



Adapting Codes for a Heterogeneous Multi-Core **Red Storm**

**Douglas Doerfler
Courtenay Vaughan
Sandia National Laboratories
Scalable Computer Architectures Dept.**

**Los Alamos Computer Science Symposium
October 13 - 15, 2008**



Red Storm

True MPP, designed to be a single system

- Full 3-D mesh interconnect
- 12,960 compute nodes
 - 6,240 AMD quad-core Opterons @ 2.2GHz
 - 6,720 AMD dual-core Opterons @ 2.4 GHz
 - $24,960 + 13,440 = 38,400$ cores
- Plus 640 Service and I/O Nodes
- 80.6 Terabytes of memory
- 2.36 Petabytes (1.8 PB formatted) of disk storage
- 3715 embedded RAS processors

Sandia contributions include

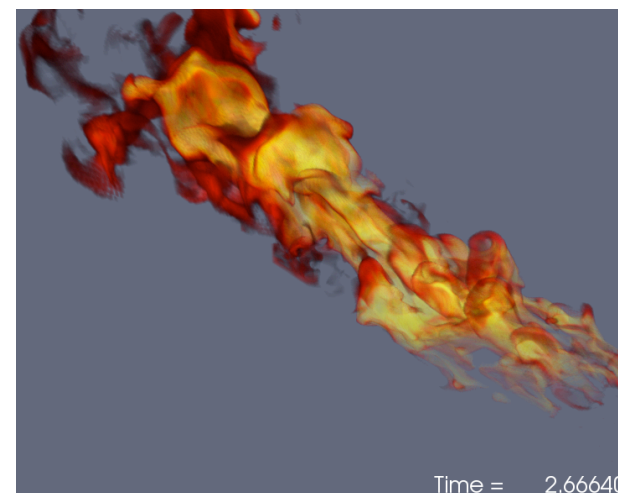
- MPP system architecture
- Helped design interconnect
- Lightweight kernel strategy
- Red/Black switching

Excellent performance

- 102.2 TF on HPL
 - Expect > 200 TF for Nov. 2008 list
- 101.4 TF on Nov. 2007 list
 - Second system ever to exceed 100TF
 - First general-purpose system to exceed 100TF

