SAND2011-2530C

A Comparison of the Performance Characteristics of Capability and Capacity Class HPC Systems

Douglas Doerfler, Mahesh Rajan, Marcus Epperson, Courtenay Vaughan, Kevin Pedretti,
Richard Barrett, Brian Barrett
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185
dwdoerf@sandia.gov, mrajan@sandia.gov, mrepper@sandia.gov, ctvaugh@sandia.gov,
ktpedre@sandia.gov, rfbarre@sandia.gov, bwbarre@sandia.gov

ABSTRACT

In this paper we report on our recent performance investigations on our most recent capability system, Cielo (1.03 PFLOPS Cray XE6), and capacity system, Red Sky (264 TFLOPS Intel Nehalem, QDR InfiniBand Cluster). Tri-Lab (SNL, LANL, LLNL) applications used for acceptance of Cielo form the basis for our analysis and provide for a rich variety in computation and communication behavior. The architectural and application characteristics are evaluated for each platform at up to 16,384 cores using applications and micro-benchmarks to determine at what scale each platform is most effective. We investigate the performance differences seen between the two systems through deeper analysis of the application message characteristics, messaging infrastructure, and the effects of a light weight operating system.

Categories and Subject Descriptors

I.6.3 [**Applications**], J.2 [Physical Sciences and Engineering]

General Terms

High performance computing, message passing programming model, PetaScale performance, application scalability

Keywords

Parallel scaling, InfiniBand, HPC clusters, OS noise, MPI

1. INTRODUCTION[1]

Traditionally capacity systems were typically configured with less than a few thousand nodes, but have recently grown in size and have begun to target applications and users considered to be in the realm of capability systems. The distinction between these two kinds of systems has also been blurred by the advances in the processor technology and systems interconnect. Although the usage model differs, with capability systems targeted at a few users or a single user using the entire system, while a capacity system targets hundreds of simultaneous users, the question of performance from an application perspective still remains. Sandia's HPC investments and technology investigations have traditionally comprised of three classes of systems: capability, capacity and advanced architectures. Management decisions that fund HPC systems would certainly be aided by analysis of the kind we provide in this paper, pointing out their strengths and providing justification for continued investments in capability systems that permit application runs at extreme scales.

The two systems compared are Cielo, a Cray XE6 that uses AMD Magny Cours processors and Gemini interconnect, rated at 1.03 PFLOPS, and Red Sky a cluster that uses Intel Nehalem processors and QDR InfiniBand interconnect rated at over 264 TFLOPS. We have historically undertaken similar investigations comparing performance of ASCI Red against Cplant [1] and more recently comparing performance of Red Storm against Tri-Lab Linux Capacity Cluster (TLCC) [2]. That study showed that for many applications TLCC best served the needs of applications requiring 128 processing elements (PEs) or less. Performance degradations on TLCC when compared to Red Storm were caused by a few key factors, the impact of: NUMA coherency over-head, decreased memory bandwidth per core, process migration, and MPI global operations. Recently we reported that Cielo, a Cray XE6, is an improvement to its evolutionary precursors: Cray XT6, XT5, and XT4 [3]. We discussed the impact of the node and systems interconnect on the observed performance comparisons. We also pointed out the benefits of Cray's new node interconnect with the Gemini routing and communications ASIC. A recent paper [4] further probed into applications performance of Cielo mostly focusing on scaling up to 1,024 cores and showing the benefit of the interconnect with applications that send many small messages.

In this paper, we have benchmarked and analyzed six applications that we had successfully used for performance acceptance tests of Cielo [5]. The acceptance tests, a joint effort by the Tri-Lab (Sandia National Labs, Los Alamos National Lab, Lawrence Livermore Natonal Lab) and Cray teams. The Red Sky cluster, designed to meet the increasing demand for capacity computing cycles and with a goal to achieve mid-range (512 to 2048 PEs) scalability of our applications, was enabled by a joint partnership between Sandia, Sun Microsystems, Intel and Mellanox. This large system with dual socket Intel Nehalem quad-core processor nodes and QDR InfiniBand interconnect was one of the first such commodity clusters to achieve a top 10 position in the HPL Linpack benchmark. Performance comparisons of Red Sky to Red Storm (which often has served as our golden standard for scalability because of its balanced architecture) showed 2X improvement [6].

Although comparison of performance between Cielo and Red Sky is complicated by a number of factors: processor/node architecture, interconnect, OS, file system, compilers, and libraries, a number of interesting observations emerge from such

[1] This work was supported in part by the U.S. Depart¬ment of Energy. Sandia is a multi program 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.

an effort. The following findings, discussed in detail in subsequent sections, are particularly note worthy:

- The applications benchmarked span a range of physics of interest like hydrodynamics, particle transport, electrical device simulation and sparse solvers.
- The newer capacity clusters such as Red Sky are enabling efficient runs at mid-range scales of a few 1000's PEs for many of our applications.
- We have identified application characteristics, as typified by one of our benchmarks - Charon (an electrical devise simulation code) that benefits greatly from unique characteristics of Cielo while exposing limitations of Red Sky.
- Red Sky performance at small scales is better than Cielo in all cases. We discuss architectural factors to explain this comparing an 8 core Intel Nehalem Processor node to the 16 core AMD Magny-cours processor node.
- We analyze the impact of current trends that keep increasing core counts per compute node, particularly with respect to continued use of the MPI programming model.
- Deeper analysis of applications that reveal scaling limitations on Red Sky point to inordinate growth in the time spent in MPI global operations. MPI Profiles are used to show that the increased time although attributed to an MPI call such as Allreduce, is more precisely traceable synchronization time of the global operation impacted by loss of good load balance for preceding computations [8].
- We have conducted controlled experiments to shed light on the impact of OS noise using tools developed at Sandia for OS research [4 kitten, Pedretti reference]
- We have provided analysis of the performance differences seen between the two systems through deeper analysis of the messaging characteristics of each application.

In the Section 2 we begin with a description of the Red Sky and Cielo architectures. Section 3 looks at the major factors that impact performance and facilitate analysis of the application performance through simple benchmark data. Section 4 gives a brief description of the test application and the performance comparison with measurements up to 16,384 PEs. Section 5 provides an analysis of the observed performance and uses MPI profile and results from additional experiments on Red Sky to investigate the impact of OS noise on scalability.

2. ARCHITECTURE COMPARISONS

2.1 Red Sky Architecture

Red Sky is a modular system that consists of three sections, identified by the three names: Red Sky, Red Horizon, and Red Mesa, but integrated as required to meet programmatic and operational needs of our customers. Harnessing the total capability of these groups, the system currently holds the 14th rank in the Top500 Linpack benchmark measuring 433.5 TFLOPS utilizing 42,440 cores. However for the purposes of this paper all the results pertain to a group of racks in 3 rows of cabinets that has 2,823 dual socket/quad core nodes (22,584 cores) yielding a peak performance of 265 TFLOPS.

The Red Sky cluster is built with Sun Vayu blades for compute and service partitions. There are two nodes per blade. Each node has dual-sockets fitted with the Intel 5570, "Nehalem-EP" 2.93 GHz quad-core processors with each processor having three channels of an integrated memory controller connected to 6GB of 1333 MHz DDR3 SDRAM memory. Also shown in the block diagram are the cluster management 10/100 Mbps and Gigabit Ethernet channels integrated into the blade. The midplane (NEM) integrates 4x QDR IB switches. The Mellanox HCA connects the nodes to the IB router and has a peak bandwidth of 40Gbits/sec to the NEM modules.

Figure 1 shows the two 36-port QDR InfiniBand (IB) switch used per chassis and the port connections for the toroidal interconnect. For each of the 36-port switches (SW) twelve ports are used to connect to the node HCAs, nine to connect SW0 to SW1, and the remaining fifteen ports form the external X,Y,Z links for the 3D torus. Each row, consisting of 12 racks, forms a 6x2x8 (X,Y,Z) torus building block. The logical 6x6x8 (X,Y,Z) torus maps to physical 12x3x8 node configuration. Logical Y dimension "folds over" at the last physical row and the torus is completed in an adjacent rack of physical row 1. Logical X dimension skips every other rack in the physical X dimension. Logical Z dimension is self contained within a rack and is fixed at 8.

