SAND2007-2387C

Capability Computing on Red Storm: Performance of Large Applications

Robert E. Benner, Robert A. Ballance, Karen Haskell, John P. Noe and Joel O. Stevenson

Sandia National Laboratories

Albuquerque NM 87185 USA

In preparation for the testing and acceptance of the 10K processor Red Storm system, a suite of ten applications/benchmarks was developed to assess whether major applications would realize at least a seven-fold performance increase on the new system relative to its predecessor. This methodology has subsequently proven valuable in addressing diverse performance issues: e.g. the benefits of processor and memory upgrades, particularly the benefits of dual-core processors. We find that although most of these applications show a 15-fold performance improvement over the ASCI Red system, there are interesting outliers: PARTISN shows run time speedups of up to 65X while Salinas manages a mere 6-8X performance increase. Conversely, PARTISN sees meager performance gains in dual-core mode, whereas the other 7X applications realize 60-90% performance gains from the second core. The results validate Red Storm as a capability platform for major scientific and engineering codes on 2K-10K processors.

This work was supported in part by the U.S. Department of Energy. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States National Nuclear Security Administration and the Department of Energy under contract DE-AC04-94AL85000.

INTRODUCTION

The goal of the 7X performance testing is to assure Sandia National Laboratories, Cray, and the DOE that Red Storm achieves its performance requirements [7]. The performance tests are defined as a comparison between ASCI Red and Red Storm. In general, the Red Storm contract calls for a series of speedup comparisons using selected applications at various problem sizes. The focus is on problem sizes and processor counts representative of capability computing; i.e. single application runs that use 20% to 100% of the processors. Our approach is to identify one or more problems for each application, run those problems at two or three processor sizes in the capability computing range, and compare the results between ASCI Red and Red Storm. This approach led us to a set of 25 test problems. This paper describes the two computer systems, the ten applications in the "7X Suite", the 25 test problems, and the results of these tests. Red Storm is more than 7X faster on all of the tested applications and in some cases more than 20X faster.

The 7X performance suite consists of ten applications/benchmarks used in performance testing: Alegra, CTH, ITS, PARTISN, Presto, SAGE, Salinas, sPPM, UMT2K and Calore. A short description of each of these applications is given in this paper. Of these applications, Calore was not successfully built for ASCI Red, so it is omitted from the results to follow, and benchmark runs for Alegra and Presto have not been completed on Red Storm.

The Red Storm supercomputer, originally built in 2005 with 2.0 Ghz single core Opteron processors, received several upgrades to processors, memory and system software in 2006. Red Storm compute nodes are now dual core, 2.4 Ghz Opterons. Users have the option of running applications on both cores on each node of the system, or on only one core per node. The latter option is useful for those applications which are memory intensive and cannot abide having a node's memory partitioned between the two cores. During the course of executing 7X applications on Red Storm, results were collected in both modes. The results validate the efficacy of the dual-core upgrade, as most of these applications make efficient use of the second core.

The availability and applicability of this test suite to answer design questions and evaluate upgrade options, such as the dual-core upgrade, further validates the need for evaluation of capability-class, massively parallel systems with real applications.

ASCI Red

ASCI Red was the first computer in the Advanced Strategic Computing Initiative (ASCI) DOE program, built by Intel and installed at Sandia in late 1996. The design was based on the Intel Paragon computer. In December 1996, three quarters of ASCI Red was measured at a world record 1.06 TF on MP LINPACK and held the record for fastest supercomputer in the world for several years, maxing out at 2.38 TF after a processor and memory upgrade in 1999.

ASCI Red's distributed memory MIMD design provided high degrees of scalability for I/O, memory, compute nodes, storage capacity, and communications; standard parallel interfaces also made it possible to port parallel applications to the machine. The machine was structured into four partitions: Compute, Service, I/O, and System. ASCI Red used two operating systems, the Teraflops Operating System (distributed OSF UNIX) on the Service, I/O, and System Partition, and a Sandia-developed lightweight kernel (Cougar) on the Compute nodes. The Teraflops Operating System is Intel's distributed version of UNIX. It is a full-featured UNIX, used for boot and configuration support, system administration, user logins, user commands/services, and development tools. Cougar, a very efficient and high-performance operating system providing program loading, memory management, message-passing support, some signal handling and exit handling, and run-time support for the supported languages. This combination of operating systems makes it possible to specialize for specific tasks and standard programming tools to make the supercomputer both familiar to the user and nonintrusive for the scalable application. The machine provides a single system image to the user.

In normal operation, disconnect cabinets divided ASCI Red into two sides; unclassified and classified. In this situation, each side appeared as a separate plane in the mesh topology. The configuration was 1168 compute nodes on the unclassified end and 1166 compute nodes on the classified end. The middle section consisted entirely of 2176 computational nodes, which could be switched from the unclassified end to the classified end and back again. The total number of compute nodes was 4510.

