SAND2011-0896P

Capability vs. Capacity; HPC systems application performance comparisons: Cielo vs. Red Sky

Doug Doerfler and Mahesh Rajan

Sandia National Laboratories

2/9/2011

SAND 2011-XXXX
Unlimited Release
Printed February, 2011

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04- 94AL85000.

NNSA National Nuclear Security Administration ASC

Sandia National Laboratories

Acknowledgements

- Sophia Corwell, Steve Monk and the Cap-Viz team for all the support during Red Sky dedicated time.
- Marcus Epperson for Red Sky large core count run support.
- Paul Lin for all his support with Charon on Cielo and Red Sky.
- Brian Barrett for his OpenMPI expertise.

Objective

- Share findings of recent Red Sky dedicated time application runs
- Provide ASC management with comparative performance between capacity and capability systems
- Investigations in support of planned Supercomputing 2011 paper submission
- Findings helps with application support on SNL HPC systems with deeper understanding of applications and architectures

Presentation Outline

- Quick overview of Cielo and Red Sky architecture
- Micro Benchmark results
- Application performance results
  - Cielo 6x Tri-Lab system acceptance suite
  - Other applications of interest to SNL (if time permits)

Executive Summary

- Current trends of increasing core counts per compute node has an impact on system balance by increasing computation to communication balance ratios
  - Red Sky results are using TWICE AS MANY nodes as Cielo, 8 PPN versus 16 PPN respectively
  - Red Sky performance at small scale is better in many cases; This is attributed to: excellent Intel core architecture, higher clock rate and better memory bandwidth per core
- Red Sky scaling for large number of cores suffers from poor MPI global operations (sync time) and consequently we observed better scaling properties on Cielo
- Cielo, as per design goals for a capability system, scales well at large core counts; light weight OS and better MPI global ops performance contribute to scaling
- Red Sky QDR IB has good latency and bandwidth, but suffers from OpenMPI scalability issues (needed BTL MEMOPTS to successfully run at 8k, 16k for four of the six Cielo 6x apps)
- Run time variation on Red Sky can be substantial
- Red Sky numa_wrapper as has been previously reported as essential for applications to minimize run time variation and reduce solution time. In Cielo affinity control impact is not that dramatic, but easily handled through aprun options
- Further investigations on MVAPICH and OpenMPI (btl) options should be studied for performance tuning and feasibility of running IB at larger scales
- Red Sky allocation policies are not conducive to “mid-range” computing, despite this being one of its design goals
  - Job submissions subsequent to dedicated time for small to medium jobs sizes (64 to 256 nodes) has resulted in LONG (days to a week) to MAY NEVER RUN? (> 2 weeks waiting) turn around times.
  - This should be addressed to enable and encourage higher capability analysis

System Comparison

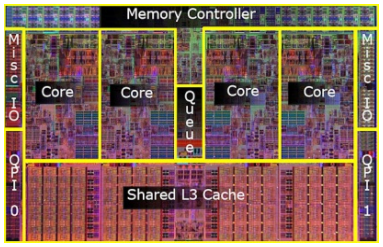
SYSTEM	Red Sky	Cielo
Num Compute Nodes	2318	6704
Num Compute Cores	18,544	107,264
Processor	Dual Intel Nehalem 2.93 GHz	Dual AMD Magny-Cours, 2.4 GHz
Cores / node	8	16
Memory / Core	1.5 GB	2 GB
Peak Node GFLOPS	93.76	153.6
Memory	3 channels/socket, DDR3, 1333 MHz	4 channels/socket, DDR3,1333 MHz
Cache	L1=4x32KB I,D L2=4x512KB L3=8MB	L1=8x64 KB, I,D L2=8x512KB L3=12MB (10MB)
Interconnect / Topology	QDR IB, Torus	Gemini, Torus
Compute Node OS	TOSS	CNL
MPI	OpenMPI 1.4.1	MPT 5.1.4
Compilers	Intel 11.1	PGI 10.8 and 10.9

