SAND2011-0653C

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

Mahesh Rajan, Courtenay Vaughan, Doug Doerfler, Richard Barrett, Paul Lin, Kevin Pedretti and K. Scott Hemmert
Sandia National Laboratories
Albuquerque, NM, USA
Email: mrajan, ctvaugh, dwdoerf, rfbarre, ptlin, ktpedre, kshemme@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) weak scaling 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

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[2]. Recently we reported that Cielo is an improvement to its evolutionary predecessors from Cray[15], [16], 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 requirement of a six times improvement in application capability presented unique challenges as this passage between computer generations spanned the transition of architecture to the multi-core era. In particular, single processor compute power remained relatively stagnant, pressure upon on-node bandwidth accelerated, and node interconnect capabilities evolved to support these changes.

Our focus is on codes from the Cielo 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, though only hinted at in this report, we find that codes that send many small messages realize significantly stronger performance[15], [4], an issue critical to effective use of very high processor counts, a situation expected to be magnified at the exascale[1], [8].

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 compute nodes connected by the Federation High Performance Switch, with an aggregate peak performance of 81.2 TFLOPS.

The Purple compute node architecture consisted of eight IBM Power5-based Dual Chip Modules (DCM) that together operated as a single SMP system. Each DCM contained a Power5 processor chip coupled with a separate 36 MegaByte (MB) L3 cache chip. Among many innovative features, the Power5 processor was among the first dual-core processors, incorporated on-chip memory controllers, and included robust support for two-way symmetric multi-threading (SMT) with software assignable thread priorities [11]. In Purple, however, only one core per Power5 processor was enabled, leaving the full L3 cache and memory capacity available to the single active core on each chip. This choice was beneficial for HPC

[1] http://www.sandia.gov/nnsa/asc/

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

applications, which are typically memory bandwidth bound. SMT was enabled by default, but the second "virtual" core was used only for OS and system tasks, not MPI application processes. The resulting 16 OS-visible Power5 cores per compute node were managed by a single instance of IBM's AIX operating system.

The Purple network is built from IBMs proprietary Federation interconnect. The interconnect consists of switch network interfaces (SNI) on each node and High Performance Switches (HPS) connecting the nodes. Each Purple node has a single two port SNI, where each port is capable of 4 GigaBytes per second (GB/s) peak bi-directional bandwidth; when used together. The node can sustain about 3 GB/s of injection and 3 GB/s of ejection bandwidth, for a total of 6 GB/s of sustained bidirectional bandwidth. The high performance switches are connected in a fat tree topology with 3 levels of switches.

The hardware latency through the switch network is on the order of hundreds of nanoseconds. The total latency for an MPI message is about 4.5 microseconds, pointing to a large amount of software overhead for MPI messages. This high software overhead also points to relatively limited MPI message injection rate which negatively impacts the efficiency of sending small messages.

Cielo, an instantiation of a Cray XE6, is composed of AMD Opteron Magny-Cours processors (8-cores per processor), which are connected using the Cray Gemini interconnect and run a light-weight operating system called Compute Node Linux (CNL). The complete system, delivered toward the end of 2010, consists of 6,654 compute nodes, for a total of 106,464 processor core elements, capable of 1.02 PFLOPS peak performance. The final system, to be delivered in the spring of 2011 will consist of 8,894 compute nodes, for a total of 142,304 cores (1.37 PFLOPS peak). The high level system configuration is shown in Figure 1.