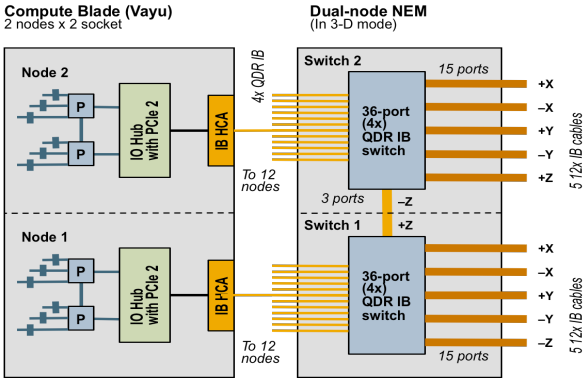


Figure 1. Red Sky node and IB interconnect (courtesy of Sun Microsystems)

The software environment on Red Sky uses the TOSS 1.3-4 which is based on RHEL 5 with some patches. The InifiBand uses the OFED 1.4.1 software stack configured with OpenSM Subnet manager incorporating a custom routing engine developed at Sandia for the 3D torus. The Lustre file system underlies the user's /home and /projects space. In addition, for image distribution NFS over IB is used. Slurm 2.1.15 and Moab 5.4-2 provides job scheduling and resource management capabilities. All the major compilers, Intel, PGI, and GNU are available for application development.

2.2 Cielo Architecture

Cielo is the latest ASC Tri-Lab capability computing system. It is a Cray XE6 [7] and incorporates the AMD 8-core Magny-Cours[8] processor and a new Cray interconnect called Gemini. The Cielo system in its current configuration consists of 6,704 dual processor compute nodes, for a total of 107,264 processor core elements and has a peak performance of 1.03 PFLOPS. The system will be upgraded in April 2011 to 8,894 compute nodes,

for a total of 142,304 cores and 1.37 PFLOPS peak performance. The XE6 node and interconnect are pictured in Figure 2. Each compute node has two AMD Opteron 6136 series processors as shown in the figure, with each socket consisting of two dies for a total of sixteen cores, arranged as four separate NUMA regions. 16 bit Hypertransport links with a peak bandwidth of 12.8 GB/s connect the dies along the edges as shown, while an 8 bit Hypertransport link connects them diagonally. As one would expect the NUMA nature of the node needs to be considered when optimizing node performance. Note the arrangement of the four DDR3 memory channels (two per die) providing direct access to 4GB DIMMS for a total of 32 GB at a node.

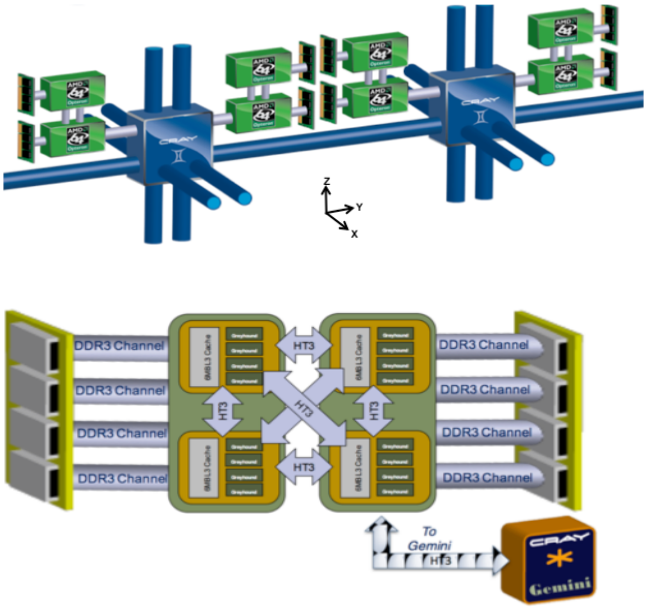


Figure 2. Cielo node and the Gemini interconnect (courtesy Cray, Inc,)

The Gemini interconnect is the heart of the Cray XE6 system. Capable of tens of millions of MPI messages per second, the Gemini ASIC is designed to support multicore processor nodes [7]. Each dual-socket node is interfaced to the Gemini interconnect through HyperTransport™ 3.0 technology. This direct connect architecture bypasses the PCI bottlenecks inherent in commodity networks and provides a peak of over 20 GB/s of injection bandwidth per node. The Gemini router's connectionless protocol scales from hundreds to hundreds of thousands of cores without the increase in buffer memory required in the point-to-point connection method of commodity interconnects. The significant improvement the Cray Gemini interconnect provides over the previous generation Cray interconnect, called SeaStar, is the message injection rate.

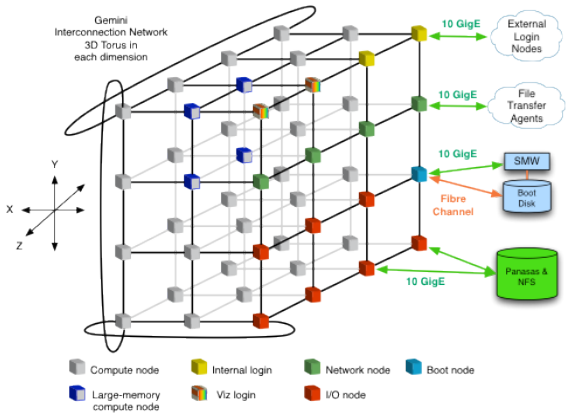


Figure 3. Cielo configuration diagram

Other features of the Gemini interconnect include improved resiliency, support for atomic memory operations, adaptive routing, improved bandwidth and latency. [4]. The Cielo system is configured as an 18x16x24 3D-torus network. A pictorial representation of Cielo identifying the principal components is shown in Figure 3.

Table 1 provides a quick reference of the differences between two architectures for easy side-by-side comparison

Table 1. Red Sky and Cielo system comparison

SYSTEM	Red Sky	Cielo
Num Compute Nodes	2823	6704
Num Compute Cores	22,584	107,264
Processor	Dual Intel Nehalem 2.93 GHz	Dual AMD Magny-Cours, 2.4 GHz
Cores / node	8	16
Memory / Core	1.5 GB	2 GB
Peak Node GFLOPS	93.76	153.6
Memory	3 channels/socket, DDR3, 1333 MHz	4 channels/socket, DDR3,1333 MHz
Cache	L1=4x32KB I,D L2=4x256KB L3=8MB	L1=8x64 KB, I,D L2=8x512KB L3=12MB (10MB)
Interconnect / Topology	QDR InfiniBand, 3DTorus	Gemini, 3D Torus
Compute Node OS	TOSS	CNL
MPI	OpenMPI 1.4	MPT 5.1.4
Compilers Used	Intel 11.1	PGI 10.x; Cray CCE 7.x

3. PERFORMANCE FACTORS

When comparing performance of computing systems that vary in many respects, it is useful to identify key components and their separate impact with simple measurable metrics. This facilitates easier analysis of the six parallel applications we have studied. Red Sky and Cielo differ in the node processor, node memory hierarchy and layout, system interconnect, MPI library, compute node operating system, and compilers used in our benchmarks. All our application benchmarks used an 'MPI everywhere' model assigning one MPI task per core.

3.1 Processor and Memory impact

In this section the compute node characteristics are investigated to better understand the processor and memory architectural differences.

Table 2: Memory Bandwidth (GB/s) Cielo node

	Mem 0	Mem 1	Mem 2	Mem 3
Numa node 0	13434	6877	6770	5641
Numa node 1	7003	13809	5643	6819
Numa node 2	6864	5593	13866	6839
Numa node 3	5673	6707	6831	13795

The most significant difference is the number of cores on a node: Cielo has twice as many as Red Sky. This implies that when we compare scaling of MPI applications, for the same number of MPI tasks, Red Sky requires twice as many nodes as Cielo. The NUMA nature of the node architecture could have significant impact on the performance. Penalty for non-local memory access could be significant as shown in the Tables 2 and 3 showing the STREAMS Triad benchmark results with 4 MPI threads.

