Application-driven Analysis of Two Generations of Capability Computing Platforms: Purple and Cielo

Mahesh Rajan, Courtenay Vaughan, Richard Barrett, Doug Doerfler, Paul Lin, and Kevin Pedretti
Sandia National Laboratories
Albuquerque, NM, USA
Email: ctvaugh, mrajan, rfbarre, dwdoerf, ktpedre@sandia.gov

***Abstract*—Cielo, a Cray XE6, is the Department of Energy NNSA Advanced Simulation and Computing (ASC) campaign's newest capability machine. Rated at 1.37 PFLOPS, its primary mission objective is to enable a suite of the ASC applications implemented using MPI to scale to tens of thousands of cores. Towards that end, a primary acceptance criteria for the initial phase of Cielo was to demonstrate a six times (6x) performance improvement on a suite of ASC codes relative to its predecessor, the Purple platform, an IBM Power5-based architecture. In this report we investigate the architectural characteristics of Cielo that enabled this level of performance.**

***Index Terms*—High performance computing; parallel architectures; message passing communication; performance evaluation; scientific applications.**

I. Introduction

Note to reviewers: due to the only recent installation and acceptance of Cielo, the program committee has granted us a one week extension. However, Julian Kunkel asked that we submit a preliminary draft so that it may be incorporated into the review process. We expect that the general discussion and results herein will remain relatively unchanged (with the possible exception of additional results). We apologize for any inconvenience this may cause.

Cielo, a Cray XE6, is the Advanced Simulation and Computing (ASC[1]) Campaign's newest capability machine. Rated at 1.37 PFLOPS, Cielo represents the latest evolution in multicore-based HPC architectures.

The two major programmatic requirements for Cielo are ease of migration of existing ASC multi-physics applications and strong performance of these applications at capability scale[4], [2]. Recently we reported that Cielo is an improvement to its evolutionary predecessors from Cray[12], [13], providing evolutionary and possibly revolutionary capabilities that may be important in future code configuration issues, especially as we progress to even larger computing scales.

In this report we examine the performance capabilities of Cielo in relation to its mission predecessor, Purple. The performance requirement of a six times improvement with regard to application capability presented unique challenges as this passage between computer generations spanned the transition of computing to the multi-core era. In particular, single processor compute power remained somewhat stagnant, pressure upon on-node bandwidth accelerated, and node interconnect capabilities evolved to support these changes.

Our focus is on codes from ASC Applications Acceptance Test suite, which represent a broad set of requirements of the ASC Tri-lab organizations (Lawrence Livermore, Los Alamos, and Sandia National Laboratories). Although it is difficult to attribute performance effects clearly, our interpretation of the results lead us to some strong conclusions. First, the dual-socket Magny-Cours-based node architecture, with four NUMA regions each with independent memory controllers and dual-channel DDR3 memory configuration, improves support for the bandwidth requirements of our applications. Further, we find that the Gemini interconnect provides an evolutionary performance improvement to codes that send large messages. More importantly, we find that codes that send many small messages realize significantly stronger performance, an issue critical to effective use of very high processor counts, which expected to be critical at the exascale[1].

This report is organized as follows: We begin with a description of the Purple and Cielo architectures. We include micro-benchmark results that help us understand the processor, node, and interconnect performance. We then describe our full application experiments, focusing on the issues required to achieve strong performance at large scale, followed by a summary of this work and our future plans.

II. Architecture Overviews

The ASC Purple platform[2] is Cielo's predecessor as the production capability computer for the ASC program. Sited at and operated by Lawrence Livermore National Laboratory, Purple was initially deployed in 2005 and retired in November, 2010. An instantiation of IBM's POWER Architecture, Purple consisted of 1,336 IBM p5 575 nodes connected by the Federation High Performance Switch, running IBM's AIX operating system. Although a dual-core architecture, for Purple only one core was enabled.

Cielo, an instantiation of a Cray XE6, is composed of AMD Opteron Magny-Cours processors, connected using a Cray custom interconnect named Gemini, and a light-weight kernel (LWK) operating system called Compute Node Linux. The initial system, delivered toward the end of 2010, consists of

[1]http://nnsa.energy.gov/aboutus/ourprograms/defenseprograms/futurescienceandtechnologyprograms/advancedsimulationandcomputin

[2]Additional details may be found at http://asc.llnl.gov/computing_resources/purple/.

6,654 compute nodes, for a total of 106,464 processor core elements, capable of 1.02 PFLOPS. The final system, to be delivered in the spring of 2011 will consist of 8,944 compute nodes, for a total of 143,104 cores. The system configuration is shown in Figure 1.