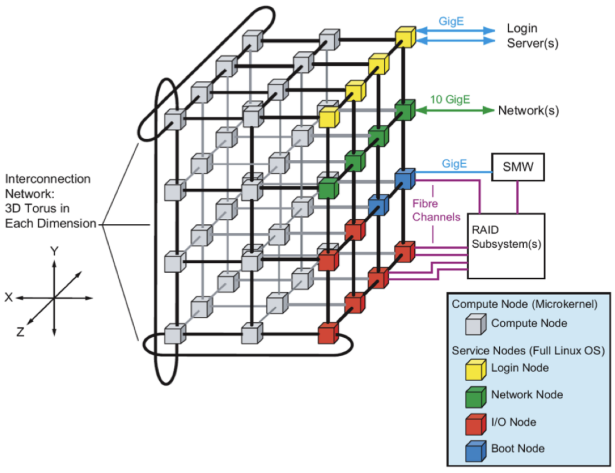


Fig. 1. Cielo XE6 architecture. Image courtesy of Cray Inc.

A Cielo node is created from two AMD Opteron 8-core Magny-Cours processors, each with 4 memory channels, and a Gemini interconnect chip as shown in Figure 2(a). Each Magny-Cours processor is divided into two memory regions, called 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 3. All links run at 6.4 Giga-Tranfers per second (GT/sec). So the 16-bit links between die in a processor run at 12.8 GigaBytes per second (GB/sec), the 16-bit links between processors also 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. Table I shows

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.

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

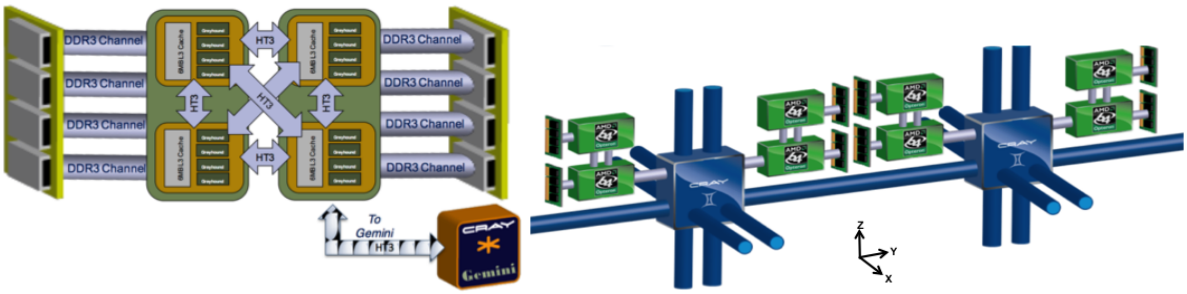
The Gemini interconnect is a custom system-on-a-chip ASIC developed by Cray that implements a high performance 3-D torus interconnect where each router is connected to its six nearest neighbors, as shown in Figure 2(b). Gemini achieves high packaging density by supporting two physical nodes per Gemini 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 the previous generation of Cray supercomputers (Cray XT with SeaStar interconnect), 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 Cray XT platforms 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. 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

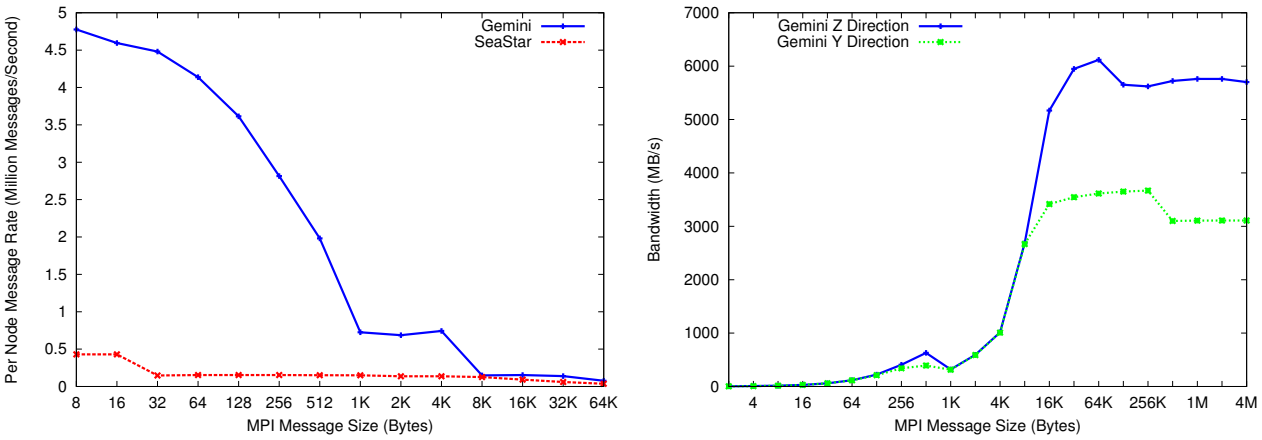
[3]Magny-Cours processors are also available with 12 cores divided into 6-core NUMA nodes, which form the basis of the new Hopper II computer at NERSC (http://www.nersc.gov/nusers/systems/hopper2/).