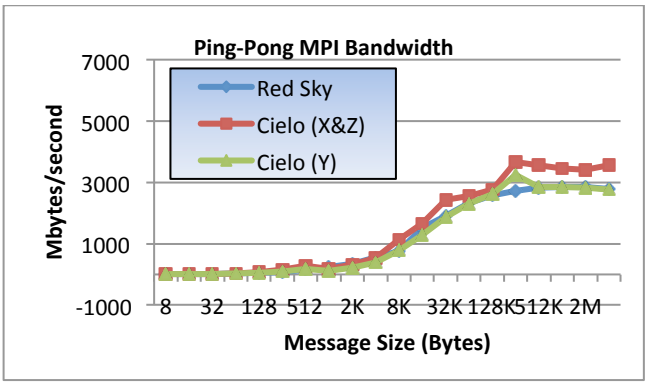
Table 3: Memory Bandwidth (GB/s) Red Sky node

	Mem 0	Mem 1
Numa Node 0	15400	8811
Numa Node 1	8757	15546

Observe that the per node and per core memory bandwidth on Cielo are 54GB/s and 3.375GB/s, while for Red Sky these are 33GB/s and 4.125GB/s. Both have 1333MHz memory DIMMS. Simple tests also show that the Nehalem processor has 4, 9, and 47 clock cycle latency to the three levels of cache while the Magny-Cours processor shows 4, 15 and 57 clock cycle latencies. Memory latency is approximately 81 ns for the Nehalem and 98 ns for Magny-Cours. It is interesting to compare MP Linpack performance on a node. We measured (with no special effort to tuning) on the Red Sky node about 91 GFlops and on the Cielo node about 125 GFlops. A hybrid Linpack run with threaded DGEMM was used. Comparing this to the peak node performance, Red Sky node is seen to have a higher fraction of the peak. AMD web page shows a SPECfp_rate for the Magny-Cours Opteron 6136 8core 2.4 GHz processor is 516/4 = 129. For the Nehalem 5570 2.93 GHz processor it is publicized as 197.

3.2 Interconnect Impact

In this section we present results of simple MPI benchmarks to gauge the impact of the node interconnect on application performance. Figure 4 shows the MPI node-to-node ping-pong bandwidth and latency plots using IMB benchmark [9] and assigning and pinning one MPI task per node. The two interconnects have similar latency except in the 100 to 1,024 byte range.



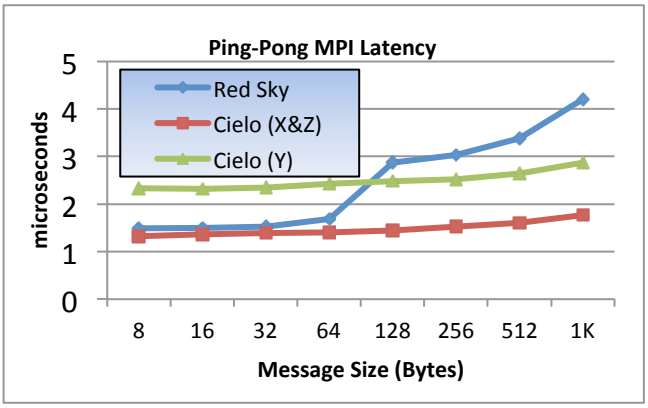


Figure 4. MPI Ping-Pong Bandwidth and Latency

A related streaming MPI ping-pong that posts multiple send /receive pair is useful in gauging the networks ability to process multiple outstanding messages. Results are shown in Figure 5 and shifting of the curve to the left is indicative of increasing messaging rate.

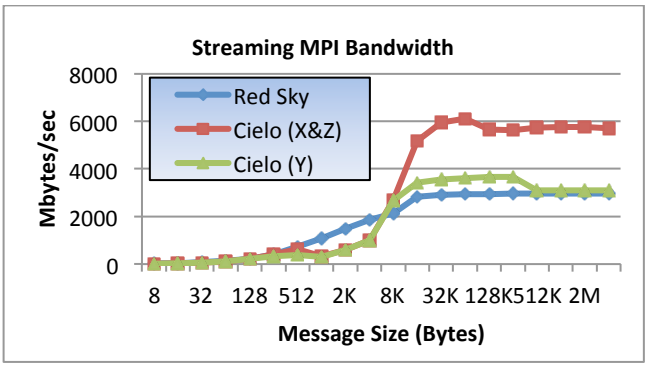


Figure 5. Streaming MPI Ping-Pong Bandwidth

In a Random messaging benchmark, thousands of small message sizes (varying from 100 bytes to 1KB) are sent to random MPI rank destinations. The messaging rate from each process and the average messaging rate are computed. The average message rate is compared in Figure 6. Observe the significantly lower performance on Red Sky. The message rate per PE goes from a factor of 10 slower at 32 cores to a factor of 220 slower at 8k cores. This simple MPI benchmarks that approximates messaging pattern for applications that sends lots of small messages among many MPI tasks approximates the characteristics of the application Charon discussed in section 4.

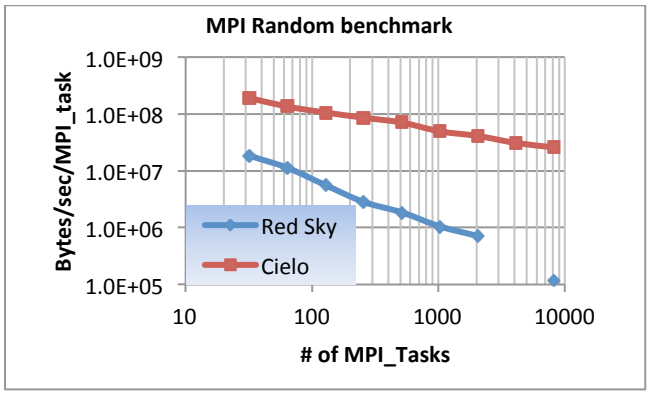


Figure 6. Random MPI Messaging Bandwidth

3.3 MPI Globals Impact

A few of the applications that we discuss in section 4 are sensitive to MPI time in global operations such as MPI_ALLREDUCE. In this section we provide comparative performance between Cielo and Red Sky.

