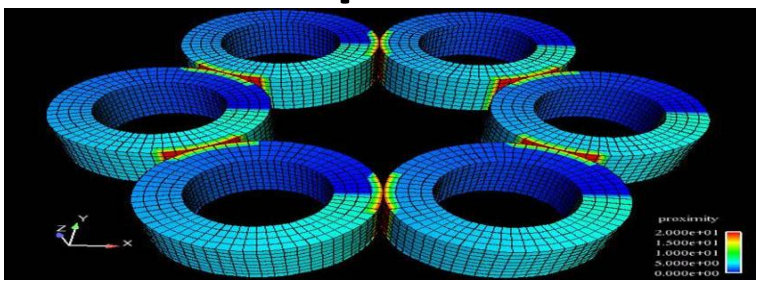
Mini-Application: Mantevo HPCCG

Illustrates Node Memory Architectural Impact

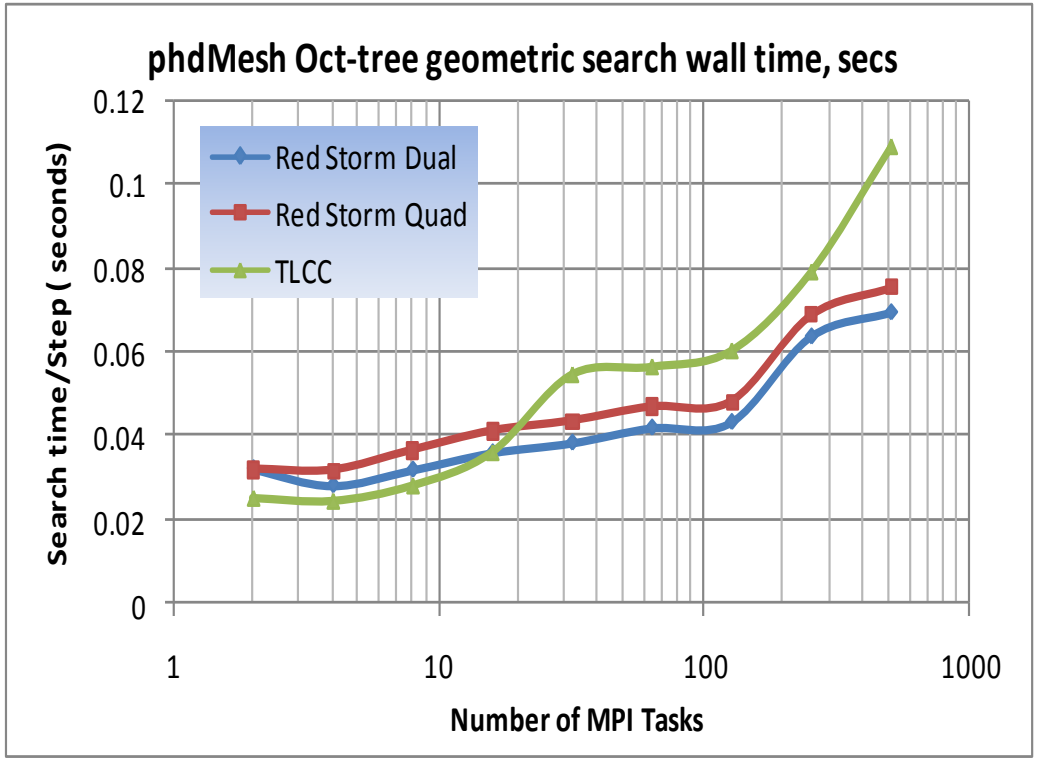
- Mike Heroux's Conjugate Gradient mini-application
- Coefficient matrix stored in sparse matrix format
- Most of the time dominated by sparse matrix vector multiplication
- Parallel overhead small fraction
- TLCC-16ppn runs show strong benefit of using *numactl* to set process and memory affinity bindings
- Once best performance within a node is achieved weak scaling curve is near perfect
- TLCC 2 to 4 MPI tasks 37% loss; 8 to 16 MPI tasks another 44% loss
- 1.7 X slower TLCC performance; This ratio approaches worst Quad/TLCC byte-to-FLOPS ratio of 2.3 discussed earlier



Mini-Application – Mantevo: phdMesh



- Benchmark used for research in contact detection algorithms
- Figure shows a weak scaling analysis using a grid of counter rotating gears: 4x3x1 on 2 PEs, 4x3x2 on 4 PEs, etc
- Search time / step of an oct-tree geometric proximity search detection algorithm is shown.
- TLCC shows quite good performance except at scale at 512 cores where it is about 1.4X slower than Red Storm Quad.