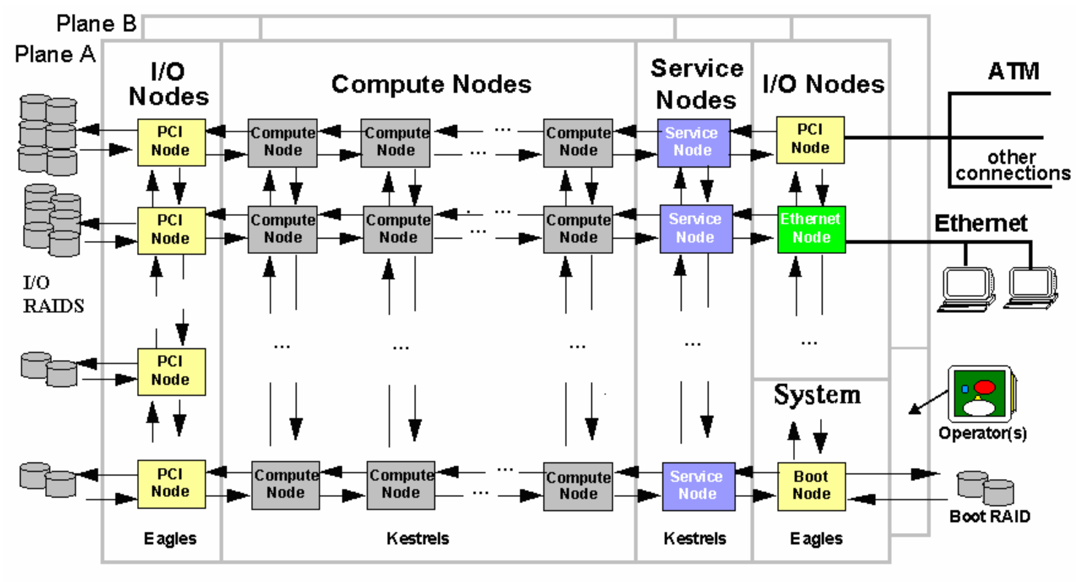

Figure 1. Diagram of the ASCI Red system.

While message passing is used between nodes, shared memory mechanisms are used to exploit parallelism on a node. Each compute node has two processors. The 7X testing was performed on ASCI Red in Proc 0 and Proc 3 modes only: (1) with the Proc 0 yod option – the default mode which ignores the second processor – the entire 256 MB of system RAM on the node is available to the application; (2) with the Proc 3 yod option – the "virtual node mode" that treats each processor as a separate compute node – the processors share memory so only half the system RAM is available to the application (128 MB) on each processor.

ASCI Red was decommissioned in late 2005, shortly after completing the 7X runs reported below.

RED STORM

Red Storm, the follow-on computer to ASCI Red, was developed jointly by Cray and Sandia [7]. It was manufactured by Cray and installed at Sandia in early 2005 as an XT3 system. In 2005, Red Storm was measured at 36 TF on MP LINPACK. Following an upgrade from 2.0 Ghz single-core Opteron processors to 2.4 Ghz dual-core Opterons in 2006, Red Storm was measured at 101.4 TF on MP LINPACK.

The distributed memory MIMD architecture of Red Storm follows the model of ASCI Red: Users interface to the system via a Linux operating system, and the compute nodes run a lightweight kernel.

Red Storm combines commodity and open source components with custom-designed components to create a system that can operate efficiently at immense scale. Cray XT3 systems scale from 200-30,000 processors – the ability to scale to such proportions stems from the design of system components.

The basic scalable component is the node. There are two types of nodes. Compute nodes run user applications. Service nodes provide support functions, such as managing the user's environment, handling I/O, and booting the system. Each compute node and service node is a logical grouping of a processor, memory, and a data routing resource. Cray XT3 systems use a simple memory model – for applications distributed across numerous nodes, each instance of the application has its own processor.

The system interconnection network is the data-routing resource that Cray XT3 systems use to maintain high communication rates as the number of nodes increases. The system interconnection network enables the system to achieve an appropriate balance between processor speed and interconnection bandwidth.

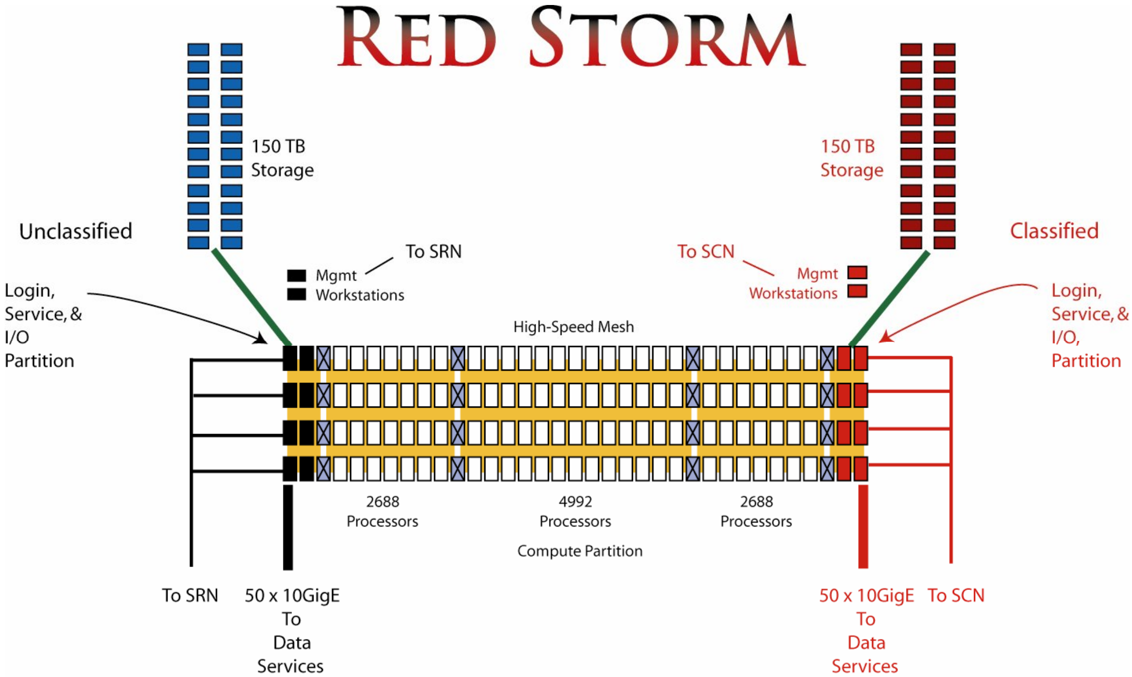

Figure 2. Cabinet-level diagram of the Red Storm system.