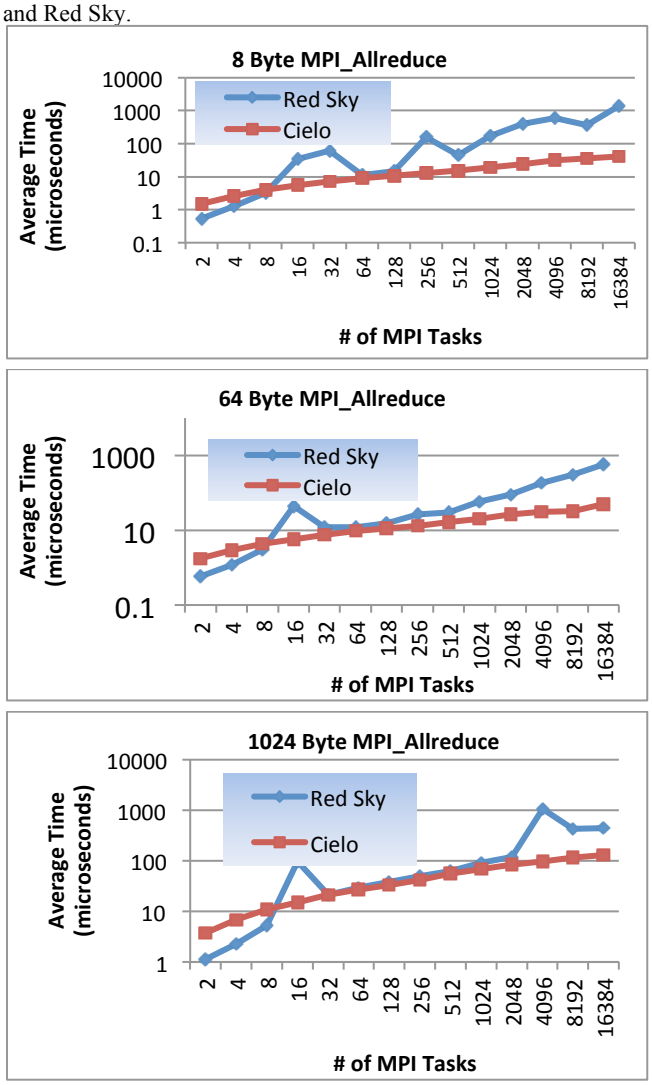


Figure 7.. MPI_Allreduce performance

The factors that influence these observed differences are many: different MPI implementation, node and interconnect hardware and software. However as the applications are impacted by the same differences it is instructive to do this comparison. Figure 7 shows three different MPI_ALLREDUCE comparisons, using 8 byte, 64 byte and 1,204 byte transfers. Observe order of magnitude difference at large scales. It should also be noted that Red Sky showed a very large variance in results from one run to the next. The results plotted are the minimum of three trials. This is further investigated in section 5.

4. APPLICATION PERFORMANCE

Six applications that were used for the Cielo application acceptance tests and scale tested on it to greater than 64k cores constitute our application benchmarks . The six applications were drawn from physics and engineering simulations of interest to the Tri-Labs and varied greatly in their targeted applications, programming languages and parallel algorithms.

4.1 CHARON

Charon is a semiconductor device simulation code [6] designed for use on high performance parallel computers using the MPI-everywhere model. The drift-diffusion model is used, which is a coupled system of nonlinear partial differential equations that relate the electric potential to the electron and hole concentrations. Finite element discretization of these equations in space on an unstructured mesh produces a sparse, strongly coupled nonlinear system. A fully-coupled implicit Newton-Krylov approach is used: the equations are linearized with Newton's method, and a Krylov solver is used for the solution of the sparse linear systems. A multigrid preconditioner is used to significantly improve scaling and performance. The FOM is the time per linear solve iteration. Communication required by the multigrid preconditioner is complex. The smoothers on each level require communication with nearest neighboring sub domains. Projection/restriction operators between levels need to be produced, the solutions and residuals need to be transferred between levels, and the coarser levels need to be generated with a triple matrix product. The coarsest level solve requires a serial direct factorization.

The performance of Charon (weak scaling study with about 31,000 DOF/core) is shown in Figure 8. The topological communication pattern showing the communication count for a 32 way parallel run is illustrated in Figure 9 and is the most complex communication structure of the six applications studied.

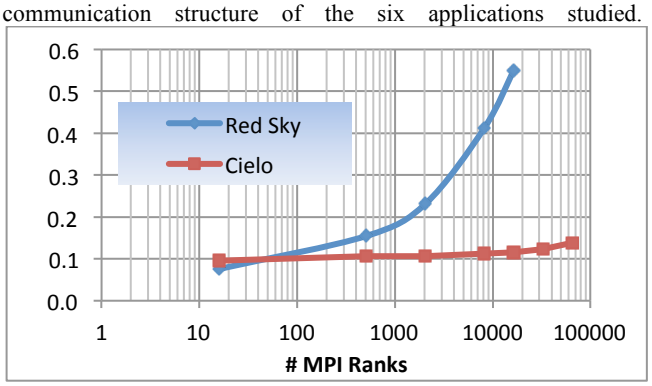


Figure 8 Charon Solve Time per Iteration (Lower is better)

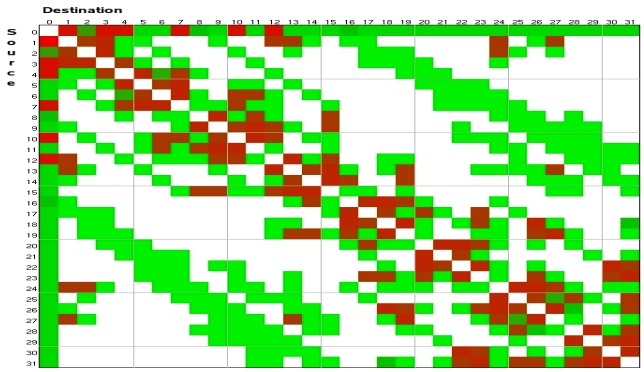


Figure 9. Charon inter-processor communication CrayPat plot

4.2 CTH

CTH is a multi-material, large deformation, strong shock wave, solid mechanics code developed at Sandia National Laboratories [7]. CTH has models for multi-phase, elastic viscoplastic, porous and explosive materials, using second-order accurate numerical methods. For these tests, we used the shaped charge problem, in three dimensions on a rectangular mesh. The weak scaling configuration places a grid of size (x, y, z) = (80, 192, 80) cells onto each PE. The Figure of Merit is the wall time required to perform 100 time steps, so lower is better. Computation is characterized by regular memory accesses, and is fairly cache friendly, with operations focusing on two dimensional planes. Inter-process communication aggregates internal-boundary data for all variables into message buffers, subsequently sent to up to six nearest neighbors. For the problem studied here, this maximum number of neighbors is reached once 128 cores are employed and each message is on the order of 3 MBytes. Figure 10 shows the scaling plot. The good scaling behavior on both Red Sky and Cielo is because CTH's very large message aggregation scheme. Each time step, CTH makes 90 calls to MPI collective functionality (significant, but about seven times fewer than Charon), 19 calls to exchange boundary data (two dimensional ``faces''), and three calls to propagate data across faces (in the x, y, and z directions).

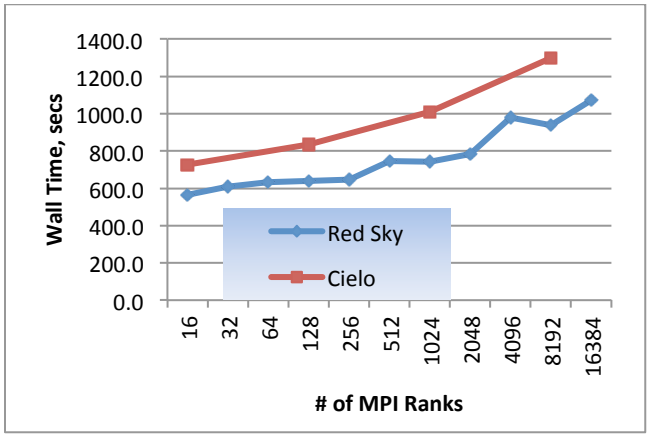


Figure 10. CTH Scaling Performance (Lower is better)

Collective communication is typically a reduction (Allreduce) of small message sizes. Figure 11 shows the topological communication pattern for 128 cores, illustrating the nearest neighbor communication pattern.

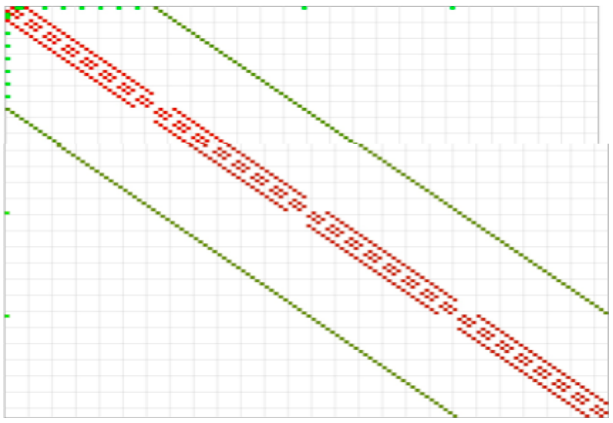


Figure 11. CTH inter-processor communication CrayPat plot

4.3 xNobel

xNOBEL is a one, two, three dimensional, multi-material Eulerian hydrodynamics code developed for solving a variety of high deformation flow of materials problems, with the ability to model high explosives[8]. Runtime is communication intensive, requiring the transmission of many relatively small messages. The benchmark is a 3D simulation of a 105 mm shaped charge calculation. This run exercised Continuous Adaptive Mesh Refinement (CAMR) portions of the code and high-explosive burn models. The model defined is an Octant of the full 3D problem. The weak scaling benchmark executes 50 cycle runs and uses a Figure of Merit: cyc_cc/sec/pe. Larger values of FOM are better and implies greater computational rate. This measure is inversely related to the average execution time measured as seconds/cycle. Figure 12 shows the weak scaling behavior and Figure 13 the prediminently nearest neighbor topological communication pattern.