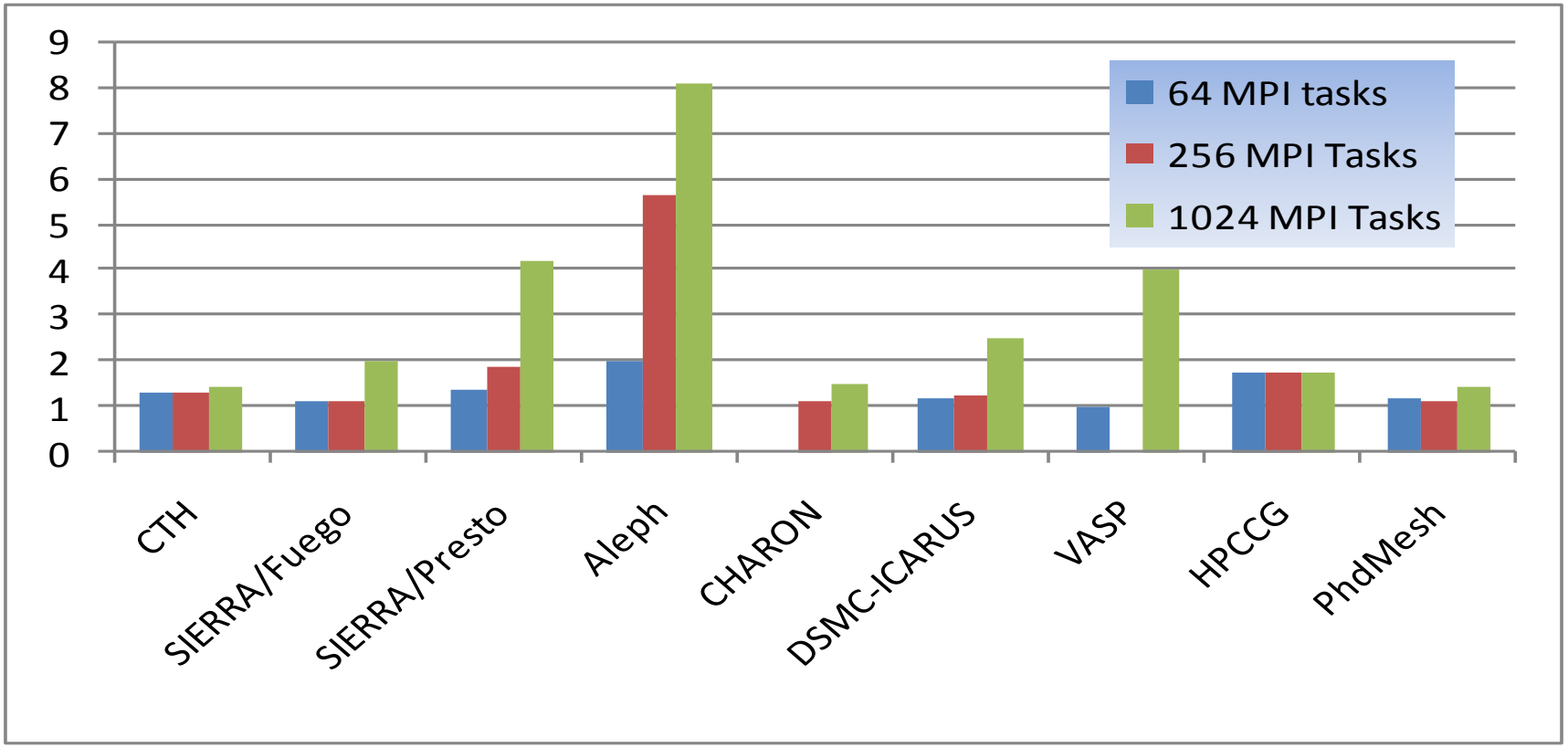


Red Storm, TLCC Application Performance Comparison

TLCC/Red Storm Wall Time Ratio

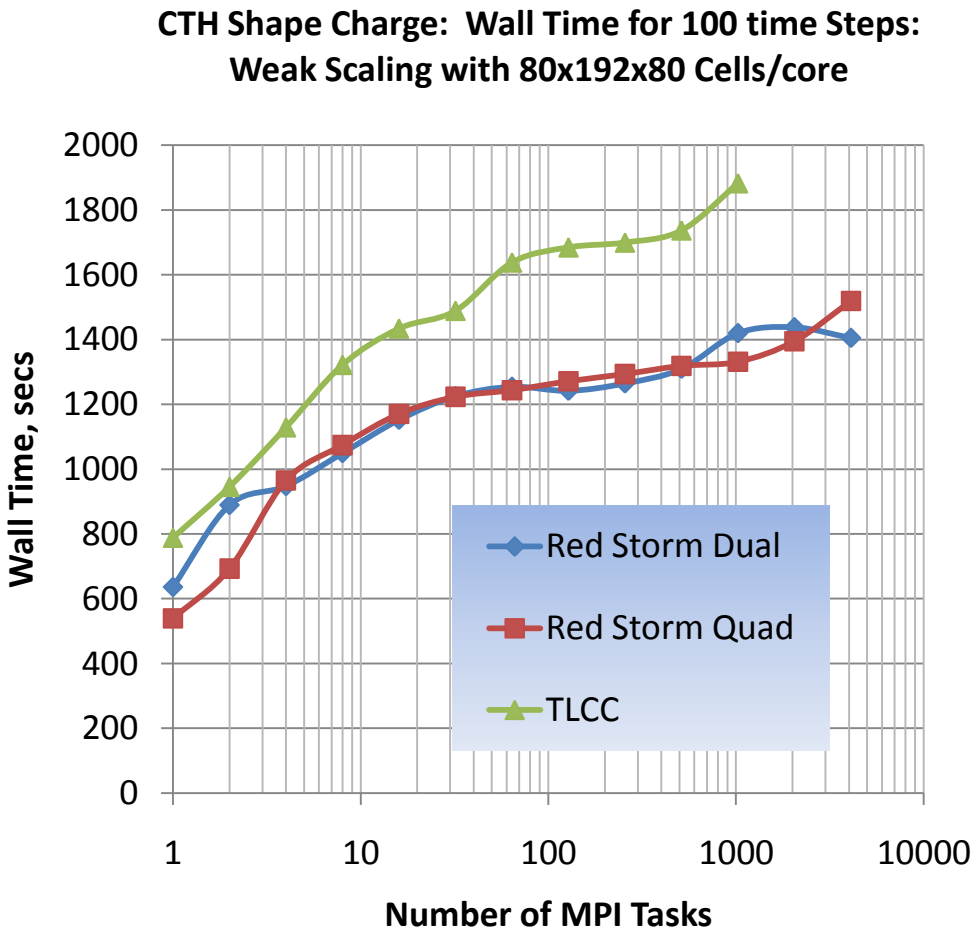
Ratio = 1, runs take the same time

Ratio =2, TLCC takes twice as long



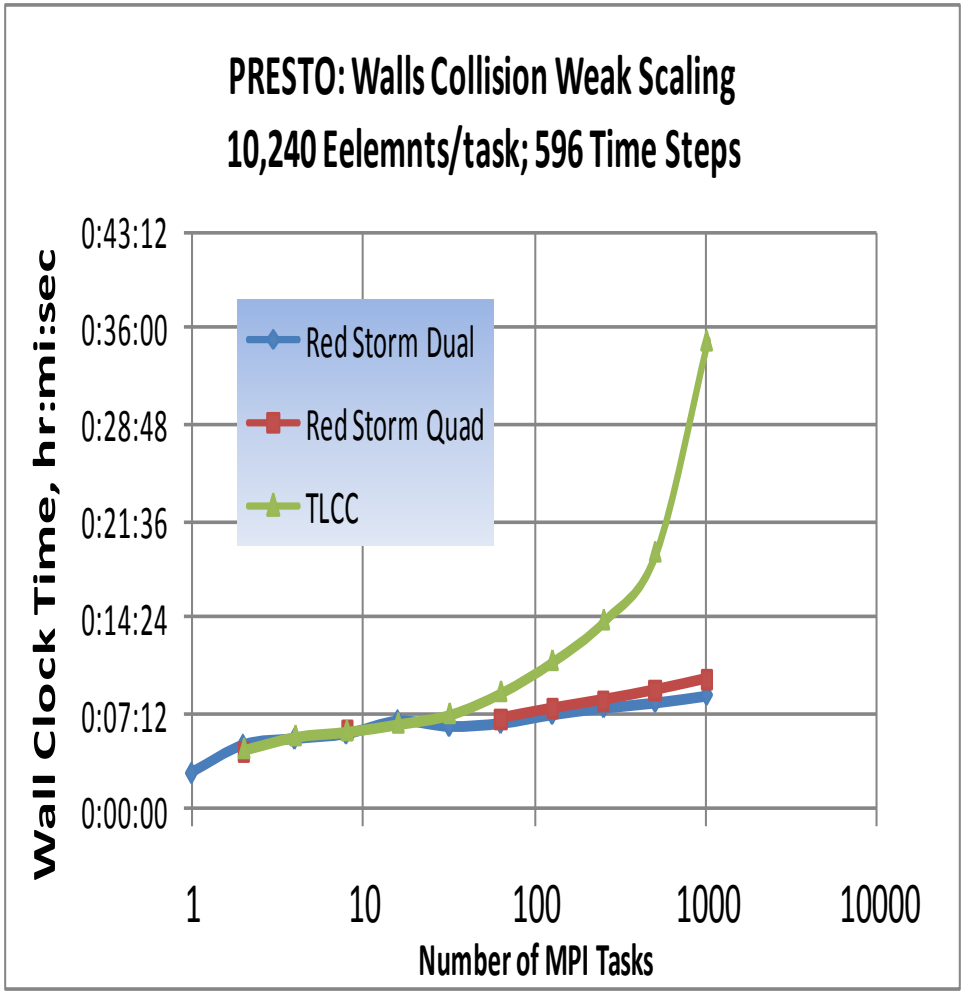
CTH – Weak Scaling

- CTH is used for two- and three-dimensional problems involving high-speed hydrodynamic flow and the dynamic deformation of solid materials
- Model: shaped-charge; cylindrical container filled with high explosive capped with a copper liner.
- Weak scaling analysis with 80x192x80 computational cells per processor.
- Processor exchanges information with up to six other processors in the domain. These messages occur several times per time step and are fairly large since a face can consist of several thousand cells
- Modest communication overhead with nearest neighbor exchanges
- At 16 cores Red Storm Quad is 1.23X faster than TLCC; at 512 cores to 1.32X; This is close to the memory speed ratio of 800/667=1.2
- CTH does not greatly stress the interconnect