The Red Storm architecture facilitates simultaneous usage on the unclassified (black) and classified (red) sides of the machine. In normal operation, disconnect cabinets divide Red Storm into two sides; unclassified and classified. The initial configuration was 2688 compute nodes on the unclassified end and 2688 compute nodes on the classified end. The middle section consisted entirely of 4992 computational nodes, which could be switched from the unclassified end to the classified end and back again. The total number of compute nodes was 10368.

A 5th row of cabinets was added in an August-October 2006 upgrade, bringing node count to 3360 on the unclassified side and 3360 on the classified side. The middle section contains 6240 nodes. Total number of compute nodes is now 12960. Each compute node was upgraded to dual core topology, bringing total processor count to 25920. Processor speed was upgraded from 2.0 to 2.4 Ghz.

The 7X tests are performed on Red Storm in either SN or VN mode: (1) the SN option, which is the default, ignores the second processor and makes the entire system

Table 1. System parameters for ASCI Red and post-upgrade Red Storm

	ASCI Red	Red Storm (initial op)	Red Storm (post-upgrade)
Compute Nodes (Red/RedBlack/Black)	4510 (1166/2176/1168)	10368 (2688/4992/2688) Compute partition	12960 (3360/6240/3360) Compute partition
Compute Processors (Red/RedBlack/Black)	9020 (2332/4352/2336)	10368 (2688/4992/2688) Compute partition	25920 (6720/12480/6720) Compute partition
Service Nodes (Red/Black)	52 (26 / 26)	512 (256/256) Service and I/O	512 (256/256) Service and I/O
Disk I/O Nodes (R/B)	73 (37 / 36)	See above.	See above.
System Nodes (R/B)	2 (1 / 1)	RAS & System Mgmt. partition	RAS & System Mgmt. partition
Network Nodes (R/B)	12 (6 / 6) Ethernet ATM	100 (50/50) 10GigE to RoSE 20 (10/10) 1GigE to login nodes	100 (50/50) 10GigE to RoSE 20 (10/10) 1GigE to login nodes
System Footprint	~2500 Square Feet	~3100 Square Feet (1.7 MW power requirement)	~3800 Square Feet (2.5 MW power requirement)
Number of Cabinets (Compute / Switch / Disk)	104 (76 / 8 / 20)	140 (108 /16 /16 service & I/O)	175 (135 /20 /20 service & I/O)
System RAM	1212 GB Total	33.38 TB	39.19 TB (will support 79.7 TB)
Topology	Mesh (38 X 32 X 2)	Mesh (27 X 16 X 24)	Mesh (27 X 20 X 24)
Node Link (Node to Node) Bandwidth - Bi-directional	800 MB/s	6.0 GB/s Bi-Directional Link B/W	6.5 GB/s Bi-Directional Link B/W (after encoding overhead removal)
Cross Section Bandwidth - Bi-directional	51.2 GB/s	2.3 TB/s Minimum Bi-Section B/W	2.5 TB/s Minimum Bi-Section B/W
Total Number of Processors	9298 Pentium II Xeon (333 Mhz)	10368 Opteron compute (2.0 Ghz) + 512 Service and I/O	25920 Opteron compute (2.4 Ghz) + 512 Service and I/O
Processor to Memory BW	533 MB/s	4 GB/s	4.8 GB/s
Compute Node Peak Perf.	666 MF	4 GF	4.8 GF
System Peak Performance	3.1 TF	41.47 TF	124.4 TF
Linpack Performance - Full System	2.38 TF	36.19 TF	101.4 TF
RAID Disk Storage - Total / per Color	12.5 TB / 6.25 TB	~300TB/150TB	Unchanged
RAID I/O Bandwidth – Total / per Subsystem	4.0 GB/s / 1.0 GB/s	50 GB/s sustained transfer rate	Unchanged

RAM on the node available to the application; (2) the VN option, which treats each processor as a separate compute node and makes only half the system RAM available to the application.

The software environment of Red Storm is summarized as follows. Operating systems include Linux on service and I/O nodes (SuSE Enterprise Server 9.0), Catamount VN lightweight kernel on compute nodes, and Linux on RAS monitors. The run-time system includes a logarithmic job launch utility (yod), the node allocator (CPA), and the batch system (PBS Pro/MOAB). The high performance file system is Lustre. The user environment includes PGI compilers (Fortran, C, C++), various libraries (MPI, I/O, Math, MPI-2), the showmesh utility for displaying node states and job layouts on the mesh, the Totalview debugger, and a performance monitor.

The lightweight compute node OS is fundamental to the Sandia architecture. It is essential for: (1) maximizing CPU resources, by reducing OS and runtime system overhead; (2) maximizing memory resources, with a small memory footprint and large page support; (3) maximizing network resources, with no virtual memory and physically contiguous address mapping; (4) increasing reliability, with a small code base and reduced complexity; (5) deterministic performance, with a high degree of repeatability; (6) scalability, for which OS resources must be independent of job size.

Application and Problem Selection Criteria

One of the mandates of 7X testing is that applications and their problem sets shall be "real". Each of the chosen applications is either a significant DOE production application or an idealized benchmark application that is based upon and closely resembles the behavior of a major DOE production application. The 7X testing effort represents production job behavior with actual input files and algorithms. Identical calculations are run on ASCI Red and Red Storm. The primary metric is wall-clock time as measured by the elapsed time to execute the entire job script, including any pre- and postprocessing. Calculations on ASCI Red and Red Storm should give equivalent answers.