Red Sky Software

- OS: CentOS Red Hat Linux based with patches
- MPI: OpenMPI & MVAPICH (IB OFED stack)
- Scheduler: Slurm and Moab
- Compilers: Intel and GNU
- Debugger: TotalView
- Math Libraries: BLACS, FFTW, MKL
- Performance Tools: OpenSpeedShop, TAU, mpiP
- User Control: via Modules

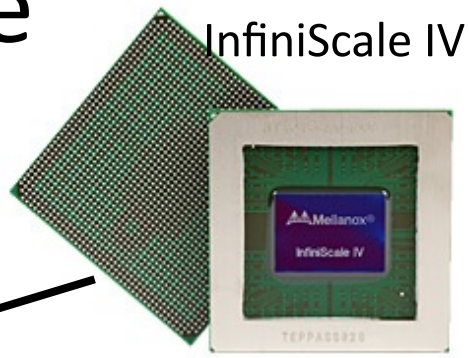
Node & Chassis Architecture

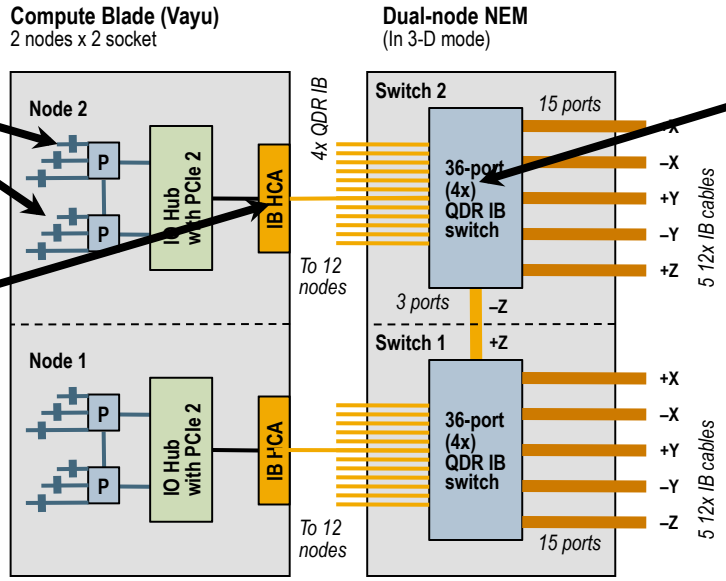
Intel Nehalem

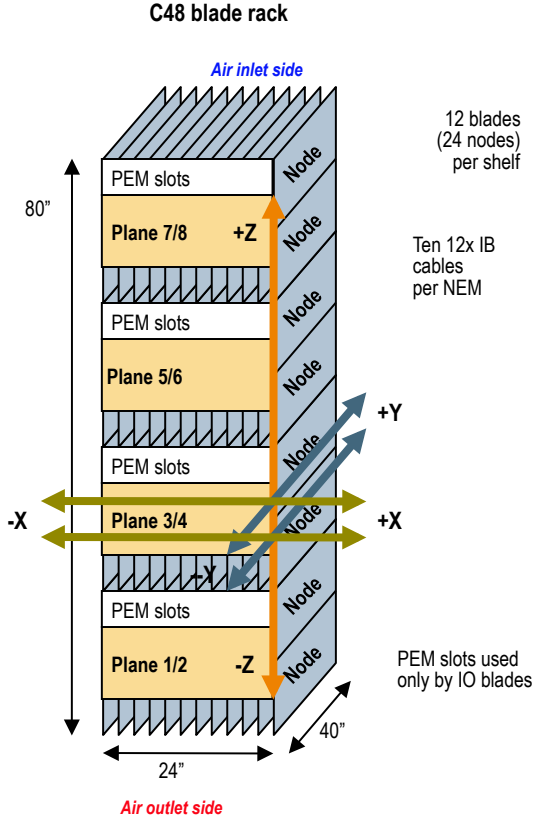


Mellanox ConnectX

Mellanox InfiniScale IV







Sun microsystems

National Nuclear Security Administration ASC

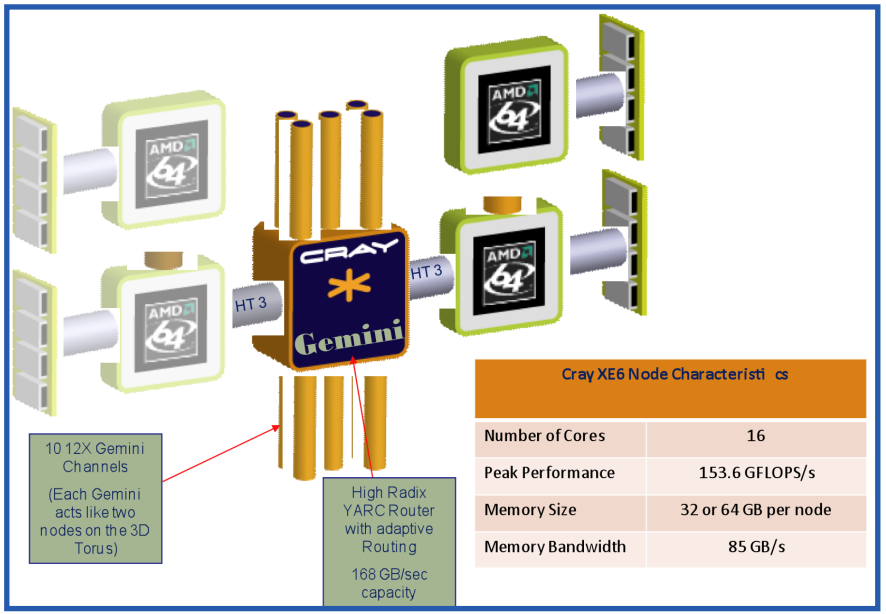
Sandia National Laboratories

High-Level Software Architecture: Cray's CLE

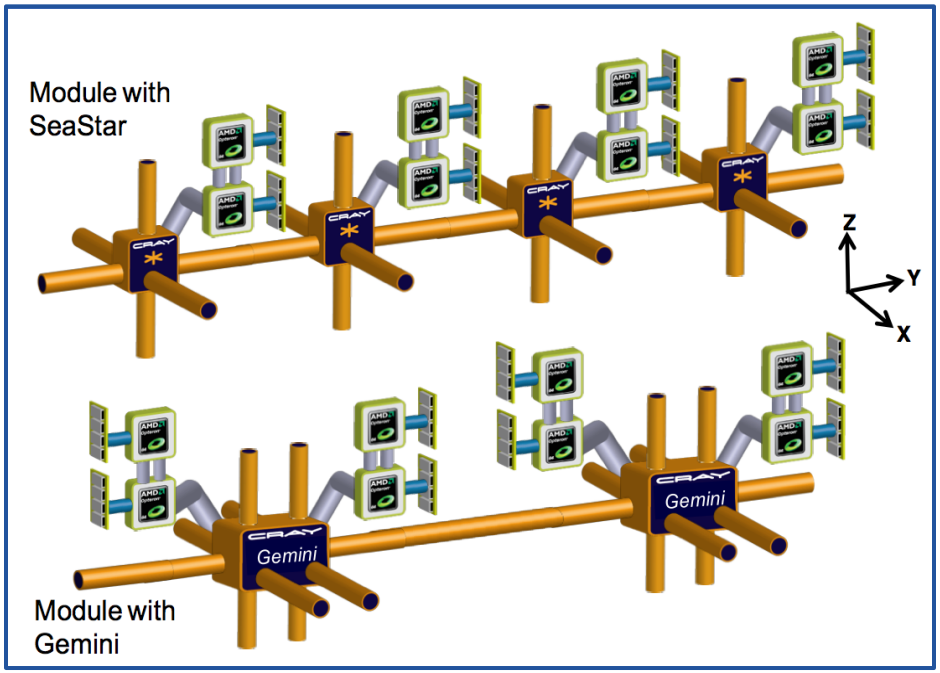
Feature	LWOS	FFOS
Pedigree	Linux derivative	Linux derivative
Personality	Streamlined for good application performance; configurable job-by-job	Full featured
Target functionality	Compute	Compute & Service
Language Support	Fortran, C, C++, Python, Perl, Java, Shells	Fortran, C, C++, Python, Perl, Java, Shells
Programming Models	MPI-2 within LWOS, OpenMP, POSIX Threads	SLES support for MPI, OpenMP, POSIX Threads
High-speed Interconnect protocols	Native high-speed for MPI-2	Native high-speed for MPI-2, Sockets, NFS & TCP/IP
Supported Libraries	Cray optimized scientific libraries, libm, libgsl, FFTW, BLAS1-3, LAPACK, dynamic libs and dlopen()	Standard libraries as part of SLES distribution
Application tools	CrayPat and Apprentice2 w/support for hardware counters, memory usage, performance profilers, MPI tracing and profilers	Hardware counters, memory usage, performance profilers, MPI tracing and profilers
Application Debugger	Moab up to 8192 MPI ranks	
Data Analysis & Geometry Extraction	Support for VisIt, ParaView and EnSight	
Other	Support for MOAB, scalable job launch	Support for MOAB, scalable job launch