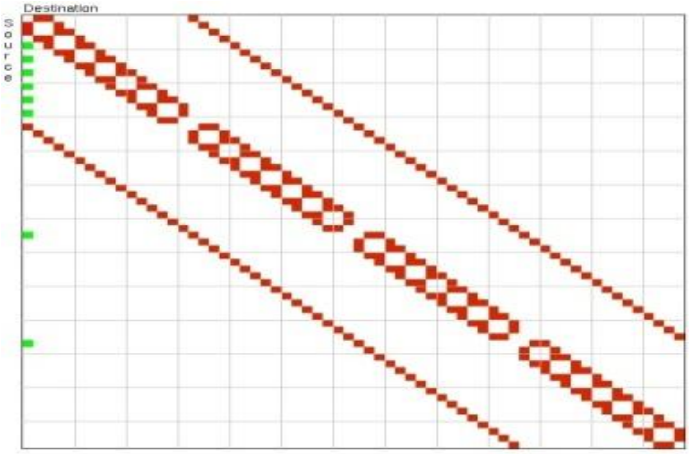


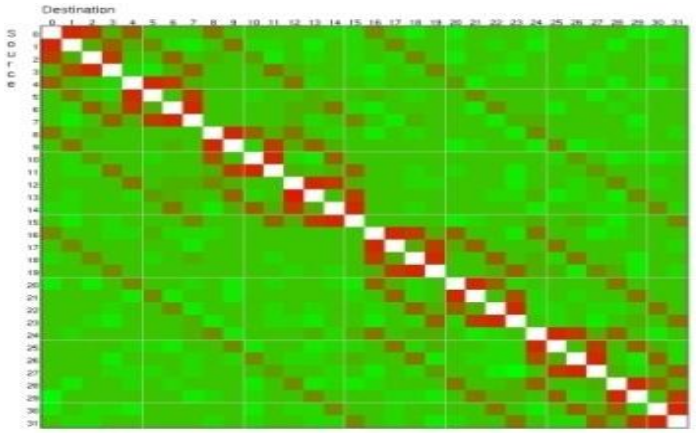
SIERRA/Presto – Weak Scaling

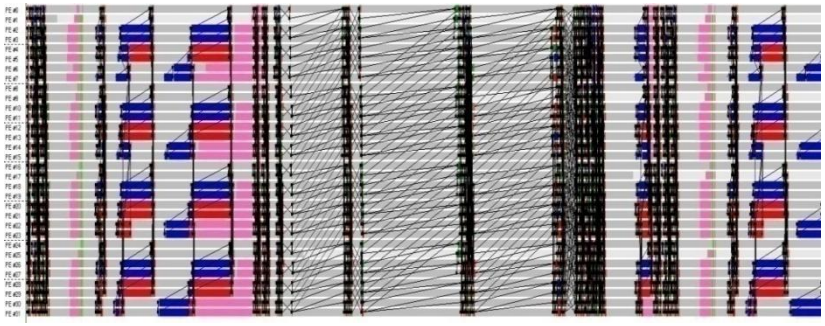
- Explicit Lagrangian mechanics with contact
- Model: Two sets of brick-walls colliding
- Weak scaling analysis with 80 bricks/PE, each discretized with 4x4x8 elements
- Contact algorithm communications dominates the run time
- The rapid increase in run time after 64 processors on TLCC can be directly related to the poor performance on TLCC for random small-to-medium size messages
- TLCC/Quad run time ratio at 1024 is 4X.

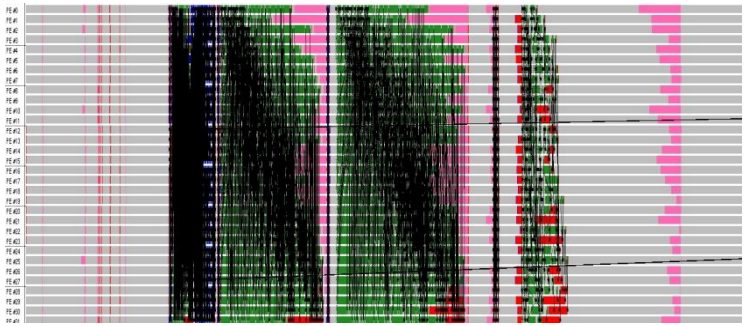


CTH, Presto(using Pronto)scaling analysis with cray_pat
Message density in Mosaic explains Red Storm, TLCC Scaling