Problems should be chosen to use as many ASCI Red resources (processor, memory) as possible in order to place reasonable stress on Red Storm. Problem sizes are deliberately chosen so that jobs run on ASCI Red should range from ~4-8 hours of wall clock time. Simplified geometries are preferred in order to simplify input file creation and to avoid meshing problems during benchmarking. All applications should use standard production-use capabilities including I/O, checkpoint/restart, and visualization files. When an application can be run using alternative algorithms, such as Alegra with and without contact, said application may have more than one benchmark problem in the suite.

We will test applications in three modes: standard, stretch, maximum.

Standard mode - The standard size problem should easily run and be accurately measured on both platforms. This standard is used to calibrate the testing and to check for shifts in performance due to changes in the underlying system software. Standard refers to “Large – proc 0 mode” on ASCI Red and “Small” on Red Storm, as in Figure 3.

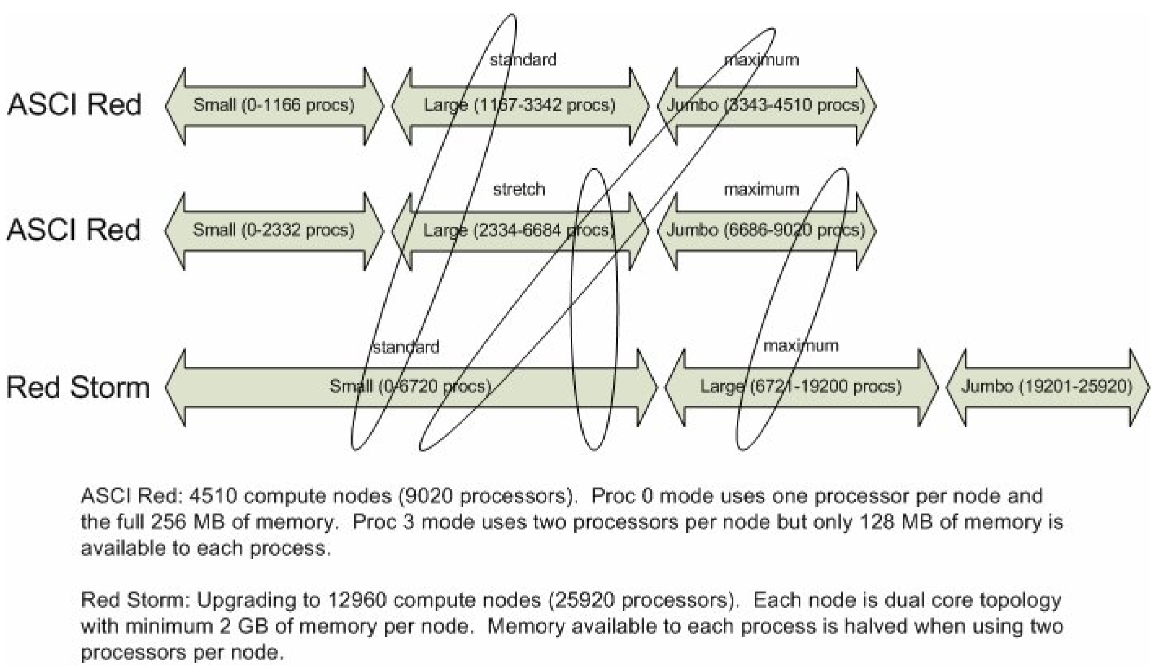

Figure 3. Standard, Stretch and Maximum modes on ASCI Red and Red Storm.

Stretch Mode - The stretch size problem will fully occupy the large configuration of ASCI Red. Stretch refers to "Large – proc 3 mode on ASCI Red and "Small" on Red Storm. Problem sets need to accommodate the reduced memory available in ASCI Red stretch mode; therefore, not all of the selected problems and applications can be run in stretch mode.

Maximum Mode - Selected applications may also be run in maximum size that requires an operational configuration of ASCI Red's entire compute node partition. Maximum refers to "Jumbo – proc 0 mode" and "Jumbo – proc 3 mode" on ASCI Red and "Large" on Red Storm.

APPLICATION AND BENCHMARK DESCRIPTIONS

Figure 4 summarizes the 7X applications and problem sizes that were studied here.