Cielo Hardware Architecture

- **Topology**
  - Gemini High-Speed Interconnect
  - Phase 1: 18x8(16)x24 3D Torus
  - Phase 2: 24x8(16)x24 3D Torus
- **Node: dual-socket, AMD Magny-Cours**
  - 16 total cores (8 per socket)
  - 2.4 GHz core clock rate
  - 8 channels (4 per socket) 1333 MHz DDR3 memory
  - 4 FLOPS per clock per core
  - 32 GB total memory
  - 153.6 GF peak (double-precision)
  - 85.3 GB/s peak memory BW



	Phase 1	Phase 2
Total # of racks	72	96
Total # of compute blades	1,676	2,236
# of compute nodes	6,704	8,944
# of cores	107,264	143,104



National Nuclear Security Administration ASC

Key parameters that impact performance

- Memory Bandwidth
- Clock speed
- Memory hierarchy and size
- NUMA impact
- MPI: Ping-Pong Bandwidth & latency
- MPI: Message Injection Rate
- MPI: Global Ops
- OS Noise
- Others that we have not investigated: compiler, page size, topology (task placement), MPI parameters

Memory Bandwidth – Streams Triad

	mem 0	mem 1
Numa node 0	15400	8811
Numa node 1	8757	15546

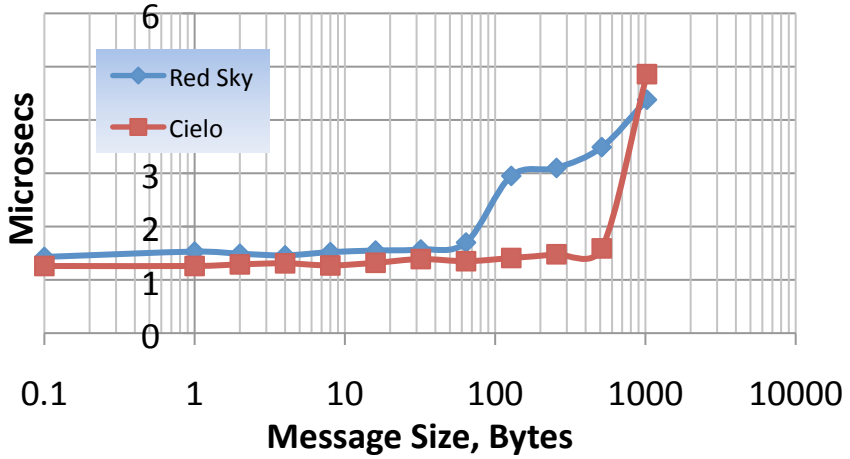
Red Sky Streams Triad results showing NUMA effects;
Total node BW = 2*15.5MB/s ~**33GB/s**

	mem 0	mem 1	mem 2	mem 3
Numa node 0	13434.939	6877.3351	6770.985	5641.965
Numa node 1	7003.2609	13809.1659	5643.637	6819.851
Numa node 2	6864.1749	5593.0695	13866.23	6839.503
Numa node 3	5673.7927	6707.4935	6831.497	13795.28