Successful technology transition

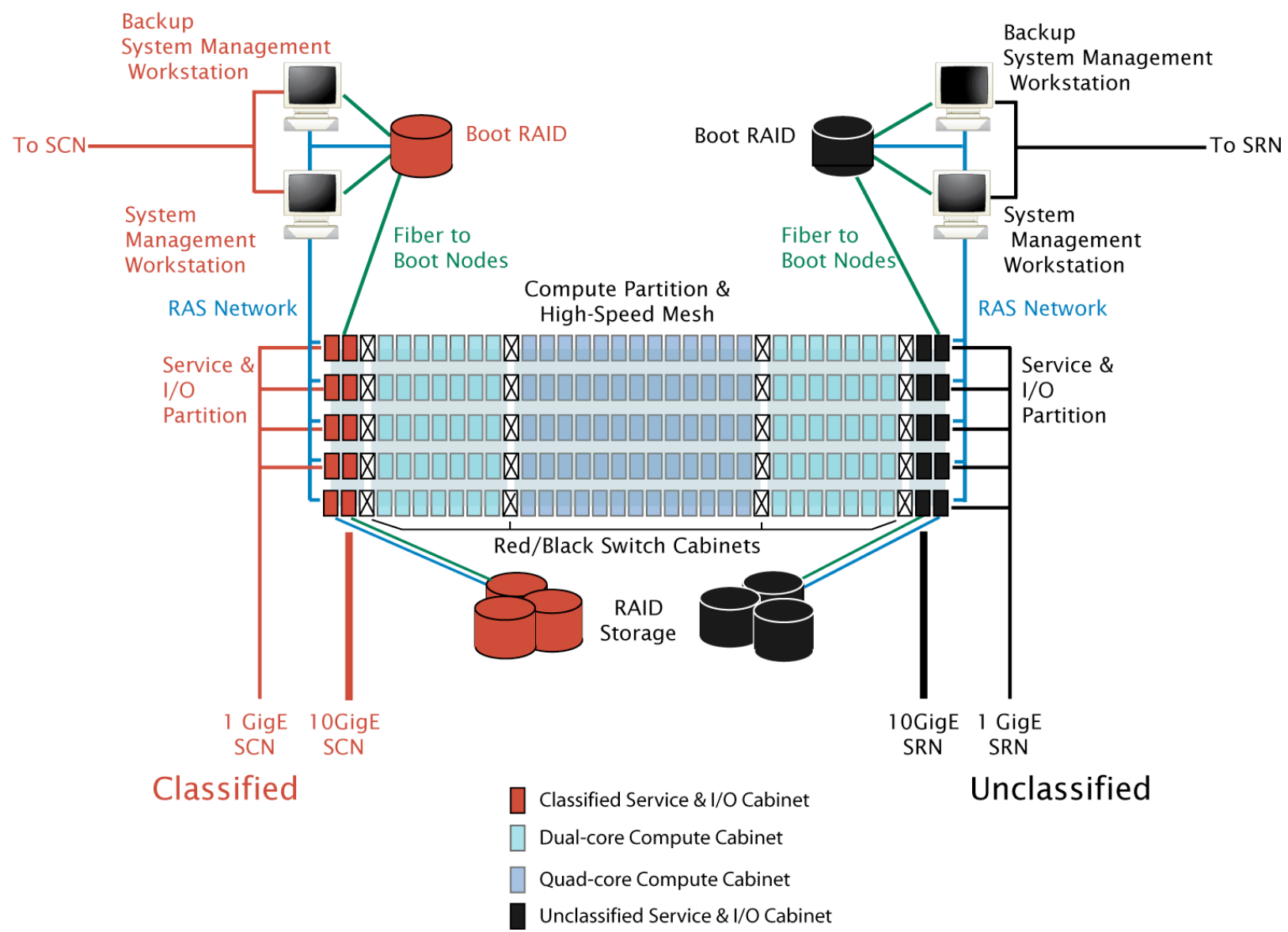
- The 1,000th Cray XT cabinet had been assembled on September 4, 2008, marking Red Storm and the Cray XT family as the most successful line of MPPs in the history of supercomputing.



Fire simulations are being used to certify new test facility and will be used to analyze weapons safety issues



Red Storm Physical Layout





Red Storm 2008 Upgrade

- Center compute section (65 cabinets) upgraded to quad-core AMD Budapest Processors and 800 MHz DDR2 memory subsystem
 - New XT4 compute blades
 - New Compute Boards
 - New 2.2 GHz AMD Budapest Quad Core Processors
 - New 800 MHz DDR2 Memory 2 GB/core, 49.9 TB total
 - New VRMs
 - Reuse of Seastar 2.1 Mezzanines, L0s, Cabinets, Cables, PDUs, etc.
- Red and Black compute sections (35 cabinets each) remain dual-core AMD Opteron with 400 MHz DDR2 memory subsystem
 - Memory Upgrade to 2 GB/core, 26.9 TB total



Red Storm Configuration Over Time

	Red Storm (05)	Red Storm (06)	Red Storm (08)
Theoretical Peak Performance	43.52 TF	130.56 TF	290.30 TF
MP-Linpack Performance	36.19 TF	102.2 TF	TBD
Total Memory	33.4 TB	39.2 TB	80.6 TB
System Memory B/W	57.99 TB/s	78.14 TB/s	126.9 TB/s
Disk Storage (User Formatted) (Red ✖ Black)	170 TB ✖ 170TB	170 TB ✖ 170 TB	1.5 PB ✖ 710 TB
Parallel File System B/W (Red ✖ Black)	100 GB/s 50 GB/s ✖ 50 GB/s	100 GB/s 50 GB/s ✖ 50GB/s	170 GB/s 100 GB/s ✖ 70 GB/s
External Network B/W (Red ✖ Black)	25 GB/s ✖ 25 GB/s	25 GB/s ✖ 25 GB/s	25 GB/s ✖ 25 GB/s