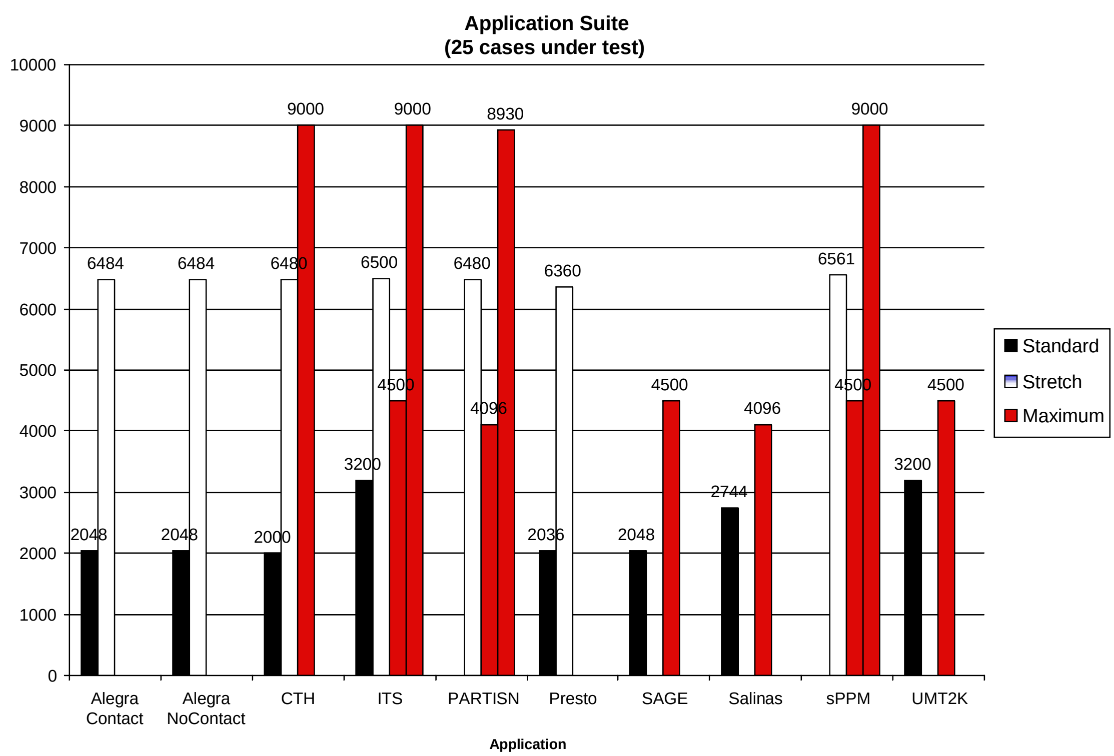

Figure 4. 7X Application Suite and Problem Sizes.

Ten applications comprise the 7X test suite: Alegra, CTH, ITS, PARTSN, Presto, SAGE, Salinas, sPPM, UMT2K and Calore. We now briefly summarize each of these applications and test problems [3]:

(1) Alegra is used to simulate the dynamic material response of complex configurations [12]. It solves coupled physics problems in 2D or 3D using Lagrangian, Eulerian, and/or ALE coordinates. The code runs efficiently on massively parallel computers and contains a large variety of physics options including hydrodynamics, magnetohydrodynamics with external circuit coupling, radiation transport, thermal conduction, and dual ion and electron temperatures. Two test problems were chosen for Alegra – one with a dynamic contact region and one without a contact region. For Alegra with contact a quasistatic electromechanics (QSEM) problem was chosen in which a curved impactor depoles a potted active ceramic element. For Alegra without contact a QSEM problem identical to the contact problem was chosen, except the boundary condition is a prescribed displacement rather than an impactor, eliminating the need for contact. In both cases problem sizes are Standard (2048 processors) and Stretch (6484).

(2) CTH is a multimaterial, large-deformation, strong shock wave, solid mechanics code [8]. CTH has models for multiphase, elastic viscoplastic, porous and explosive materials. In CTH 3D rectangular meshes; 2D rectangular and cylindrical meshes; and 1D rectilinear, cylindrical and spherical meshes are available. CTH uses second order accurate numerical methods to reduce dispersion and dissipation and to produce accurate, efficient results. CTH is used for studying armor/antiarmor interactions, warhead design, high explosive initiation physics, and weapons safety issues. The test problem is the shock physics in 3D of a large conical shaped charge. Problem sizes are Standard (2000 processors), Stretch (6480) and Maximum (9000).

(3) ITS is the Integrated Tiger Series code, a radiation transport Monte Carlo code [9]. The test problem is Monte Carlo solution of linear time-independent coupled electron/photon radiation transport problems, with or without the presence of macroscopic electric and magnetic fields of arbitrary spatial dependence. The problem

sizes are Standard (3200 processors), Maximum (4500 and 9000 processors in P0 and P3 modes, respectively, on ASCI Red), and Stretch (6500).

(4) PARTISN is the Parallel TIme-dependent SN code package [11], which is designed to solve the time-independent or dependent multigroup discrete ordinates form of the Boltzmann transport equation in several different geometries. PARTISN provides neutron transport solutions on orthogonal meshes with adaptive mesh refinement in 1D, 2D or 3D. Much effort has been devoted to making PARTISN efficient on massively parallel computers. The package can be coupled to nonlinear multiphysics codes that run for weeks on thousands of processors to finish one simulation. The test problem is "Sntiming", in which flux and eigenvalue convergence are monitored by PARTISN.
The problem sizes are Maximum (4096 and 8930 processors in P0 and P3 modes, respectively, on ASCI Red) and Stretch (6480).

(5) Presto is a Lagrangian, three-dimensional explicit, transient dynamics code [4] for the analysis of solids subjected to large, suddenly applied loads. Presto is designed for problems with large deformations, nonlinear material behavior, and contact. There is a versatile element library incorporating both continuum and structural elements.

The contact algorithm is supplied by ACME. The contact algorithm detects contacts that occur between elements in the deforming mesh and prevents those elements from interpenetrating each other. This is done on a decomposition of just the surface elements of the mesh. The contact algorithm is communication intensive and can change as the problem progresses.

The brick walls problem consists of two sets of two brick walls colliding with each other. Each brick is located on one processor so the only communication for the finite element portion of the code is for the determination of the length of the next time step. As the problem grows with the number of processors, the contact problem also grows.

Rectangular bricks are stacked in an alternating fashion in a plane to produce a wall three elements thick. Four of these walls are lined up in the thin direction. The walls are then given a sudden pressure loading such that they compress against each other.

Problem sizes are Standard (2036 processors) and Stretch (6360).

(6) SAGE is SAIC's Adaptive Grid Eulerian hydrocode [6], a multidimensional, multimaterial hydrodynamics code with adaptive mesh refinement that uses second-order accurate numerical methods. SAGE represents a large class of production computing applications at Los Alamos National Laboratory. It is a large-scale parallel code written in Fortran 90 and uses MPI for interprocessor communications. It routinely runs on thousands of processors for months at a time on capability computing systems in the DOE complex. The test problem is an asteroids simulation of 45 degree granite 3D asteroid impact into a stratified medium of water, calcite, granite crust, and mantle. Problem sizes are Standard (2048 processors) and Maximum (4500).