[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 Network Characteristics

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.

A comparison of Purple and Cielo specifications is shown in Table II.

	Purple	*Cielo*
# Nodes	1,532	6,654
Sockets/Node	8	2
Cores/Socket	1	8
Total Cores	12,256	106,464
Processor	IBM Power5	AMD Opteron
Frequency (GHz)	1.9	2.4
FLOPS/Clock	4	4
GFLOPS/Node	60.8	153.6
Memory Type	533 MHz DDR2	1333 MHz DDR3
Memory/Node (GB)	32	32
Mem BW/Node (GB/s)	128 (approx.)	85.3
NUMA Regions/Node	"SMP"	4
Network Interface	IBM Federation	Cray Gemini
Network Topology	Fat-Tree, 3-level	3-D Torus
PingPong Latency (μs)	4.4	1.3
Bi-dir Inj. BW/node (GB/s)	5	10
Bi-dir Link Bandwidth (GB/s)	8	9.4, 12-bit links 18.8, 24-bit links

TABLE II
CIELO AND PURPLE COMPARISON

III. EXPERIMENTS

For this study we focus on three codes from the ASC Applications Acceptance Test (6x) suite: AMG2006, Charon, and CTH, individually 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.

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. Defined in the following sections, 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[5].

Basic inter-process communication traffic is illustrated in Figure 4 (at 1,024 processor cores). Point-to-point communication (Figure 4(a)) shows that Charon sends many messages relative to CTH, but the total volume of data transmitted is quite small relative to CTH. AMG2006 does not require much of this sort of communication. Collective communication is a critical issue for AMG2006 and Charon (Figure 4(b)). Some general runtime profiling information for Charon and CTH is shown in Table III, shown here using 8,192 processor cores. (Note that MPI time is exclusive of MPI_SYNC time.)

Activity	*Charon*	*CTH*
Wall time (sec)	1,433.0	1,369.4
Iterations	7 out (52.4 in)	100
% Computation time	50.1	49.3
% MPI time	15.9	40.5
% MPI_SYNC time	34.1	11.0
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

Relevant AMG2006 data is discussed below.

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 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.

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

The FOM performance of AMG2006, Charon, and CTH on Purple and Cielo are shown in Figure 5, with lower representing better performance. Figure 6 illustrates the communication patterns for the applications. 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 the application execution space-time diagrams, where the horizontal axis represents time, the vertical axis represents individual cores. Black represents point-to-point communication, gray is computation, green is send, blue is receive, red is synchronization, and pink is reduce. These graphs, combined here for convenience, are referenced in the following sections.