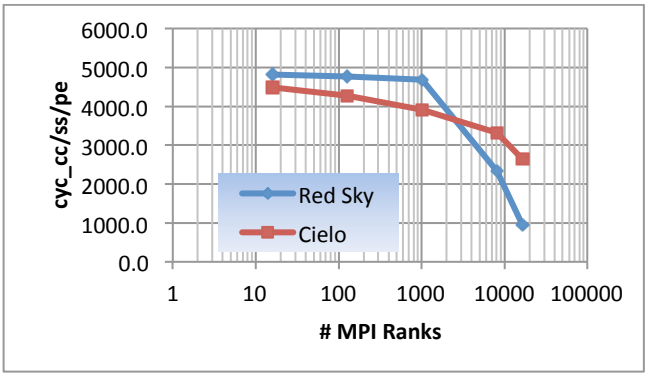


Figure 12. xNobel Scaling Performance (Larger is better)

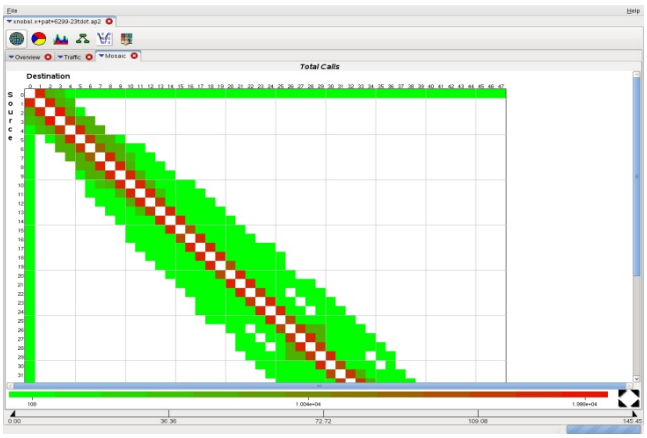


Figure 13. xNOBEL inter-processor communication CrayPat plot

4.4 SAGE

SAGE (SAIC Adaptive Grid Eulerian) is a multi-dimensional multi-material shockwave Eulerian hydrodynamics code that uses Adaptive Mesh Refinement [9]. We benchmarked SAGE in a weak-scaling mode. The input deck (timing_h) used assigns 17,500 cells to each processor core. Although the problem set used here does not include mesh refinement, the code does not take advantage of the static nature of the memory layout, and thus the runtime profile is representative. For example, inter-process communication is through a bulk-synchronous gather/scatter abstraction, which collects off-process data and inserts it into doubly indexed arrays; the receiver unpacks the message using doubly indexed arrays. The weak scaling Figure of Merit for SAGE, shown in Figure 14, is the wall time for 10 iterations and lower values are better. At the large core counts of 8k and 16k the default OpenMPI parameters led to run time failures on Red Sky. This was overcome by the use of OpenMPI btl parameters settings to increase the default settings of message exchange pairs. Figure 14 also shows the weak scaling performance with and without the these OpenMPI parameters. Figure 15 shows the CrayPat plot illustrating the topological communication pattern.

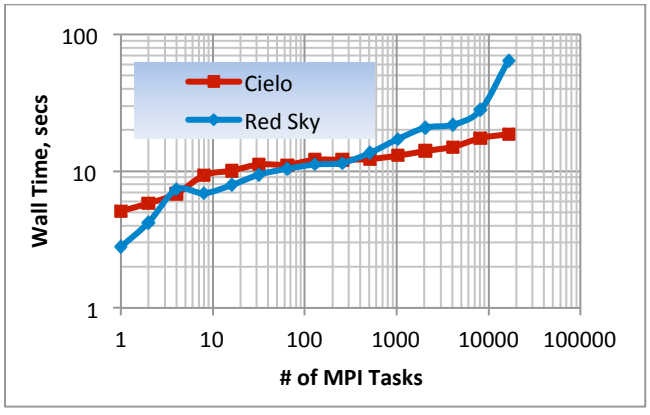


Figure 14. SAGE Scaling Performance (Lower is better)

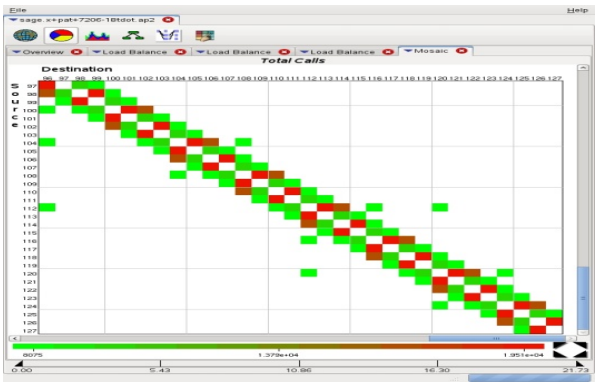


Figure 15. SAGE inter-processor communication CrayPat plot

4.5 AMG2006

AMG2006 is a parallel algebraic multigrid solver of linear systems arising from problems on unstructured grids. Based on Hypre[11] library functionality, the benchmark, configured for weak scaling on a logical three dimensional processor grid (px*py*pz) solves the Laplace equations on a global grid of dimension px*220 X py*220 X pz*220. The Figure of Merit measures the solve phase time for the preconditioned conjugate gradient solver for 100 iterations (lower is better). Performance is shown in figure 16. Runtime is dominated by the memory bandwidth requirements of the sparse matrix-vector product at small core counts and by MPI Allreduce function at large core counts with a message size of about 2 Kbytes. The other MPI routines, mostly non-blocking point-to-point communication, consume a negligible small fraction of the communication cost. Red Sky out-performs Cielo with both platforms showing near perfect scaling (after node memory bandwidth limitation is reached) as shown in Figure 17, except at 8k and 16k cores. In section 5 detailed analysis with the help of MPI profiles explains the sudden jump in run time at 8k and 16k cores on red Sky. The message communication plot is shown in Figure 17.

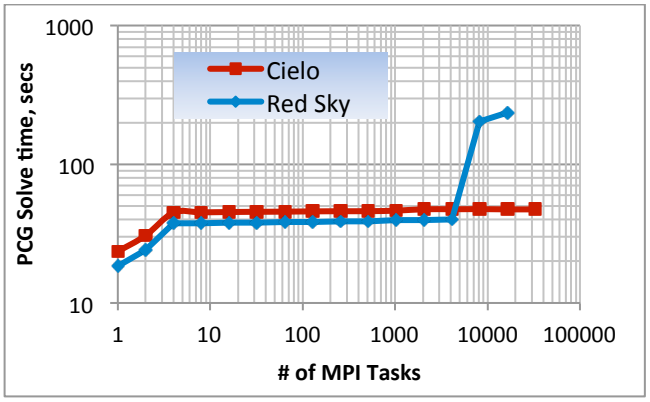


Figure 16. AMG Scaling Performance (Lower is better)