(7) Salinas is a highly scalable finite element structural analysis program which is tailored to run extremely large models in hostile environments. The test problem is a transient dynamics problem based on one unit cube model. Problem sizes are Standard (2744 processors) and Maximum (4096).

(8) sPPM is a Structured mesh, Piecewise Parabolic Method CFD code [1, 5]. The code is written in f90, with mixed-mode interprocessor communications: MPI and OpenMP. The test problem involves shock physics: solution of a 3D gas dynamics problem on a uniform Cartesian mesh. Problem sizes are Maximum (4500 processors), Stretch (6561) and Maximum (9000).

(9) UMT2000 is a benchmark code [2, 5] for unstructured mesh photon transport that represents the requirements of a computationally intensive, programmatically important class of application codes at Lawrence Livermore National Laboratory. Most of the code is F90 but the core kernel is C, with 90+% of cpu time being spent in the latter. The test problem is 3D, deterministic, multigroup, photon transport code for unstructured meshes. Problem sizes are Standard (3200 processors) and Maximum (4500).

(10) Calore performs large-scale nonlinear thermal analysis and is used, e.g., to simulate the thermal response of a weapon in a fire. Calore problems have been run as

part of the 7X test suite on Red Storm, but they were never run on ASCI Red, due to unresolved C++ compiler bugs on that.

RESULTS

An effort was made to set up the test problems so that each would require about 8 hrs. to execute on ASCI Red and, therefore, about 1 hr. on Red Storm. This goal was largely met, as seen in Figure 5, although it was difficult to scale the PARTISN test problem to that level. Conversely, the Salinas test problems ran for slightly more than an hour on Red Storm, as the Salinas speedups on Red Storm were only a factor of 6 to 8, lower than for all of the other applications.

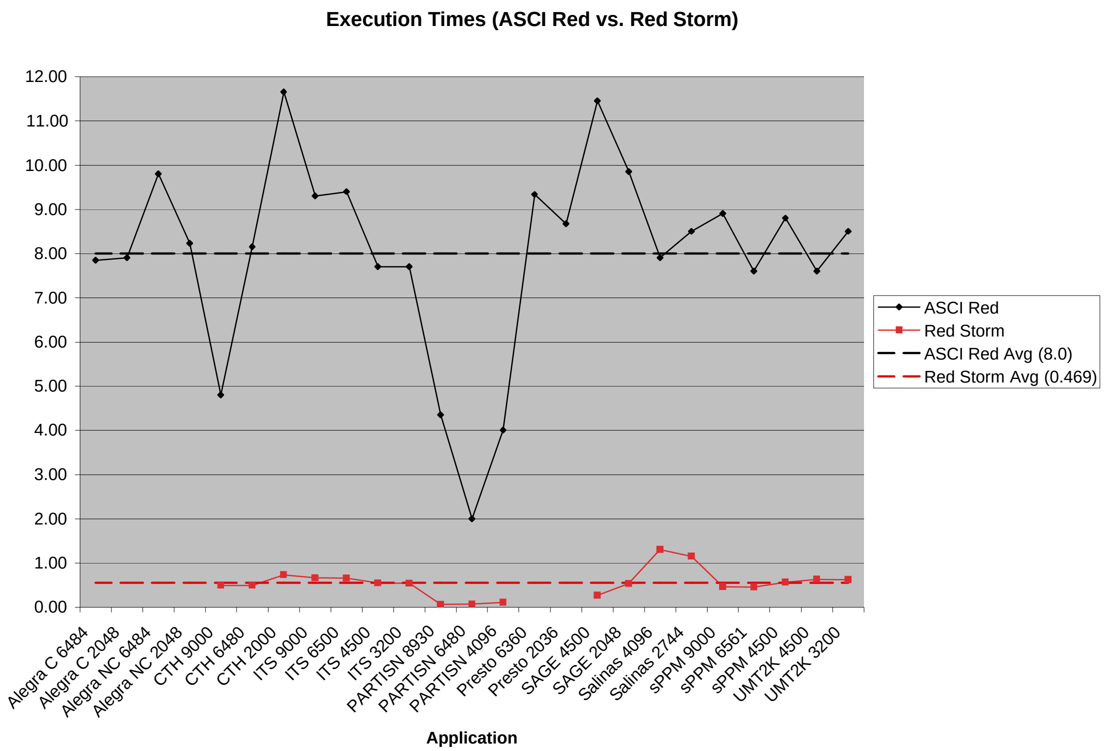

Figure 5. Benchmark execution times on ASCI Red and Red Storm.

Figure 6 shows the speedups achieved on Red Storm relative to ASCI Red. Excluding Alegra and Presto, for which the Red Storm results are not yet available, an average speedup of nearly 20 is observed across the test suite, far above the hoped-for seven-fold improvement. Two additional caveats are in order: (1) the average speedup is unduly influenced by the extremely large speedup measured for PARTISN (and, to a lesser extent, by SAGE) on Red Storm. We can speculate that ASCI Red may have been in a degraded state when the PARTISN runs were made, but this cannot be proven since the system is no longer available; (2) processor speeds were upgraded by 20% on Red Storm before these results were gathered, so the real target should now be 8.4X, not 7X.