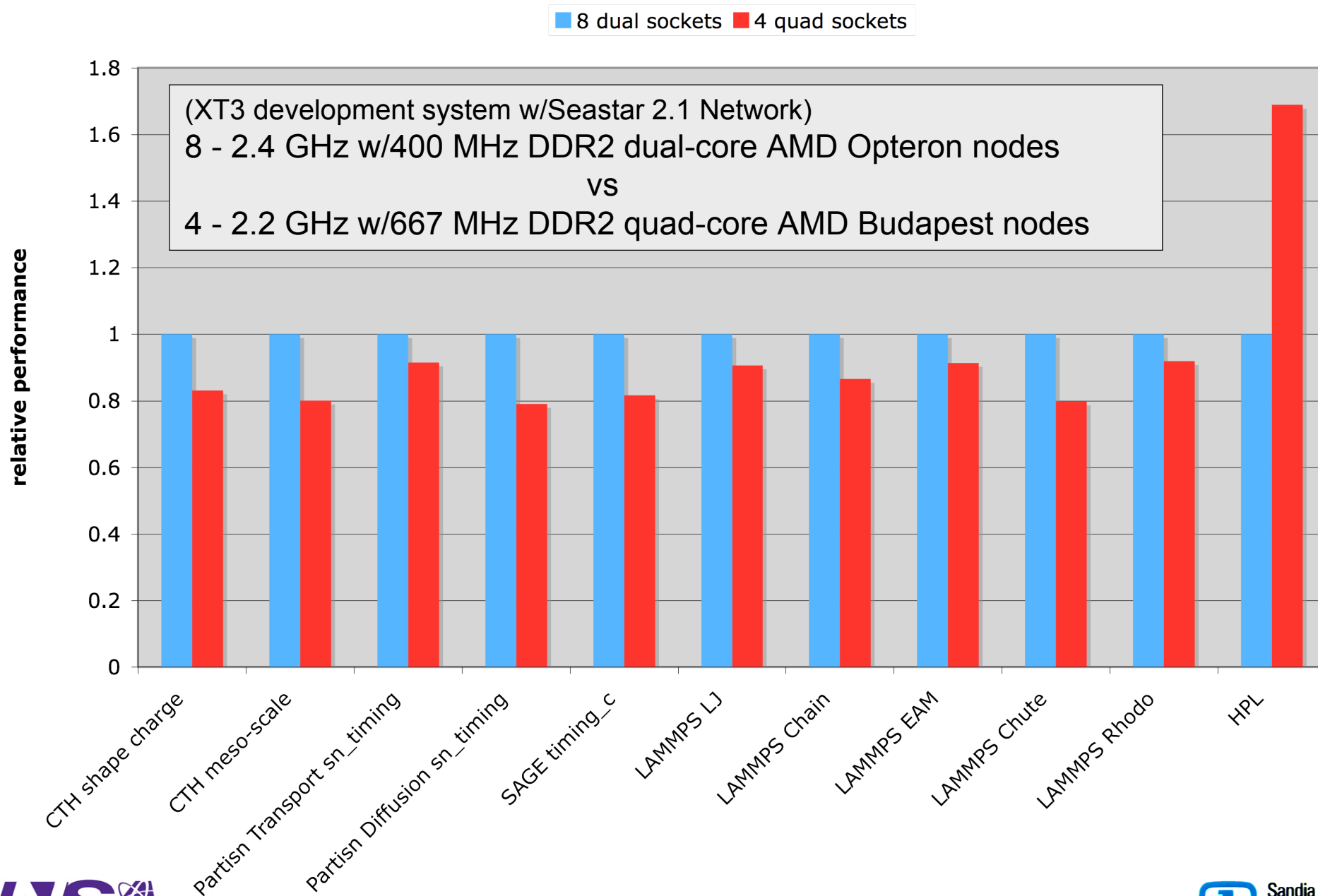


Heterogeneous **Red Storm**: Quads & Duals

Major AMD Quad-Core Opteron Features

- Increased memory subsystem performance
 - 800 MHz DDR2 as opposed to 400 MHz DDR2
 - Note that test results are 667 MHz parts
- Increased SSE length
 - 256 bit SSE as opposed to 128 bit
 - 4 FLOPs/clock as opposed to 2 FLOPS/clock
- Increased large page (2MB) TLB entries
 - 128 TLB entries as opposed to 8 entries
 - 512 small pages (4KB) entries remains the same
 - Red Storm runtime allows choice of large page mode or small page mode at job launch time
 - Larger table has “flopped” runtime preference for most applications to large page mode as opposed to small page mode
 - Many are < 3% improvements
 - HPCC Random Access - 14% to 23% better
 - HPCC PTRANS - 2.8x to 3.2x better!

Dual vs Quad Comparison: 16 Cores





HPL on Red Storm

- HPL (High Performance Linpack)
 - Performs a matrix solve by LU decomposition
 - Used by Top 500 List to rank machines
 - Some folks pay attention to this?
- What's the Issue with Heterogeneous Red Storm?
 - ⇒ HPL runs as fast as the slowest node
 - Limits potential to 184 peak TFLOPS
- How to get maximum performance?
 - ⇒ Divide the matrix to give more work to quad-core processors



HPL - Nominal Block Assignment

- HPL solves a randomly generated matrix by using a LU decomposition
- $P \times Q$ grid of processors is used
- Matrix is divided into square blocks over the grid
 - Performance varies with block size
 - Optimal block size different for different processors
 - Blocks are assigned to processors in a cyclical fashion in both rows and columns
 - Each processor can have thousands of blocks



HPL - Modified Blocking

- On single socket tests
 - a dual-core core gets ~ 4.2 GFLOPS (87.5%)
 - a quad-core core gets ~ 7.2 GFLOPS (81.8%)
 - Each has its “optimal” block size
- First approximation approach
 - ⇒ Give quad core processors 2x the work of dual core processors
 - Dual cores will be idle part of the time
 - Use quad-core block size for calculation
 - Reduces the imbalance in quad vs dual computation rate