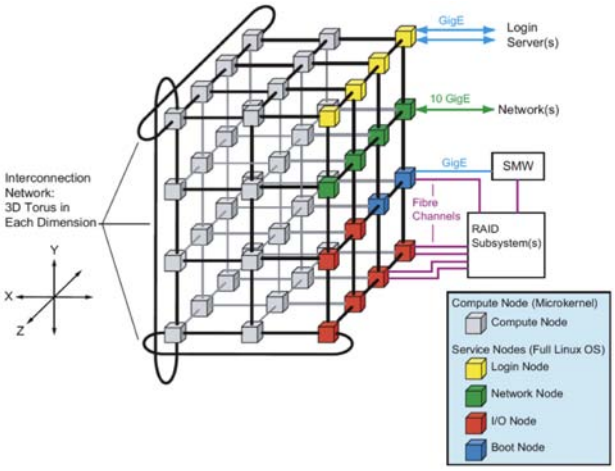


Fig. 1. Cielo XE6 architecture. Image courtesy of Cray Inc.

Two AMD Opteron 8-core Magny-Cours processors share a compute node, illustrated in Figure 2(a). Each Magny-Cours processor is divided into two memory regions, which AMD calls NUMA nodes, each consisting of four processor cores. Thus each compute node consists of 16 processor cores[3], evenly divided among four NUMA nodes, which are connected using Hyper Transport version. All links run at 6.4 GigaTranfers per second (GT/sec). So the 24-bit links between die in a processor run at 19.2 GigaBytes per second (GB/sec), the 16-bit links between processors run at 12.8 GB/sec per direction, and the "cross" 8-bit links between processors run at 6.4 GB/sec. The impact of this NUMA memory organization is investigated using the STREAMS benchmark, taking care to set processor and memory affinity of all the MPI tasks running on a NUMA node using the numatcl utility. For the XT4 dual core system, we were seeing about 2.12 GBytes per second per core, and for the XT4 quad cores, we were seeing about 2.00 GB/sec per core. Table I shows Cielo's

memory / node	0	1	2	3
0	13.4	6.9	6.8	5.6
1	7.0	13.8	5.6	6.8
2	6.9	5.6	12.39	6.8
3	5.7	6.7	6.8	13.8

TABLE I
CIELO LOCAL AND REMOTE NUMA NODE BANDWIDTH, IN GB/SEC.

node level performance, measured using the STREAM TRIAD benchmark[4].

The Gemini interconnect is the most significant architectural difference between the XE6 and prior-generation Cray XT-series supercomputers, which use the SeaStar interconnect. Gemini and SeaStar are both custom system-on-a-chip ASICs developed by Cray that implement a high performance 3-D torus interconnect where each node is connected to its six nearest neighbors, as shown in Figure 2(b). Gemini achieves higher packaging density than SeaStar by supporting two physical nodes per Gemini chip, but logically each direction (X, Y, and Z) has the same number of network links. Every other hop in the Y dimension takes place within the Gemini ASIC.

Gemini has been architected to provide high performance support for fine grained remote load-store-style messaging, as is typical of partitioned global address space (PGAS) languages. This also results in significantly improved MPI messaging rates compared to SeaStar, as shown in the SMB message rate micro-benchmark [3] results shown in Figure 3(a). The Gemini achieves over an order of magnitude higher messaging rate than SeaStar for small messages. This translates to a significant performance boost for MPI applications that send many small messages in rapid succession.

Gemini also provides an evolutionary improvement to the achievable asymptotic bandwidth for point-to-point communication. As with SeaStar, there are two potential bottlenecks to consider: injection bandwidth and link bandwidth. Injection bandwidth is limited by the speed of the Opteron to Gemini HyperTransport link, which runs at 4.4 GT/s. Link bandwidth is determined by the signaling rate and the width of the link. Due to Gemini's double-density packaging, links in the X and Z dimensions are twice the width of links in the Y dimensions (24-bits vs. 12-bits wide). The uni-directional streaming bandwidth micro-benchmark results shown in Figure 3(b) illustrate this difference clearly. For the configuration tested, communication in the Y-dimension is limited by link bandwidth while communication in the Z-dimension is limited by injection bandwidth. SeaStar is always limited by injection bandwidth due to its slower 1.6 GT/s Opteron to SeaStar HyperTransport link.