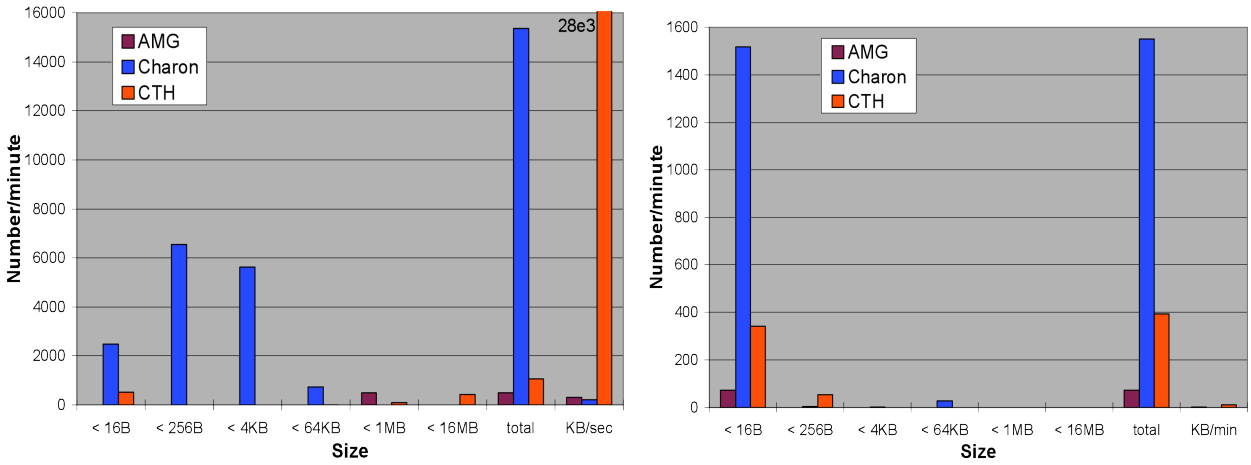
A. AMG2006

AMG2006 is a parallel algebraic multigrid solver of linear systems arising from problems on unstructured grids. Based on Hypre[6] 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 $px * 220 \times py * 220 \times pz * 220$. The Figure of Merit measures the solve phase time for the preconditioned conjugate gradient solver for 100 iterations.

Performance was shown above in figure 5(a), which includes the baseline Purple performance on 8,000 processors. At the node level and relatively small core counts runtime is dominated by the memory bandwidth requirements of the sparse matrix-vector product, a strength of Purple. However, at larger core counts, 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 synchronization 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[10].

B. Charon

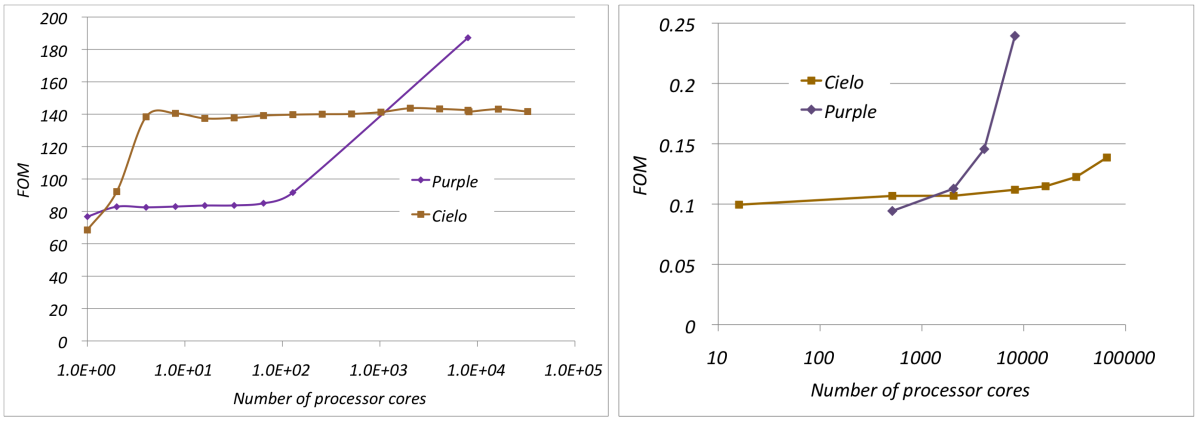
Charon is a semiconductor device simulation code[13] designed for use on high performance parallel computers using the MPI-everywhere model. The drift-diffusion model is used, which is a coupled system of nonlinear partial differential equations that relate the electric potential to the electron