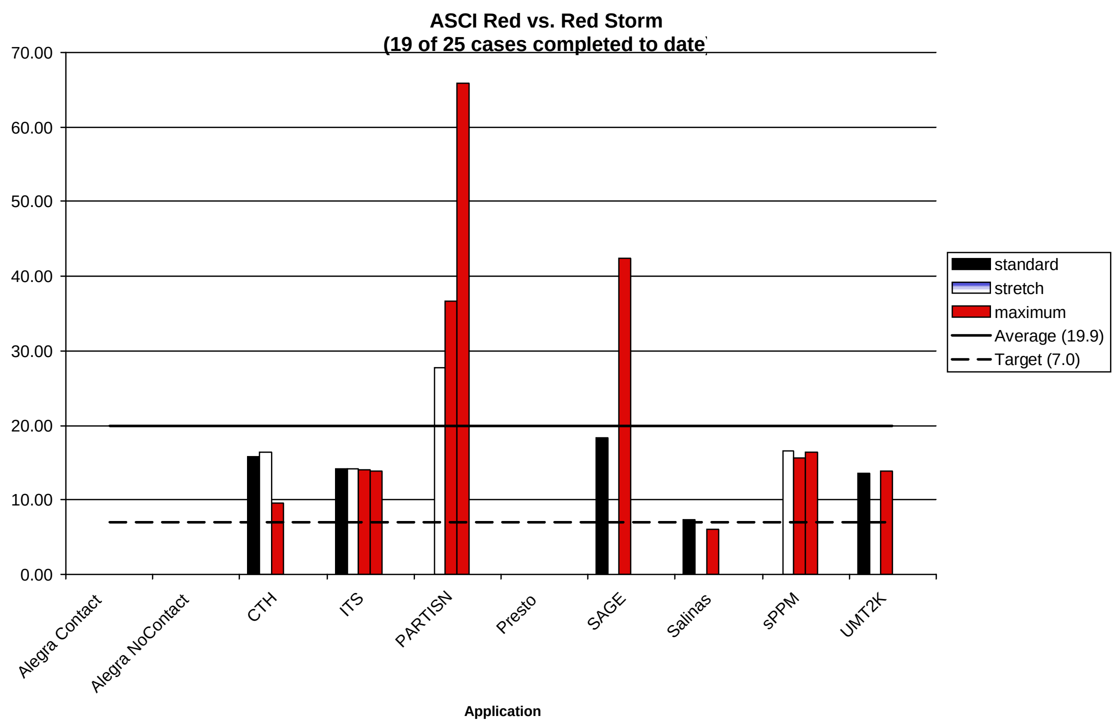

Figure 6. Benchmark Problem Speedup on Red Storm vs. ASCI Red.

In summary, an average performance gain of 15X would seem to reflect the performance of the 7X application suite on Red Storm.

EARLY VIRTUAL NODE RESULTS ON RED STORM

The recent upgrade of Red Storm to dual core sockets has provided the option of specifying either one or two cores (processors) per socket when launching an application. As noted above, the 7X tests can be performed on Red Storm in either SN or VN mode: (1) the SN option, which is the default, ignores the second processor and makes all user memory on the node available to the application; (2) the VN option, which treats each processor as a separate compute node and makes only half the user memory available to the application. If applications can run efficiently in VN mode on Red Storm, this frees up sockets for other applications.

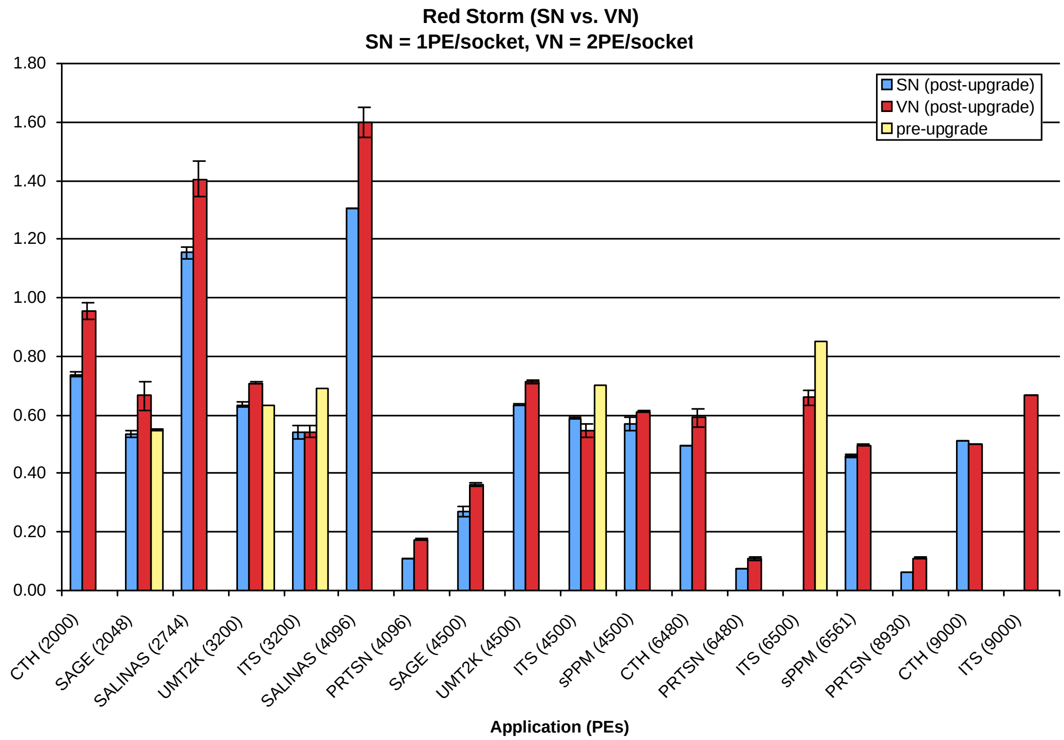

Figure 7. Comparison of Pre-upgrade Single-Core Run times on Red Storm to Post-upgrade Single-Core and Dual-Core Results.