CTH

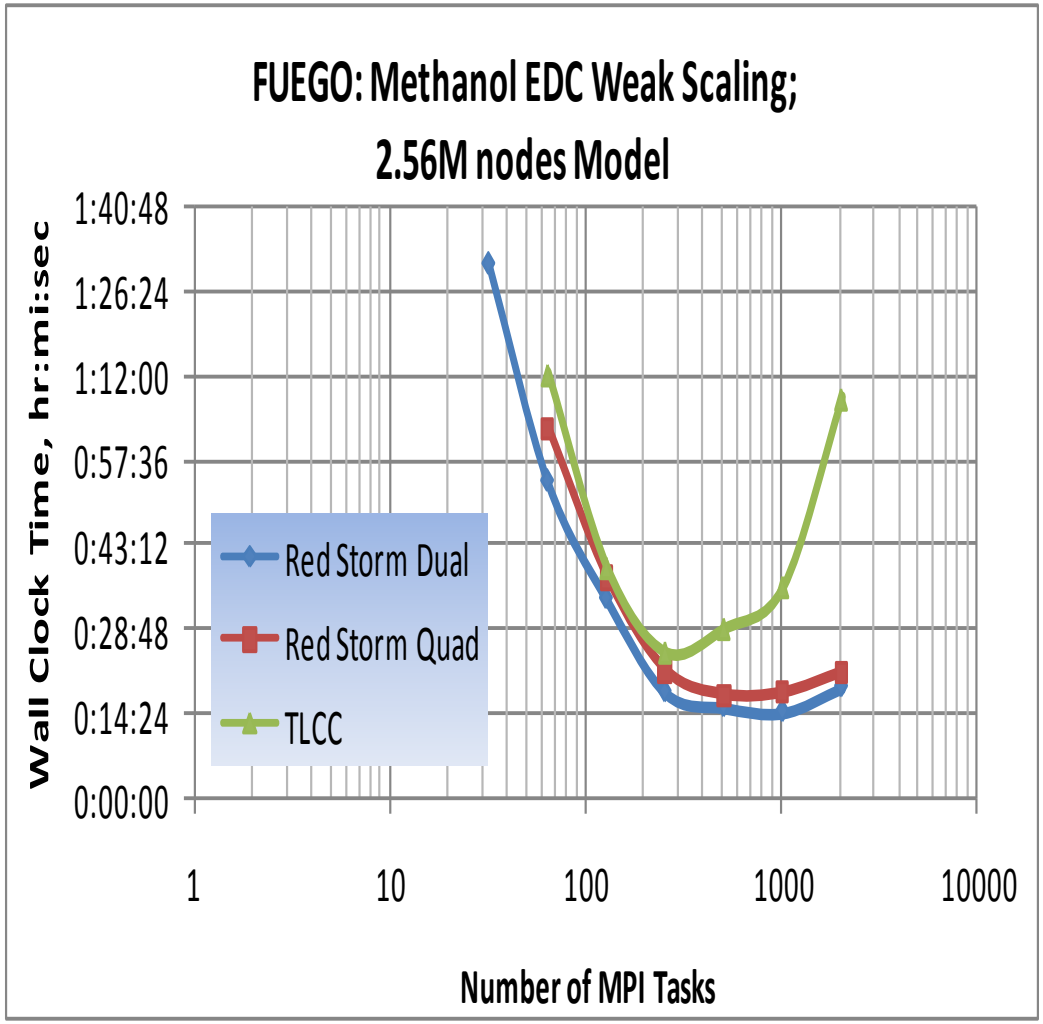
CTH Mosaic shows # of calls; regular nearest neighbor large message communications
CTH Trace: One cycle between the two MPI_bcast (Green vertical lines)

PRONTO

Pronto uses the same algorithms as Presto but was easier to profile.
Pronto Mosaic shows # of calls; Random small message communications
Pronto Trace: One cycle shows large communication to computation time ratio