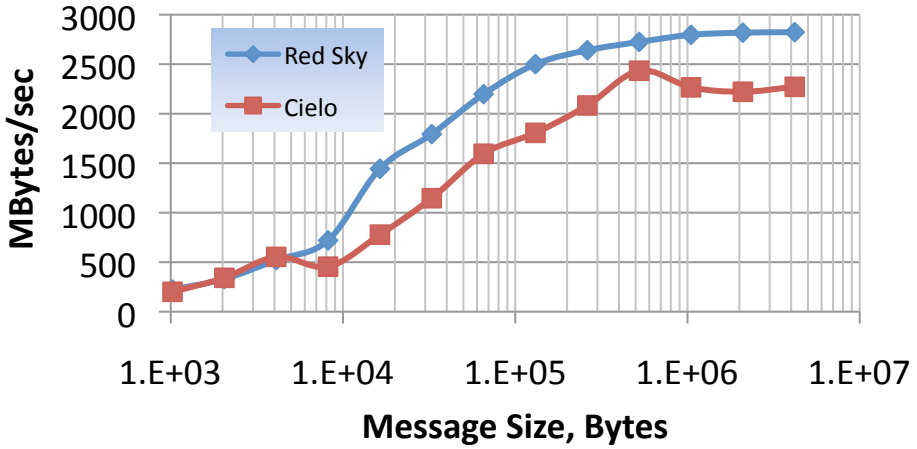
Cielo Streams Triad results showing NUMA effects;
Total node BW = 4*13.5MB/s =**54GB/s**

MPI Messaging: Ping-Pong, SendRecv Latency & Bandwidth

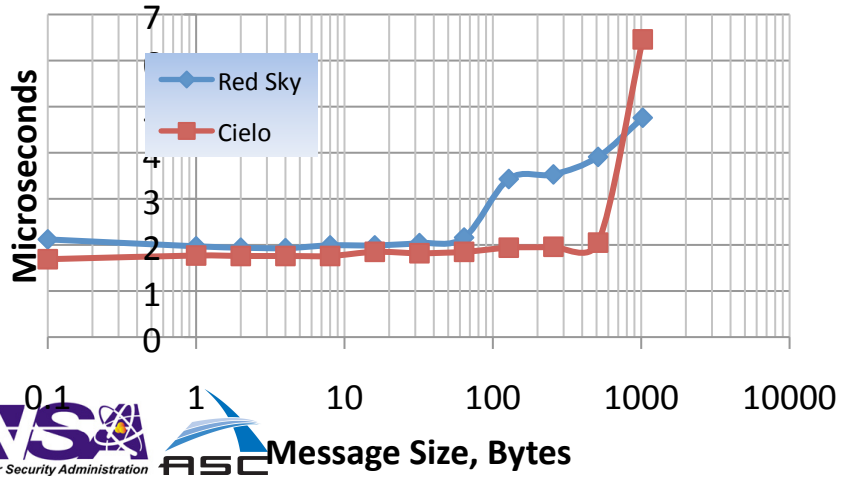
IMB Benchmark: PingPong 2 nodes; 1 task/node; Latency



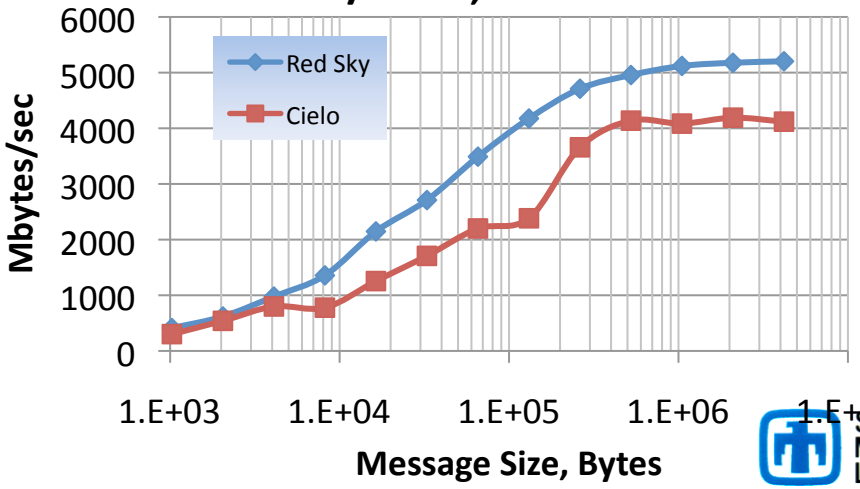
IMB Benchmark: PingPong 2 nodes; 1 task/node; Bandwidth