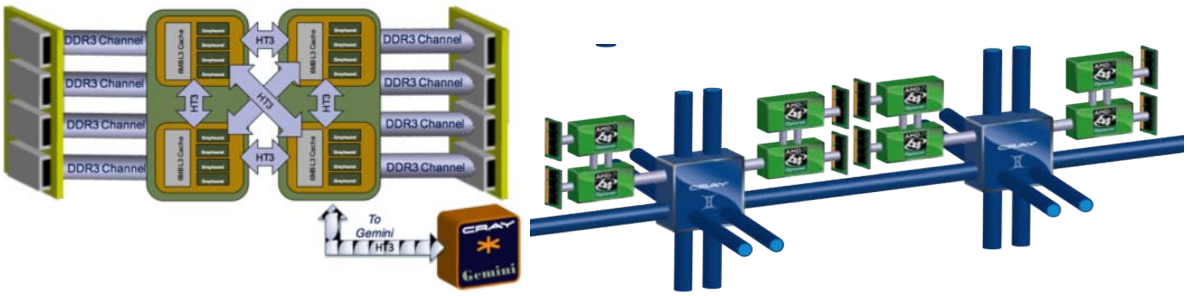
A comparison of Purple and Cielo specifications is shown in Table II. Its also interesting to note that Purple required 4.8 MWatts of power to run the system and 3 MWatts to run the cooling system. Cielo requires .

III. EXPERIMENTS

For this study we focus on three codes from the ASC Applications Acceptance Test (6x) suite, Charon, CTH, and AMG2006, described below. These codes exhibit distinct runtime profiles, facilitating understanding of the measured performance, by breaking it down into three components: the impact of the processor core, the impact of the node memory architecture, and the impact of the node interconnection network.

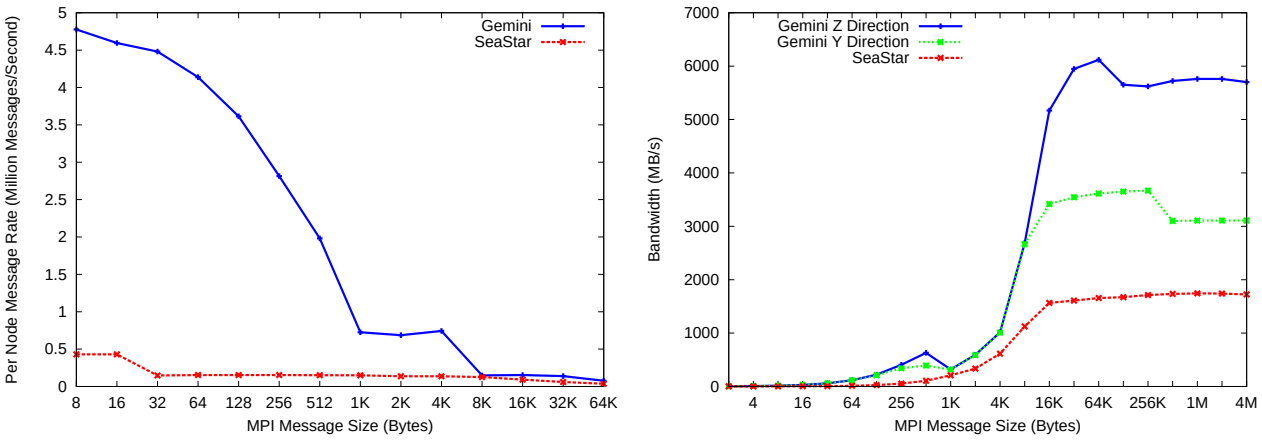
[3]Magny-Cours processors are also available with 12 cores divided into 6-core NUMA nodes, which form the basis of the new Hopper II computer at NERSC (http://www.nersc.gov/nusers/systems/hopper2/).

[4]www.cs.virginia.edu/stream



(a) Node (b) Interconnect

Fig. 2. The XE6 architecture. Images courtesy of Cray, Inc.



(a) Message injection rate (b) Point-to-Point Bandwidth

Fig. 3. Gemini vs. SeaStar

Arch	Node			Processor			Memory		Network			
	Number of nodes	*cores/ socket*	*sockets/ node*	*Number of cores*	*GHz*	*Ops / clock*	*DDR*	*GHz*	*Type*	*Latency µsec*	*Injection BW GB/sec*	*Peak Link BW GB/sec*
Purple	1,532	1	8	12,256	1.9	4	1	?	Federation	4.4	8	8
Cielo	6,654 8,944	8	2	107,264 143,104	2.4	4	3	1.33	Gem 1.2	1.3	6.1	9.4, 12-bit links 18.8, 24-bit links