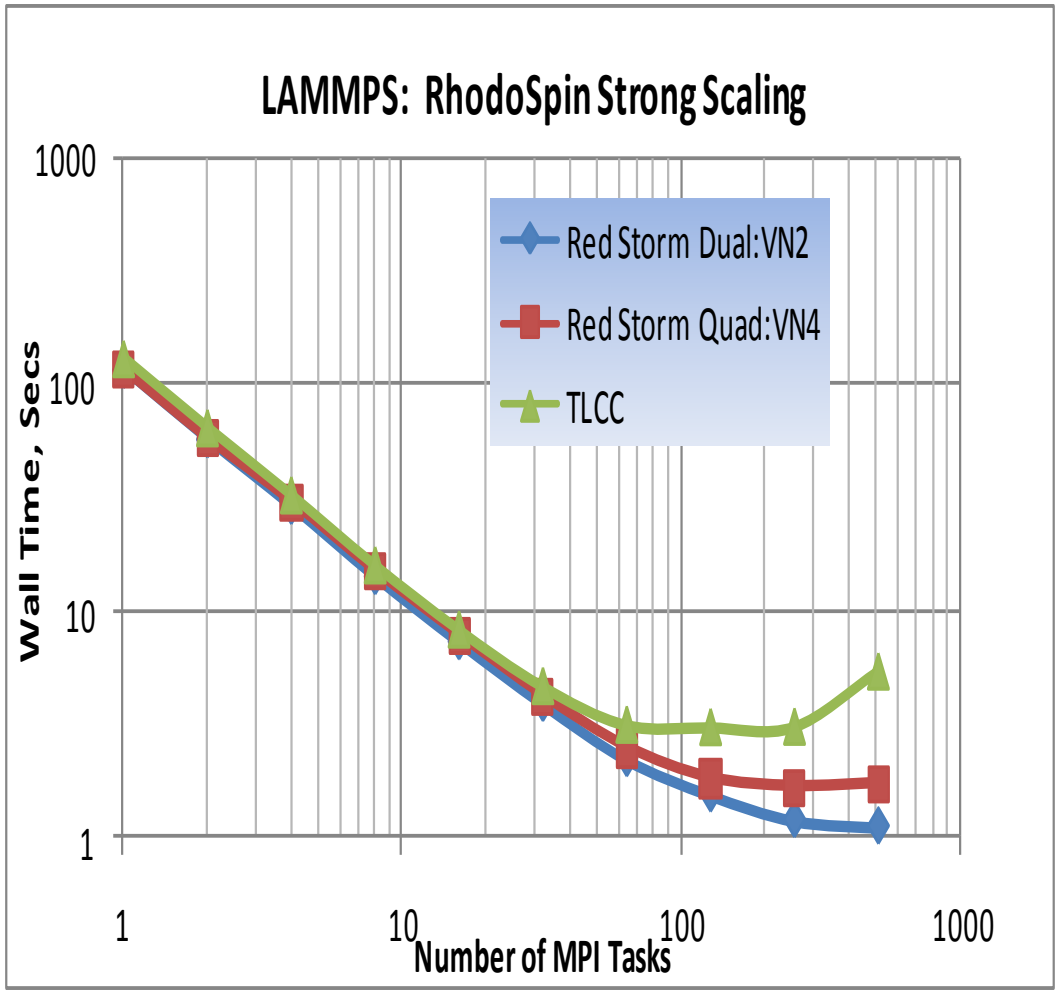
SIERRA/Fuego – Weak Scaling

- Fluids, heat transfer, participating media radiation, multi-physics code
- Model: Methanol EDC, single Fluids mesh
- Strong scaling analysis
- Scaling dominated by implicit fluid solves (ML)
- at 64, 128, 256 PEs the TLCC performance is very good
- at 512 PEs and above performance degrades; 3X slower performance on TLCC at 2048



LAMMPS – Strong Scaling

- Classical molecular dynamics
- Model: RhodoSpin benchmark
- Strong scaling analysis with 32,000 atoms for 100 time steps
- LAMMPS divides the computational domain into three dimensional sub-volumes, and makes the sub-volumes as cubic as possible, The amount of data exchanged is proportional to the surface area of the sub-volume.
- Little sensitivity to memory performance; Very good performance on TLCC till 128 PEs
- This example illustrates how even if the application does not stress the memory or the interconnect, Red Storm shows superior scalability



Conclusions

❖ The superior architecture of the Red Storm is evident from the variety of benchmarks and applications presented. The node architecture (minimizing memory contention), the Interconnect architecture (maximizing BW for unstructured, random messages), and the LWK at compute nodes (minimizing OS overheads) are all absolutely necessary to achieve scalability

❖ For 256 or fewer MPI processes similar performance was observed on TLCC and Red Storm for many applications when using all the cores on a node

❖ But for some applications like Presto, there is a need for architecture like the Red Storm as the scaling results demonstrate

❖ Although the four socket node on TLCC has cost advantages, our MPI applications show potential 2X performance penalty due to memory contention

❖ The Red Storm Quad node core has close to 2X peak FLOPS compared to the Red Storm Dual node core. But the run time are close because most applications could not take advantage of the four FLOPS/clock.

❖The inevitable cost pressures to increase core counts on a node might be detrimental in that no improvements in run time may result.