HPL - Implementation

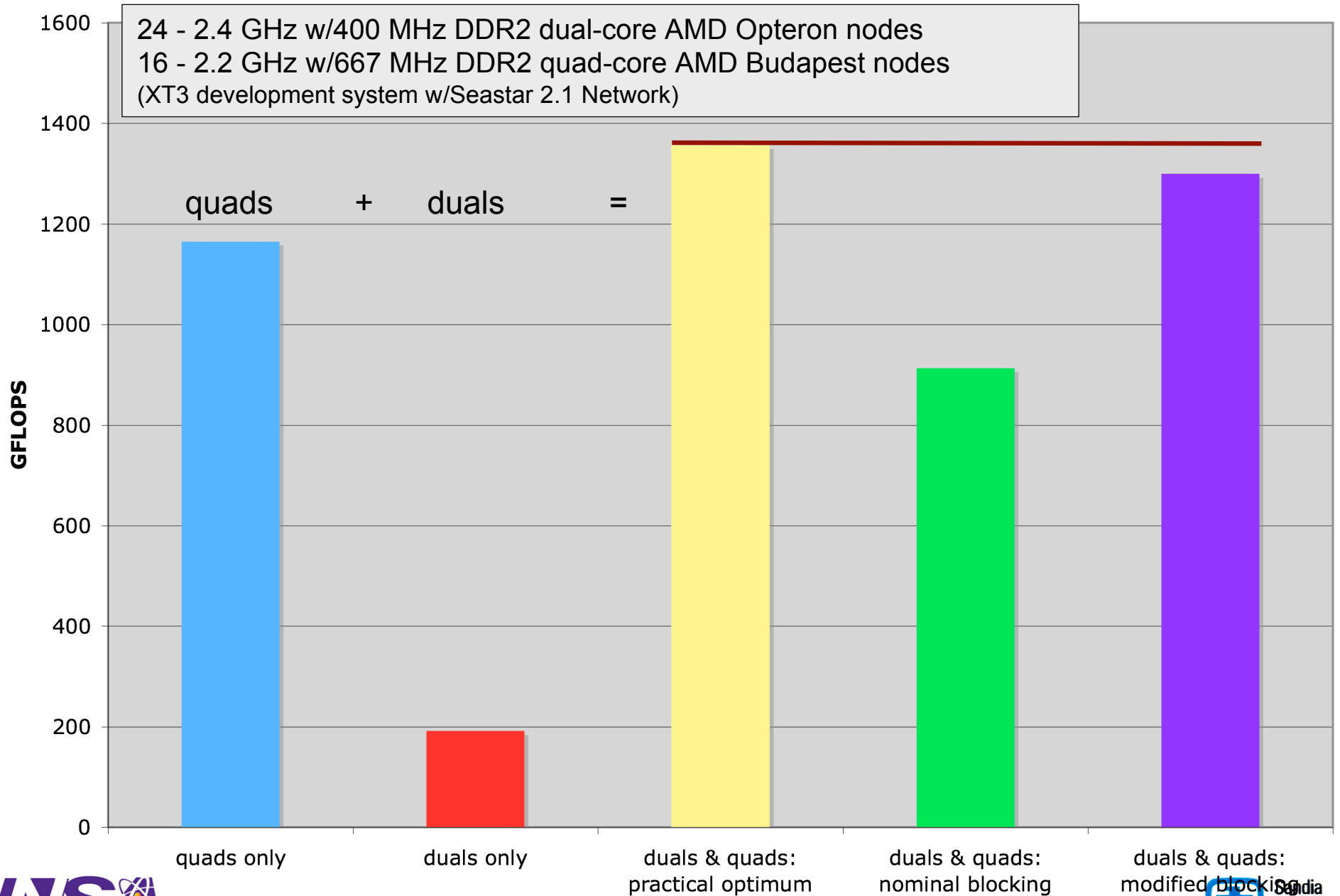
- Give two columns of blocks to each quad-core core as the blocks are distributed
- Requires that the $P \times Q$ distribution of cores be chosen so that all of the cores in a column of $P \times Q$ are either quad-core or dual-core
- Fairly easy to obtain given machine layout
 - Each row of P is partitioned by red/center/black
 - Which translates to dual/quad/dual partitioning
 - Choose P and Q such that
 - $P \times Q_1$ = number of cores per each end
 - $P \times Q_2$ = number of cores in center
 - $P \times Q = P \times (Q_1 + Q_2 + Q_1)$
- To assign 2x number of blocks to quad-core cores, create a “virtual” column dimension, Q' , and divide blocks among Q' virtual columns
 - $Q' = \# \text{ columns of dual-core cores} + 2 * \# \text{ columns of quad-core cores}$
 - $Q' = (Q_1 + Q_1) + (2 * Q_2)$



HPL - Code Changes

- Matrix generation changed to generate additional blocks
- Column index functions changed to properly index which processor owns each column of matrix
- Changes to backsolve routine to account for the additional blocks
 - Code doing different things for current column and previous column
 - Potentially on same processor

HPL: Results on Red Storm Dev System





Next Steps

- Application Analysis at Scale
- HPL at Scale
 - Watch SC08 announcements
 - Perhaps there needs to be a load balancing algorithm integrated into HPL?



Questions

dwdoerf@sandia.gov
ctvaugh@sandia.gov