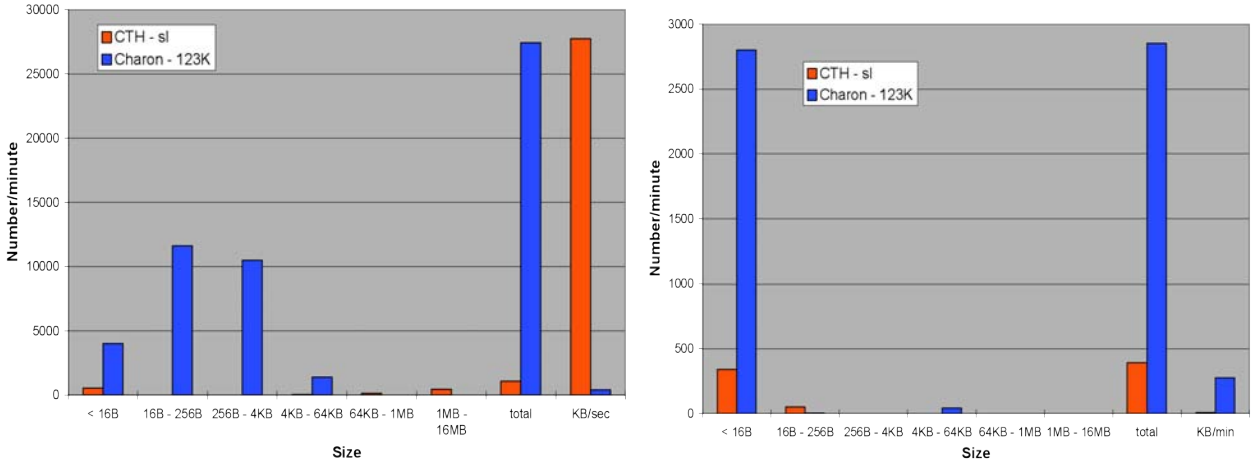
TABLE II
CIELO AND PURPLE COMPARISON

Each of the applications in the 6x suite measures performance based on an application-relevant "Figure of Merit" (FOM). The FOM are carefully chosen for each application to be representative of the performance characteristic of interest, and are intended to measure Cielo's ability to scale across tens of thousands of cores. That is, the goal is to capture the runtime characteristics of the application code rather than its algorithmic performance. For these codes, lower is better.

All experiments were run in weak scaling mode, in an MPI-everywhere configuration, whereby each MPI rank is assigned to a distinct processor core. Placement of the MPI processes onto the system is explicitly managed using executable launch command line options that enforce processor-memory affinity.

Point-to-point message traffic for these experiments (at 1,024 processor cores) is shown in Figure 4. This shows that Charon sends many messages relative to CTH, but the total volume of data transmitted is quite small relative to CTH. Some general runtime profiling information is shown in Table III, shown here using 8,000 processor cores. (Note that MPI time is exclusive of MPI_SYNC time.)

From a practical perspective, it's difficult to schedule all 1,336 of Purple's compute nodes. For this study 1,024 nodes (8,192 cores) were used to form the Purple baseline. In order to be fair, the number of Cielo compute nodes was limited to a similar fraction of the total number of compute nodes, no more than 5,138 nodes (82,208 cores). A comparison of key experimental settings for the two platforms is summarized in Table IV. Although the peak floating-point of Cielo is 12.7 times that of Purple, many of ASC's codes are memory subsystem bound and Cielo's peak memory bandwidth is only



(a) Point-to-point

(b) Collectives

Fig. 4. Message traffic for Charon (left) and CTH.

Performance Metric	*Purple*	*Cielo*	*Ratio*
Number of nodes	1,024 (of 1,336)	up to 5,138 (of 6,704)	5.02x
Number of cores	8,192	up to 82,208	10.0x
Peak FLOPS	62.3 TF	789 TF	12.7x
Peak memory BW	102 TB/s	438 TB/s	4.29x
Memory	32 TB	160 TB	5.0x
Memory per node	32 GB	32 GB	1.0x
Memory per core	4 GB	2 GB	0.5x

TABLE IV
PURPLE AND CIELO CONFIGURATIONS FOR THIS STUDY

4.3 times that of Purple. Memory capacity per node is the same between the two platforms, but Cielo has half the memory per core. This is a key metric for the current ASC code base, where 2 GB/core is considered a minimum ratio. Total memory capacity is five times that of Purple, allowing Cielo to accommodate the required larger problems.