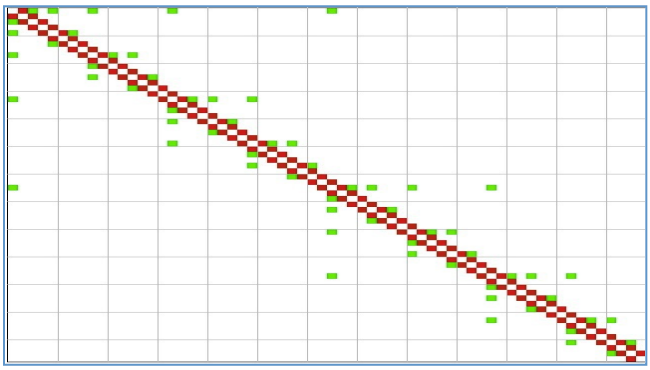


Figure 17. AMG inter-processor communication CrayPat plot

4.6 UMT2006

The UMT benchmark is a 3D, deterministic, multigroup, photon transport code for unstructured meshes. The deterministic transport code solves the first-order form of the steady-state Boltzmann transport equation. The equation's energy dependence is modeled using multiple photon energy groups. The angular dependence is modeled using a collocation of discrete directions, or "ordinates." The spatial variable is modeled with an "upstream corner balance" finite volume differencing technique. The solution proceeds by tracking through the mesh in the direction of each ordinate. For each ordinate direction all energy groups are transported, accumulating the desired solution on each zone in the mesh. Hence, memory access patterns may vary substantially for each ordinate on a given mesh, and the entire mesh is "swept" multiple times. Note, however, that having the energy group loop on the inside significantly improves cache reuse, because all of the geometrical information related to sweeping an ordinate direction is the same for each energy group. The code works on unstructured meshes, which it generates at run-time using a two-dimensional unstructured mesh (read in) and extruding it in the third dimension a user-specified amount. This allows the generation of a wide variety of input problem sizes and facilitates "constant work" scaling studies. The MPI-based parallelism in the Fortran portion uses mesh decomposition to distribute the mesh across the specified MPI tasks. The OMP-based parallelism in the C kernel then divides the ordinates among the OMP threads. This C kernel's computation time typically completely dominates the execution time of the benchmark.

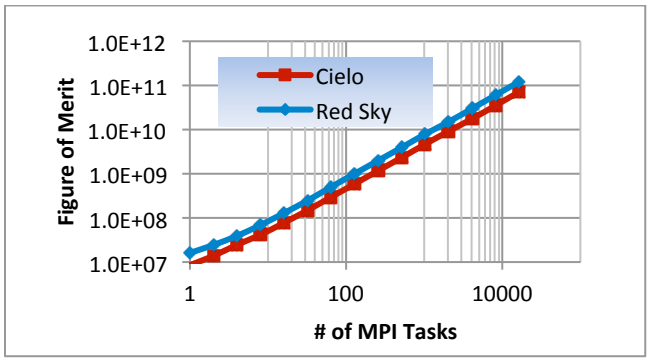


Figure 18. UMT Scaling Performance (higher is better)

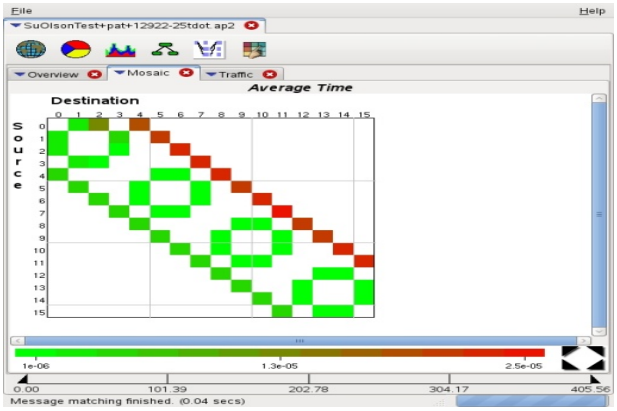


Figure 19. UMT inter-processor communication CrayPat plot

5. ANALYSIS

In all of the six applications we benchmarked Red Sky performs very well at small scale, but for some applications performance degrades, and at some crossover point Cielo performance is better. This is illustrated by looking at the ratio of Cielo to Red Sky performance, Figure 21.

For CTH and UMT Red Sky outperforms Cielo at all tested scales. For Charon, the crossover occurs with as few as 512 MPI ranks and Cielo shows a near 5x improvement at 16,384 ranks. AMG performance on Red Sky scales well up to 4,096 ranks and degrades dramatically at 8,192. SAGE's crossover is at 1,024 MPI ranks and xNobel's performance crosses somewhere between 1,024 and 8,192 ranks. From this analysis a finding from our observations emerges: Red Sky is eminently suited for most capacity sized jobs and has met its design goal of achieving intermediate scalability. But what is the cause for the performance degradations at scale? In the remainder of this section we take a deeper look at Red Sky's performance and investigate the architectural characteristics of the platform that may be at cause.

In Section 5.1 we look at the messaging characteristics of each application. Section 5.2 investigates the operating system (OS) effects by comparing performance of the nominal Red Sky OS and a striped down version mimicking a light-weight OS with reduced OS noise characteristics.

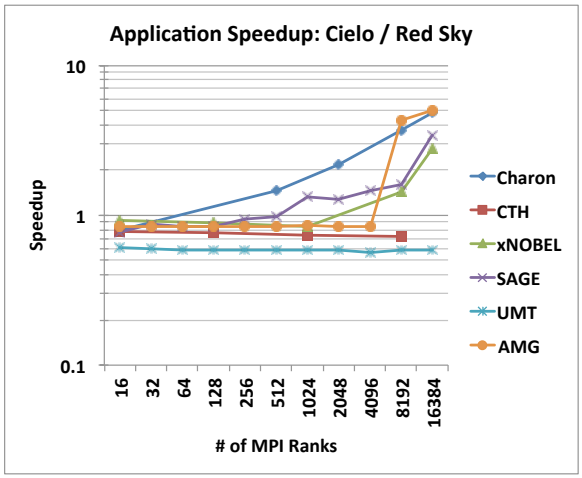


Figure 21. Cielo application speedup relative to Red Sky

5.1 Analysis of Message Characteristics

The factors that impact the messaging overhead are principally the number of collective operations, the point-to-point message communication pattern, the message frequency, and the message size. The point-to-point message communication patterns for the six applications are depicted in Figures 10, 12, 14, 16, 18 and 20. Charon's point-to-point communication pattern is much denser than the other applications. The hydrodynamics applications, CTH, SAGE and xNobel have predominantly nearest neighbor communications. AMG and UMT also show a sparse communication pattern, with only nearest neighbor point-to-point communication.

Figure 22 illustrates the messaging frequency (MPI calls per minute), binned by message size, for point-to-point calls in each application. The far right hand column is the number of bytes per second sent by the application, and the column that is second from the right is the total number of messages per minute for all message sizes.

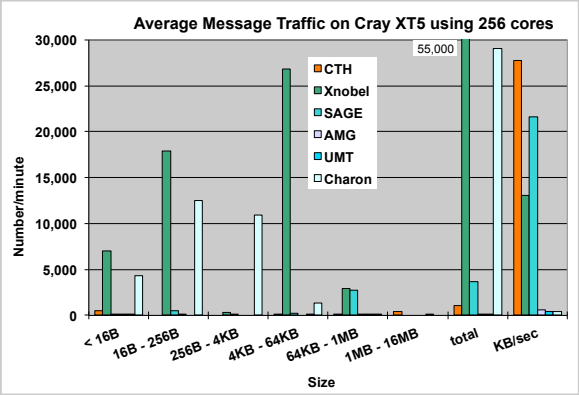


Figure 22. Application messaging characteristics: message size and message frequency

Charon and xNobel clearly demonstrate the highest small message frequencies. In fact their total frequency for all message sizes clearly dwarfs the other applications. CTH and SAGE are mostly dominated by 1MB to 3MB message calls at a relatively low frequency, but as can be seen in the last column they have a lot higher demand on network bandwidth than Charon. Note that xNobel has a high message frequency and network bandwidth demand.

It's also instructive to look at MPI profile results to identify anomalies that correlate to an unexpected degradation in application performance. In the remainder of this section we will focus on AMG and Charon as those two applications demonstrated the largest performance degradation at scale on Red Sky. We will not present all profile data collected and analyzed, but focus on only those results that provide insight into application performance characteristics.