In Figure 7 we compare the Red Storm results in terms of execution time for SN and VN runs of the test problems, as well as a few pre-upgrade SN runs. Most of the applications are demonstrating a small-to-modest performance hit (10-30%) for using the second core in VN mode. PARTISN is again an outlier with the largest dual-core performance penalty in the test suite. Interestingly, the 9000-processor CTH test shows a slight performance acceleration in VN mode relative to the same number of processors in SN mode.

Pre-upgrade runs were available for three applications in the 7X suite. ITS shows a speedup commensurate with the 20% increase in processor speed due to the upgrade, but little benefit is seen for UMT2K and SAGE for upgrading to the 2.4 Ghz processors.

SUMMARY AND DISCUSSION

The 7X test suite consists of 10 applications and benchmarks that have been used in ASCI Red vs. Red Storm performance testing. One or more problems were identified for each application, and those problems were run at two or three processor sizes, comparing the results between ASCI Red and Red Storm – 19 of 25 data points have been completed to date.

Red Storm has achieved its requirement of 7X performance over ASCI Red, posting an average speed-up of 19.9X. Furthermore, dual-core processor chips (VN mode) performed well on the 7X applications, often completing in nominally the same time as single-core (SN mode) runs - the largest efficiency drop was ~30%.

The 7X applications are now routinely used to benchmark and evaluate other new systems, e.g. highly parallel cluster systems that are acquired to serve as capacity computing systems.

However, there are some serious limitations to this methodology. Several of the applications discussed here require major porting efforts whenever a new system is to be tested. This is particularly true of the Sierra framework-based applications, such as Calore and Presto, as well as other large, modern, object-oriented applications such as Alegra. Some applications can require a week or two to be built for a new system, even if no portability issues are encountered. We see a real need for compact applications [10] based on such applications, and are actively engaged in research and development efforts to create new compact applications, so that testing and evaluation of new systems and potential procurements can be done in a timely manner.

ACKNOWLEDGEMENTS

The authors thank Courtenay Vaughan, John Van Dyke, Sue Goudy, Mahesh Rajan, and Hal Meyer for their assistance with compiling, configuring, and troubleshooting on ASCI Red and Red Storm. Many thanks also to the ASCI Red (Frank Jaramillo, Paul Sanchez, Mike Martinez, Sean Taylor) and Red Storm system administrators and support staff for their assistance. Thanks also to Mark Hamilton (1543) for assistance in setting up the Sourceforge repository.

The authors thank Cray Research engineers Paul Burkhardt, Doug Enright, Ron Pfaff, and Mike Davis for their assistance with compiling and optimizing the application codes for Red Storm benchmark runs.

Sue Goudy, Sue Kelly, Mike McGlaun, Jim Tomkins, and Courtenay Vaughan have all provided help, suggestions, and guidance as the predecessor to this paper, The 7X Cookbook, was assembled.

We also thank the application code developers for their assistance: Brian Franke (ITS), Garth Reese (Salinas), Riley Wilson (Salinas), Galen Gizler and John Daly (SAGE), Kevin Brown and Arne Gullerud (Presto), Bruce Bainbridge and Tolulope Okusanya (Calore), and Allen Robinson, Rich Drake and Josh Robbins (Alegra).

REFERENCES

1. J. Owens, The ASCI sPPM Benchmark Code, Lawrence Livermore National Laboratory, Livermore, CA, http://www.llnl.gov/asci_benchmarks/asci/limited/ppm, 1996.
2. B. Chan, The UMT Benchmark Code, Lawrence Livermore National Laboratory, Livermore, CA, http://www.llnl.gov/asci/purple/benchmarks/limited/umt, 2002.
3. R. A. Ballance, The 7X Cookbook, January 10, 2005, Version 1.5.1.
4. J. Richard Koteras and Arne S. Gullerud, Presto User's Guide Version 1.05, Sandia National Laboratories, Report SAND2003-1089, April 2003.

5. T. Spelce, Early Performance Results from the LLNL/NNSA Purple Computer, UCRL-PRES-22309, SCICOMP 12, July 17-21, 3006, Boulder, CO.

6. D. J. Kerbyson, et al., *Predictive Performance and Scalability Modeling of a Large Scale Application*, in Proceedings of the ACM-IEEE International Conference on HPC and Networking (SC01), Nov. 2001.

7. R. Brightwell, et al., *Architectural Specification for Massively Parallel Computers: An Experience and Measurement-based Approach*, Concurrency and Computation: Practice and Experience, Special Issue: The High Performance Architectural Challenge: Mass Market versus Proprietary Components, 17 (10), 1271-1316.

8. E. S. Hertel, et al., *CTH: A Software Family for Multidimensional Shock Physics Analysis*, in Proceedings 19th International Symposium on Shock Waves, 1, 274ff, Universite de Provence, France, 1993.

9. M. Rajan, et al., *Performance Analysis, Modeling and Enhancement of Sandia's Integrated TIGER Series (ITS) Coupled Electron/Photon Monte Carlo Transport Code*, in Proceedings of LACSI Symposium, Santa Fe, NM, Oct. 2005.

10. D. A. Bader et al., *Designing Scalable Synthetic Compact Applications for Benchmarking High Productivity Computing Systems*, Cyberinfrastructure Technology Watch, 2 (4B), 1-10, Nov. 2006.

11. CCS4 Transport Methods home page, Los Alamos National Laboratory, http://www.ccs.lanl.gov/CCS/CCS-4/code.html.

12. T. A. Haill, et al., *Multi-dimensional z-pinch calculations with Alegra*, in Pulsed Power Plasma Science, IEEE, Las Vegas, NV, June 2001.