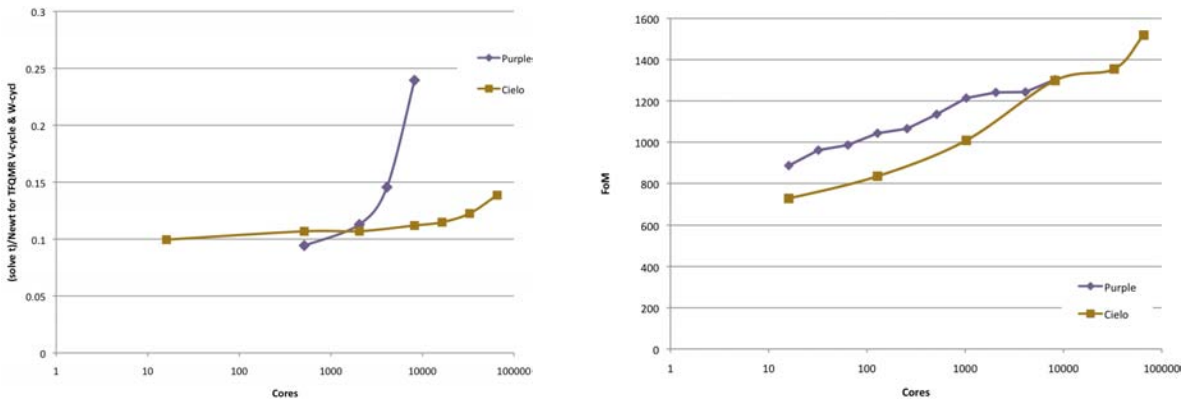
The performance of Charon and CTH on Purple and Cielo are shown in Figure 5, with lower representing better performance. Figure 6 illustrates the communication pattern for Charon and CTH on 32 and 128 cores, respectively. The processor in row i is sending to the processor in row j. Color represents the number of messages, with light green being the fewest and dark red being the largest. Figure 7 shows execution space-time diagrams for Charon and CTH, where the horizontal axis represents time, the vertical axis represents individual cores. These graphs, combined here for convenience, are discussed in the following sections.

Activity	*Charon*	*CTH*
Wall time (sec)	1,433.0	1,369.4
Iterations	7 outer, 52.4 avg inner	100
% Computation time	50.1	49.3%
% MPI time	15.9%	40.5%
% MPI_SYNC time	34.1%	
Number of collectives	68,098	9,000
MPI_All_reduce: $<$ 16B	66.8k	6,800
MPI_All_reduce:16-256B	143	
MPI_All_reduce:4k-64kB	222	1,000
MPI_Bcast: $<$ 16B	?	200
MPI_Bcast: 16 - 256B	?	200
MPI_Reduce_scatter (4KB)	562	–
MPI_All_Gather (7KB)	231	–

TABLE III
COMPUTATION AND COMMUNICATION PROFILE FOR CHARON AND CTH

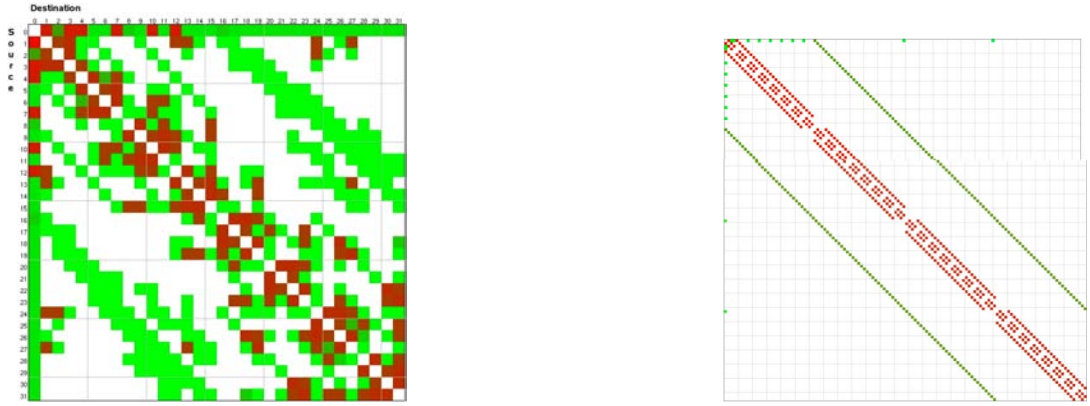
A. Charon

Charon is a semiconductor device simulation code[10] 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. An example 2D steady-state drift-diffusion solution is illustrated in Figure 8 for a bipolar junction transistor (BJT). Finite element discretization of these equations in space on an unstructured mesh produces a sparse, strongly coupled nonlinear system. A fully-coupled implicit