The MPI profiles for AMG show that a large fraction of time is being spent in the Allreduce operation at 8,192 and 16,384 MPI ranks, Figure 23. MPI overhead suddenly grows from less than 10% to about 80% at these scales. This result directly correlates with the application's performance degradation on Red Sky at those scales and we can conclude that for AMG the principal reason for performance degradation is clearly related to the inordinate growth in time of MPI_Allreduce.

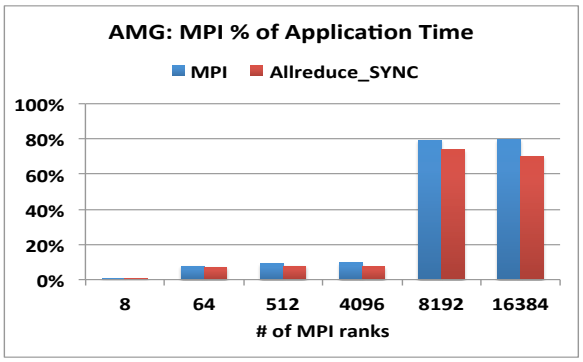


Figure 23. AMG MPI Profile

Allreduce can be broken down into two parts: 1) the time for the actual reduction operation once all ranks have arrived at that call point, and 2) MPI_SYNC time. MPI_SYNC time growth is symptomatic of some inherent and/or extraneous interference that destroys the load balance of the computations preceding this global operation, and it measures the time delay for the last function to arrive at Allreduce from the time first MPI rank arrives. AMG has been shown to scale well on Cielo, so there is no inherent application load imbalance, hence the increased MPI_SYNC time is due to extraneous interference at the machine level. This is further investigated in Section 5.3.

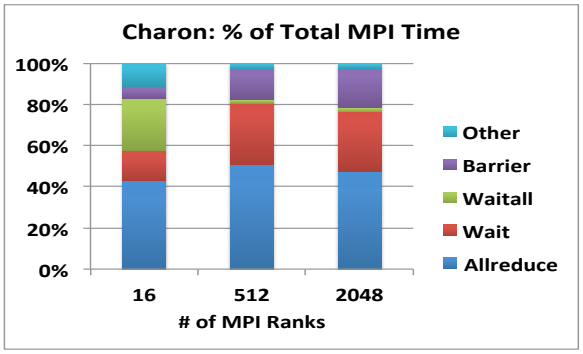


Figure 24. Charon MPI Profile

The MPI profiles for Charon show that the dominate MPI operations are Allreduce, Wait and Waitall, with the later two being an artifact of non-blocking point-to-point operations, Figure 24. A large fraction of time is also spent in the Barrier operation, which is most likely due to inherent load imbalance in the application. The time spent in the Allreduce operation is essentially constant with increasing scale. It can be concluded that the Allreduce operation is not responsible for Charon's poor scaling characteristics on Red Sky.

Charon does exhibit a few communication traits that are unique amongst the six applications. It primarily sends relatively small messages and its point-to-point communication pattern spans a much larger fraction of peers. In Section 3.2, the MPI Random Benchmark demonstrated that Cielo is much more capable than Red Sky in sending messages throughout the machine. This result and the observed messaging characteristics of Charon provide one potential explanation of its better performance on Cielo.

The evidence that AMG is impacted by poor Allreduce performance is clear. But why is Allreduce poor on Red Sky? Charon is sensitive small message performance, but the

observation that Cielo outperforms Red Sky in random message traffic is not a smoking gun. In the next section we look at the impact of OS noise as it has been demonstrated that it can have a significant impact on MPI point-to-point and collective communications performance at large scales. [20, 21]

5.2 Analysis of OS Noise Impact

The impact of OS interference on the measured performance of applications was investigated by analysis of the application performance on a Red Sky test bed. In this section all results were collected on the test bed using the nominal Red Sky OS (HWOS) and the Red Sky environment using a Linux kernel with modifications (LWOS). The size of the Red Sky test bed limited our analysis to 64 nodes, 512 cores. This constrained our analysis to only those application anomalies that were observed at 512 ranks or less.

The LWOS environment was constructed using a 2.6.38.1 kernel configured with options to try and mimic some of the modifications made by Cray for Cielo's Compute Node Linux (CNL) kernel, combined with a minimal in-memory root file system and runtime environment. The intent was to reduce the system noise while retaining application binary compatibility, so that direct performance comparisons could be made on the same hardware using the same MPI, compiler, and other user space libraries. Noteworthy differences between the HWOS and the LWOS include:

- Lower-frequency timer interrupts (from 1000 HZ to 250 HZ)
- Balanced timer interrupt handling, i.e. no single core taking all timer interrupts
- Fewer system daemons
- No periodic system health monitoring processes

It should be noted that some of the micro-benchmarks in Section 3 were rerun with the LWOS and one negative effect is that small message (<64 byte) latency increased by an average of 1 microsecond compared to the HWOS environment. This deserves further investigation and may indicate a regression in the upstream kernel InfiniBand stack.

The Fixed Work Quantum (FWQ) component of the Netgauge [22] benchmark suite was used to evaluate the effectiveness of the LWOS environment for reducing OS interference. FWQ benchmark repeatedly times how long it takes to perform a small, fixed amount of computational work. In an OS interference free environment, the computational work will always execute in the same amount of time. In an environment with OS interference, some of the iterations will take longer to execute because they are interrupted by the OS. As can be seen in Figure 25, the LWOS environment is very effective in reducing OS interference. The HWOS environment demonstrates a fairly regular pattern of spikes with magnitudes several times the baseline. This is likely due to OS kernel threads or daemons interfering with the FWQ benchmark's execution. In contrast, the LWOS environment demonstrates no significant deviations from the baseline, indicating a nearly noise free environment. As we will show, this has a positive effect on the performance and repeatability of MPI collective operations.

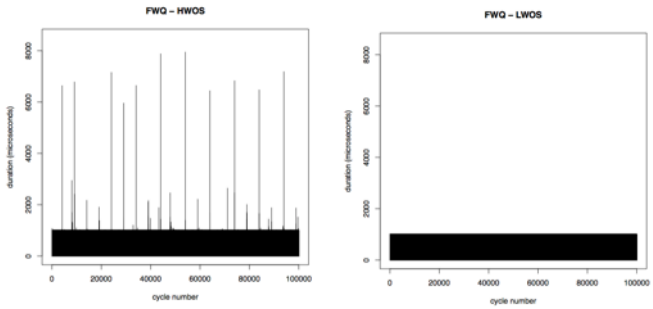


Figure 25. Netgauge fixed work quantum (FWQ) benchmark on HWOS and LWOS

In the previous section it was shown that AMG performance degradation at scale is due to increases in time spent in the MPI Allreduce operation. Allreduce performance was measured for both OS environments, Figure 26. Ten trials of the benchmark were collected and the median is used for the plotted data series with the maximum and minimum of the trials forming the error bars. This experiment clearly demonstrates that Allreduce performance is heavily impacted by OS noise and provides evidence that OS noise on Red Sky is the cause of the performance degradation at 8,192 MPI ranks, Figures 16 and 21.

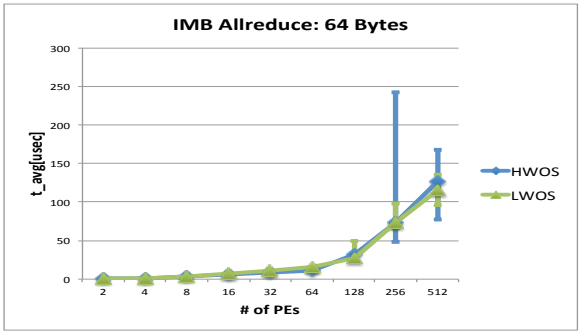


Figure 26. MPI Allreduce performance using the HWOS and LWOS environments on the Red Sky test bed

Charon performance for the two OS environments is shown in Table 4. The LWOS environment on the Red Sky test bed demonstrates a 9.7% improvement over the HWOS environment for Charon, and a 28% improvement over Red Sky[2]. It should also be noted that the LWOS approaches the performance of Cielo. If this can be shown to be true at larger scales then the authors would argue that for Charon the effects of OS noise on small message point-to-point operations is the reason for Charon's poor scalability on Red Sky.