(a) Point-to-point

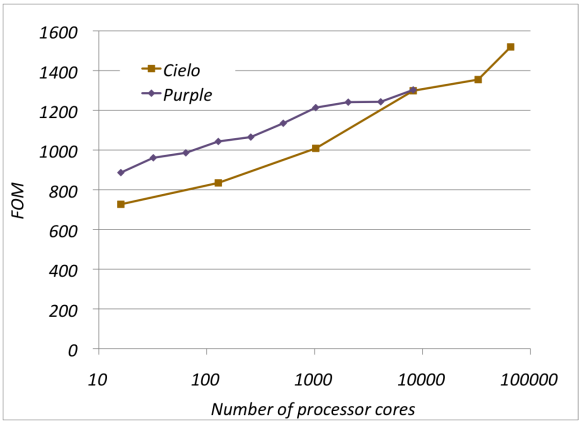
(b) Collectives

Fig. 4. Message frequency and volume



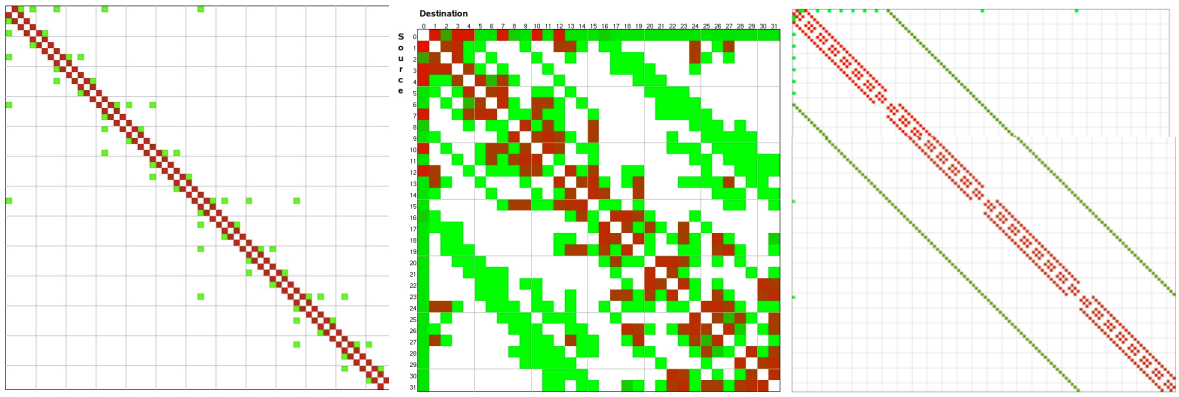
(a) AMG2006

(b) Charon



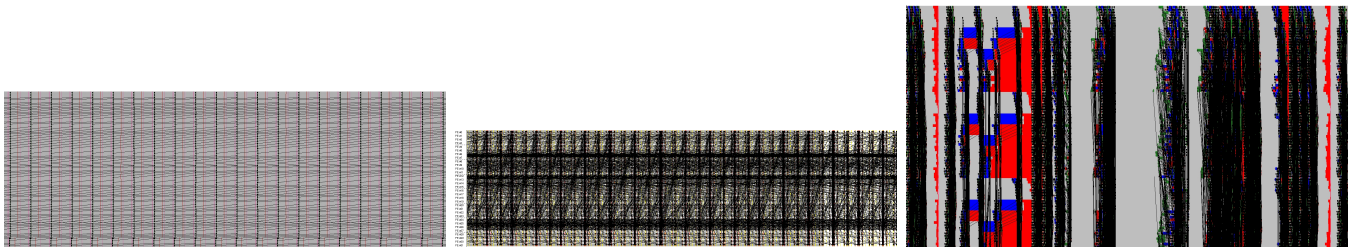
(c) CTH

Fig. 5. Scaling performance of FOM; lower is better



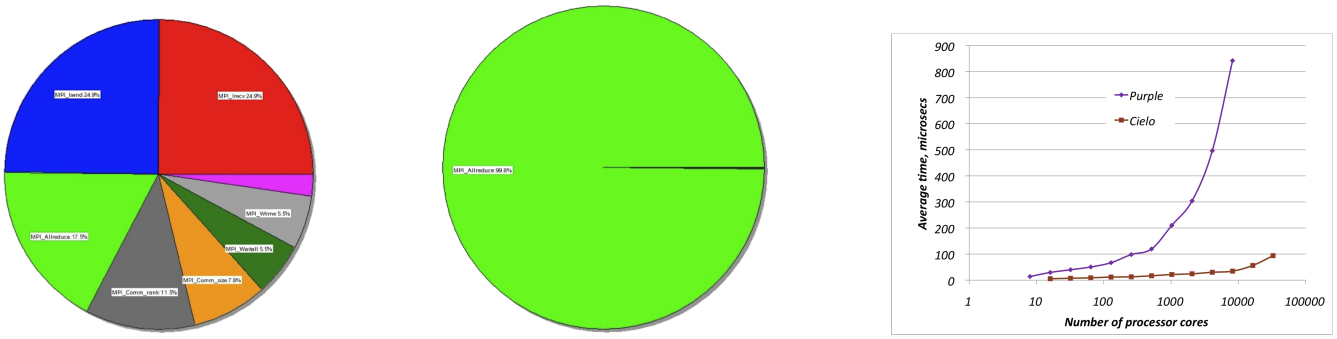
(a) AMG2006: 64 cores (b) Charon: 32 cores (c) CTH: 128 cores

Fig. 6. Point-to-point communication patterns.



(a) AMG: multiple time steps, 64 cores (b) Charon: 1 Newton iteration, 32 cores (c) CTH: 1 time step, 128 cores

Fig. 7. Runtime trace profiles.



(a) Number of MPI calls (b) Time in MPI (c) MPI_Allreduce performance

Fig. 8. AMG MPI information

and hole concentrations. An example 2D steady-state drift-diffusion solution is illustrated in Figure 9 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 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 [7] is used to significantly improve scaling and performance [12]. The FOM is the time per linear solve iteration (after a Newton's method is applied 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 linear iterations. Communication required by the multigrid preconditioner is complex. The smoothers on each level require communication with nearest neighboring subdomains. 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