(a) Charon (b) CTH

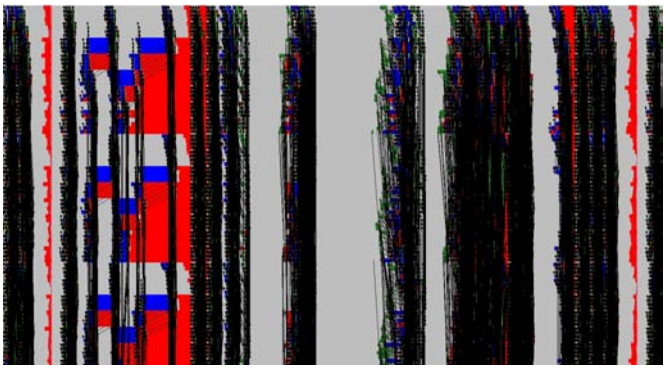
Fig. 5. Scaling Performance; lower is better



(a) Charon: 32 cores (b) CTH: 128 cores

Fig. 6. Communication patterns.

(a) Charon: 1 Newton iteration, 32 cores



(b) CTH: 1 time step, 128 cores

Fig. 7. Space-time profiles.

Newton-Krylov approach is used: the equations are linearized with Newton's method, and a Krylov solver (e.g. GMRES[11] or TFQMR[6]) is used for the solution of the sparse linear systems. A multigrid preconditioner is used to significantly improve scaling and performance [9]. The FOM is the time per linear solve iteration (after a Newton method is used to linearize the nonlinear system of equations).

For an example test case with about one million unknowns, or degrees of freedom (DOF), run on 32 cores, steady-state solution requires seven outer Newton iterations, each consisting of about 50 inner TFQMR iterations. Communication required by the multigrid preconditioner is complex. The smoothers on each level require communication with nearest neighboring subdomains. Projection operators between levels need to be produced, and the coarser levels are generated with a triple matrix product. The coarsest level requires a serial direct factorization.

The performance of Charon (weak scaling study with about 31,000 DOF/core) is shown in Figure 5(a). Figure 6(a) illustrates the communication pattern for 32 cores. Its runtime profile is shown in Figure 7(a). This problem completes in 1,433 seconds on Cielo, requiring 68,098 calls to MPI collective functionality. Of these 66.8K are small reductions (count = 1), 143 are medium size reductions (16 – 256 bytes), and 222 are large size reductions (4k – 64kbytes).

B. 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 to reduce dispersion and dissipation and produce accurate, efficient results. For these tests, we used the shaped charge problem, in three dimensions on a rectangular mesh, illustrated in Figure 9. The weak scaling configuration places a grid of size $(x, y, z) = (80, 120, 80)$ cells onto each parallel process. The Figure of Merit is time, so lower is better.

Computation is characterized by regular memory accesses, 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 three MBytes. Figure 6(b) illustrates the communication pattern for 128 cores. This clearly illustrates the nearest neighbor communication pattern. (Due to constraints in the tool graphics, this is actually two images pasted together.) The proportion of computation relative to communication was shown in Table III.

Each time step, CTH makes 90 calls to MPI collective functionality, 19 calls to exchange boundary data (2d "faces"), and three calls to propagate data across faces (in the x, y, and z directions). The message buffers are constructed from , 1/3 of which are contiguous, 1/3 of which are stride y, and one third of which are stride $x \times y$. Each boundary exchange aggregates data from 40 arrays, representing 40 variables. Collective communication is typically a reduction (`MPI_Allreduce`) of small counts. Figure 7(b) provides a space-time diagram.

At very large scale, as seen in Figure 5(b), CTH maintains its scaling profile, attributable to its "bursty" bandwidth requirements, whereby its very large messages are well managed by the node and interconnect architecture. The CTH message aggregation implementation would further benefit most strongly from increased interconnect bandwidth.

C. AMG2006

AMG2006 is a parallel algebraic multigrid solver of linear systems arising from problems on unstructured grids. Based on Hypre[5] library functionality, the benchmark, configured for weak scaling on a logical three dimensional processor grid $px \times py \times pz$, solves the Laplace equations on a global grid of dimension $px * 220 \times py * 220 \times pz * 220$. Typically $px = py = pz$, which maintains an optimal load balance. The figure of merit is defined as $SystemSize * NumIterations/SolvePhaseTime$. The solve phase time is the preconditioned CG solver time of 100 iterations. The baseline Purple performance was measured on 8000 processors.