Table 4. Charon solve_t/iteration (sec) at 512 MPI ranks

Cielo	Red Sky	HWOS	LWOS
0.1068	.1546	0.1231	0.1111

Charon was the only application that demonstrated a significant improvement using the LWOS at the 512 MPI rank scale tested on the Red Sky test bed. However, xNobel using the LWOS

[2] It's not yet understood why the Red Sky test bed outperformed Red Sky in this experiment, but it's not an unreasonable result. Given Charon's sensitivity to small message performance the test bed's smaller network diameter may provide better latency characteristics. But it merits further investigation.

environment showed steady improvement as scale increased, 0.31% at 64 ranks, 0.98% at 256 ranks and 1.9% at 512 ranks. This may be an indicator of xNobel's performance degradation after 1,024 ranks on Red Sky, Figures 8 and 21.

6. CONCLUSIONS AND FUTURE WORK

A large amount of effort was put into defining a suite of applications to be used as one of the acceptance criteria for the ASC's Cielo capability platform. It has been a long-standing practice at Sandia to periodically compare the performance of current instantiations of capacity and capability computing platforms in order to better understand their architectural and performance characteristics. This allows us to better understand issues for future acquisitions and to provide input on areas for improvement for current systems. In this study we leveraged the work put into the Cielo applications acceptance testing by repeating those tests on our Red Sky capacity platform.

Results showed that Red Sky performs very well relative to Cielo for all of the applications, but shows performance degradation for four of the six tested applications at some scale. Historically small-scale performance has always been good for capacity class systems when compared against capability class systems, but Red Sky demonstrated that it can push the crossover point to 1000's of MPI ranks, or more, were previously the crossover point was in the 100's of ranks. Red Sky has meet its design goal of a mid-range, intermediate-scale platform.

Further investigations were made to figure out the cause of some of the performance degradation on Red Sky. MPI profiling showed that the six applications varied greatly in their message passing characteristics and in particular that AMG was sensitive to a dramatic increase in MPI_ALLREDUCE times. Further investigation using a Red Sky test bed and a light-weight operating system (LWOS) showed that Charon's performance was improved significantly in the LWOS environment and is sensitive to OS noise, surmised to be the impact of OS noise on small message performance of the high-speed network.

Next steps are to further develop the LWOS environment and investigate its impact at full scale. Although it should be noted that Red Sky is a production resource in high demand, and it is very difficult to get dedicated time to study applications and system software modifications at scale. The goal is to feed these findings and improvements back into the production Red Sky environment and potentially the other capacity clusters at Sandia and the ASC Tri-labs, in particular the TLCC capacity clusters.

7. ACKNOWLEDGMENTS

This work leveraged the Cielo application acceptance effort, which included members from Cray Inc, Sandia National Laboratories, Los Alamos National Laboratory and Lawernce Livermore National Laboratory.

This work was supported in part by the U.S. Department of Energy. Sandia is a multi program 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.

8. REFERENCES

[1] Henson, V.E. and Yang, U.M. 2002. BoomerAMG: A Parallel Algebraic Multigrid Solver and Preconditioner", , *Appl. Num. Math. 41*.

[2] Kerbyson, D.J., Alme, H.J., Hoise, A., Petrini, F., Wasserman, H.J., and Gittings, M. 2001. Predictive Performance and Scalability Modeling of a Large-Scale Application. *Proceedings of the IEEE/ACM SuperComputing (SC01)*, Denver, CO, November 2001".

[3] Barker, K., Davis,K., Hoisie, A., Kerbyson, D.J., Lang, M., Pakin, S., and Sancho. J.C. 2008. A Performance Evaluation of the Nehalem Quad-core Processor for Scientific Computing. In *Parallel Processing Letters*, December 2008.

[4] Brightwell, R., Camp, W.J., Cole, B.J., DeBenedictis, E., Leland, R.W., Tomkins, J.L., and Maccabe, A.B., 2005. Architectural specification for massively parallel computers: an experience and measurement-based approach, *Concurrency and Computation: Practice and Experience*, August 2005

[5] Doerfler, D., et.al., 2011. Application-Driven Acceptance of Cielo, an XE6 Petascale Capability Platform, To be presented at *CUG 2011* (Fairbanks, AL, May 23-26, 2011)

[6] Hertel, E.S. Jr, et.al., 1993. CTH: A Software Family for Multi-Dimensional Shock. *Physics Analysis*.

[7] http://developer.amd.com/zones/Magny-Cours/pages/default.aspx

[8] http://software.intel.com/en-us/articles/intel-mpi-benchmarks/

[9] http://www.cray.com/Products/XE/Systems/XE6.aspx

[10] http://www.unixer.de/research/netgauge/osnoise/

[11] Ferreira, K., Bridges, P., and Brightwell, R. 2008, Characterizing Application Sensitivity to OS Interference using Kernel-Level Noise Injection. *In Proc. Of the 2008 ACM/IEEE Conference on Supercomputing*, pages 1-12.

[12] Oral, S.,Wang, F., Dillow, D.A., Miller, R., Shipman, G.M., and Maxwell, D., 2010. Reducing Application Runtime Variability on Jaguar XT5. *CUG 2010*, Edinburgh, May 24-27, 2010

[13] Lin, P.T. and Shadid, J.N. 2010. Towards Large-Scale Multi-Socket, Multicore Parallel Simulations: Performance of an MPI-only Semiconductor Device Simulator. *Journal of Computational Physics, 229(19)*.

[14] Rajan, M., and Doerfler, D. 2010. HPC application performance and scaling: understanding trends and future challenges with application benchmarks on past, present and future Tri-Lab computing system, *ICNAAM 2010*, Rhodes, Greece, Jul 21 -24, 2010.

[15] Rajan, M., Vaughan, C.T., Doerfler, D., Epperson M., and Ogden, J.D. 2009. Application Performance on the Tri-Lab Linux Capacity Cluster – TLCC, *International Journal of Distributed Systems and Technologies*, Volume 1, Issue 2.

[16] Saini, S., Naraikin, A., Biswas, R., Barkai, D., and Sandstrom, T. 2009. Early Performance Evaluation of a "Nehalem" Cluster Using Scientific and Engineering Applications, *SC09*, Portland Oregon, 2009.

[17] Hoefler, T., Mehlan, T., Lumsdaine, A., and Rehm, W. 2007 Netgauge: A Network Performance Measurement Framework. In High Performance Computing and Communications, *Third International Conference, HPCC* Houston, USA, September 26-28, 2007, Proceedings, volume 4782, pages 659–671. Springer, 9 2007.

[18] Hoefler, T., Shneider, T., and Lumsdaine, A. 2010 Characterizing the Influence of System Noise on Large-Scale

Applications by Simulation, *SC10: The International Conference for High Performance Computing, Networking, Storage and Analysis*, New Orleans, LA, November 2010.

[19] Gittings, M.L., Weaver, R.P., et al. 2008. The RAGE radiation-hydrodynamic code , *Computational Science and Discovery, Vol 1,* 2008.

[20] Vaughan, C.T., Rajan, M., Doerfler, D., and Barrett, R.F. 2010. *Poster:* From Red Storm to Cielo: Performance Analysis of ASC Simulation Programs Across an Evolution of Multicore Architectures. *IN SC'10: Proceedings of the 2010 ACM/IEE conference on supercomputing*.

[21] Vaughan, C.T., Rajan, M., Doerfler, D., and Barrett, R.F., and Pedretti, K. 2011. Investigating the Impact of the Cielo Cray XE6 Architecture on Scientific Application Codes. To be presented at the *IPDPS workshop on Large-Scale Parallel Processing*, Anchorage, AL, May 16-20. 2011.