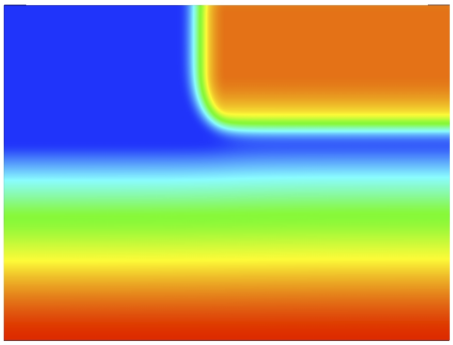


Fig. 9. Charon steady-state solution

31,000 DOF/core) is shown in Figure 5(b). Figure 6(b) illustrates the communication pattern for 32 cores. Its runtime profile is shown in Figure 7(b). The 8,192-core problem completes in 1,433 seconds on Cielo, requiring 68,098 calls to MPI collective functionality. Of these 66,800 are small reductions (count = 1), 143 are medium size reductions (16 – 256 bytes), and 222 are large size reductions (4k – 64k bytes).

C. CTH

CTH is a multi-material, large deformation, strong shock wave, solid mechanics code developed at Sandia National Laboratories[9]. 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 10. 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 essentially the 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 three MBytes. Figure 6(c) shows the communication pattern for 128 cores, illustrating 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 is similar to that of Charon (shown in Table III). However, the space-time profile (Figure 7(c)) illustrates the distinction: CTH's very large message aggregation scheme in the bulk synchronous programming (BSP) model[14] induces a strong separation of computation and communication, showing a "bursty" point-to-point communication pattern.