Performance is shown in figure 10. At the node level and relatively small core counts (highlighted in Figure 10(a)), runtime is dominated by the memory bandwidth requirements

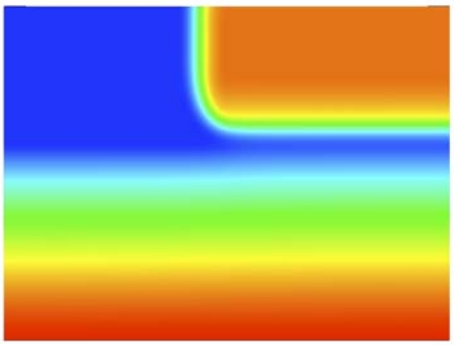


Fig. 8. Charon simulation

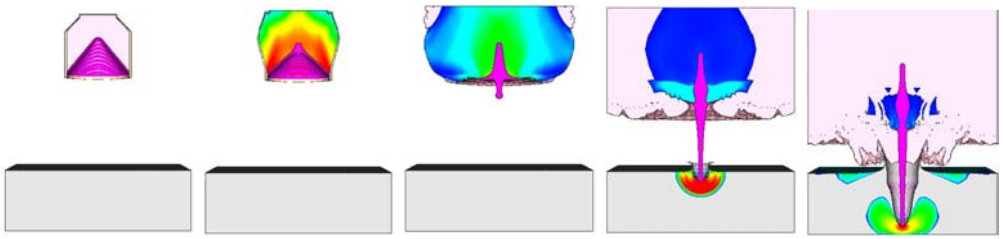
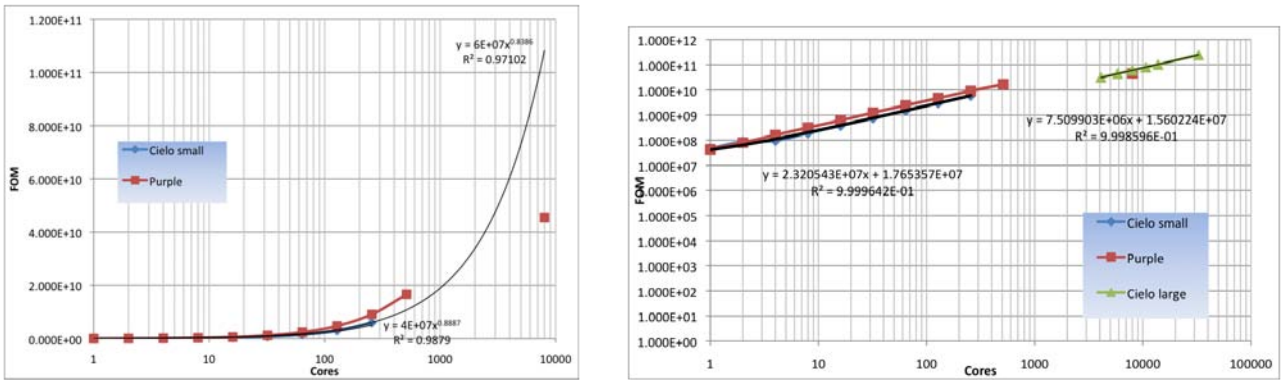


Fig. 9. CTH shaped charge simulation: time progresses left to right.



(a) Small core counts

(b) Large core counts

Fig. 10. AMG2006 performance

of the sparse matrix-vector product, a strength of Purple. However, at larger core counts (beginning at 8,192 cores, shown in Figure 10), this advantage disappears as runtime becomes dominated by inter-process communication, specifically `MPI_Allreduce`, 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.) The principal reason for this is the ability of the Cielo architecture to effectively manage the collective communication MPI_sync time, measured as the time between the first and last processor entering the function. The actual reduction functionality requires a relatively small amount of work. The sync time advantage in Cielo is primarily due to two reasons. First is the fast injection rate of messages. Second, the light-weight operating system on Cielo compute nodes minimizes the cumulative negative effect of OS interference as has been seen with Purple[8].

IV. Summary and Future Work

The ASC campaign's newest capability machine, named Cielo, met its application-driven acceptance criteria of a 6x improvement over its predecessor, ASC Purple. In order to better understand the reasons for this improvement, we studied the performance characteristics of the two machines by configuring experiments using two application codes that have been identified as critical to this computer's success. Some micro-benchmarks supplemented our understanding of the runtime characteristics of the new node and interconnect architecture.

Porting applications from Purple to Cielo has been straightforward. We find that the dual socket Magny-Cours NUMA node configuration, combined with the faster DDR-3 memory, and in combination with the Gemini interconnect, reversed the trend of multicore performance degradation, putting application performance above that of previous generations, despite the rather modest increase in processor clock speeds.

Gemini provides an evolutionary improvement to most of our applications, represented herein by CTH. More notably, Gemini's significantly increased message injection rate can provide significant performance improvements to codes that send relatively many smaller messages, as demonstrated here by Charon and AMG2006 (and in another context by xNOBEL[12]. One implication of this is the potential for an even greater impact on Partitioned Global Address Space (PGAS) languages, which we are also investigating in the context of important computations.

Although these applications are focused on problems of interest to the ASC campaign, in some important ways their implementations and runtime characteristics are representative of a much broader set of scientific computation codes. In particular, on-node bandwidth requirements, largely attributable to indirect memory addressing, the bulk-synchronous programming model, and message aggregation techniques are commonly found throughout many areas and implementations of scientific computation. The results described herein provide insight into the effects of the architectural characteristics employed by Cielo in order to address issues critical to large scale computers based on multi-core processors.

We will continue to investigate and report on the performance characteristics and capabilities of the Cielo architecture, focusing on the issues described above. Further, we look forward to comparing the effects of the Cielo node architecture with that of the Hopper II Cray XE6 recently installed at NERSC, which is based on 12-core Magny-Cours processors.

ACKNOWLEDGMENTS

The Cray Application Performance Analysis Group provided invaluable support, guidance, and advice throughout this work.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

[1] K. Alvin, R. Brightwell, and S. Dosanjh et al. On the path to exascale. *International Journal of Distributed Systems and Technologies*, 1(2), 2010.

[2] James Ang, Doug Doerfler, Sudip Dosanjh, Scott Hemmert, Ken Koch, John Morrison, , and Manuel Vigil. The Alliance for Computing at the Extreme Scale. In *Proceedings of the 52nd Cray User Group*, 2010.

[3] Brian W. Barrett and K. Scott Hemmert. An application based MPI message throughput benchmark. In *Proceedings of the 2009 IEEE International Conference on Cluster Computint (Cluster'09)*, 2009.

[4] D. Doerfler et al. Application-Driven Acceptance of Cielo, ASC's Petascale Capability Platform. Technical Report SAND tbd, Sandia National Laboratories, 2011. In preparation.

[5] Robert D. Falgout and Ulrike Meier Yang. hypre: a library of high performance preconditioners. In *Preconditioners, Lecture Notes in Computer Science*, pages 632–641, 2002.

[6] Roland W. Freund. A transpose-free quasi-minimum residual algorithm for non-Hermitian linear systems. *SIAM J. Sci. Comp.*, 14(2):470–482, 1993.

[7] E. S. Hertel, Jr., R. L. Bell, M. G. Elrick, A. V. Farnsworth, G. I. Kerley, J. M. McGlaun, S. V. Petney, S. A. Silling, P. A. Taylor, and L. Yarrington. CTH: A software family for multi-dimensional shock physics analysis. 1993.

[8] A. Hoisie, G. Johnson, D.J. Kerbyson, and M. Lang andf S. Pakin. A performance comparison through benchmarkings and modeling of three leading supercomputers: Bluegene/l, red storm, and purple. In *Proceedings of the IEEE/ACM Conference on Supercomputing SC'06*, November 2006.

[9] Paul T. Lin and John N. Shadid. Towards large-scale multi-socket, multicore parallel simulations: Performance of an MPI-only semiconductor device simulator. *Journal of Computational Physics*, 229(19), 2010.

[10] Paul T. Lin, John N. Shadid, Marzio Sala, Raymond S. Tuminaro, Gary L. Hennigan, and Robert J. Hoekstra. Performance of a parallel algebraic multilevel preconditioner for stabilized finite element semiconductor device modeling. *Journal of Computational Physics*, 228(17), 2009.

[11] Yousef Saad and Martin H. Schultz. GMRes: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[12] Courtenay Vaughan, Mahesh Rajan, Richard Barrett, Doug Doerfler, and Kevin Pedretti. Investigating the Impact of the Cielo Cray XE6 Architecture on Scientific Application Codes. Technical Report SAND 2010-8925C, Sandia National Laboratories, 2010. Under review by conference.

[13] C.T. Vaughan, M. Rajan, D. Doerfler, and R.F. Barrett. 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/IEEE conference on Supercomputing*, 2010.