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). Collective communication is typically a reduction (`MPI_Allreduce`) of small counts. Each boundary exchange aggregates data from 40 three dimensional arrays, representing 40 variables. Message buffers are constructed from faces, approximately one third of which are contiguous, one third of which are stride y, and one third of which are stride $x \times y$.

At very large scale, as seen in Figure 5(c), 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.

IV. Summary and Future Work

The ASC campaign's newest capability machine, named Cielo, met its application-driven acceptance criteria of a six times 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 three 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. Its worth noting that our analysis was hindered in some ways by the retirement of Purple, which eliminated our ability to clearly determine root causes of some performance issues that were brought up by our experiences with Cielo. However, we are confident that we have captured the relevant issues that have enabled Cielo to support the target applications at current capability scales.

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, including those examined herein. However, Gemini's significantly increased message injection rate can provide significant performance improvements to codes that send relatively many smaller messages, as hinted at here by Charon and AMG2006 (and in another context by xNOBEL[15]). One implication of this is the potential for an even greater impact on Partitioned Global AddressSpace (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. CTH is an explicit Eulerian code operating on a three dimensional structured grid. Charon is an implicit Eulerian code, operating

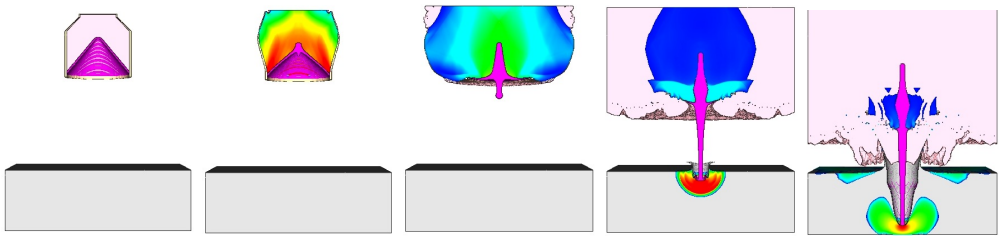


Fig. 10. CTH shaped charge simulation: time progresses left to right.

on a two dimensional unstructured grid, based on a Newton-Krylov solution approach (requiring solution of sparse linear systems) with a nonsymmetric iterative solver and a multigrid preconditioner. Together, they expose a breadth of on-node bandwidth requirements, the bulk-synchronous programming model, and message aggregation techniques 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. Scott Pakin and Mike Lang from Los Alamos National Laboratory provided additional insight and data on the Purple platform.

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 Energy's 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] Douglas Doerfler, Courtenay Vaughan, Paul Lin, Mahesh Rajan, and Richard Barrett. Application-driven Analysis of the Performance Characteristics of Capability and Capacity Computing. Technical Report SAND 2011-tbd, Sandia National Laboratories, 2011. In progress.

[5] Douglas W. Doerfler. Evaluation of the Impact Chip Multiprocessors have on SNL Application Performance. Technical Report SAND2009-6496, Sandia National Laboratories, October 2009.

[6] 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.

[7] Michael W. Gee, Chris M. Siefert, Jonathan J. Hu, Ray S. Tuminaro, and Marzio G. Sala. ML 5.0 Smoothed Aggregation User's Guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[8] A. Geist and S. Dosanjh. IESP Exascale Challenge: Co-Design of Architectures and Algorithms. *International Journal of High Performance Computing*, 2009.

[9] 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.

[10] 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.

[11] Ron Kalla, Balaram Sinharoy, and Joel M. Tendler. IBM Power5 Chip: a Dual-Core Multithreaded Processor. *IEEE Micro*, 24(2):40–47, Mar-Apr 2004.

[12] 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.

[13] 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.

[14] Leslie G. Valiant. A Bridging Model for Parallel Computation. *Commun. ACM*, 33:103–111, August 1990.

[15] 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.

[16] 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.