IMB Benchmark: SendRecv 2 nodes; 1 task/node; Latency



IMB Benchmark: SendRecv 2 nodes; 1 task/node; Bandwidth



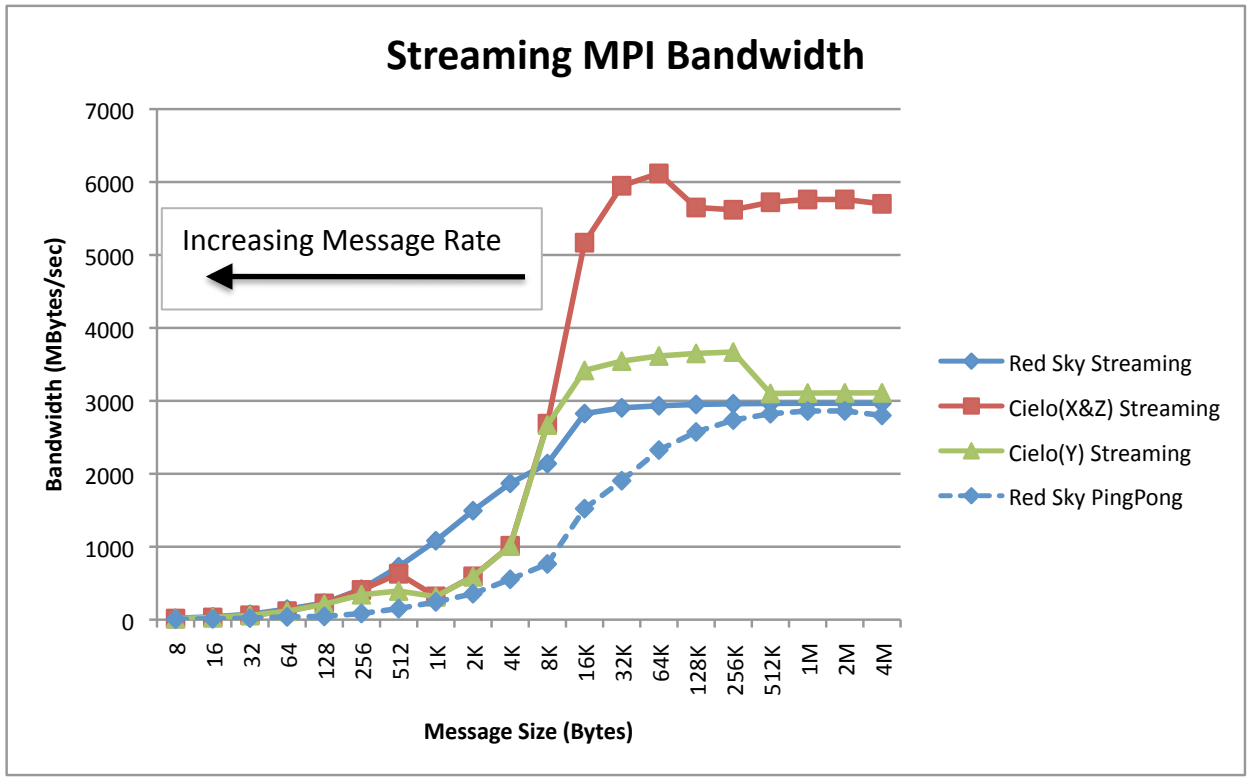
National Nuclear Security Administration

ASC

Sandia National Laboratories

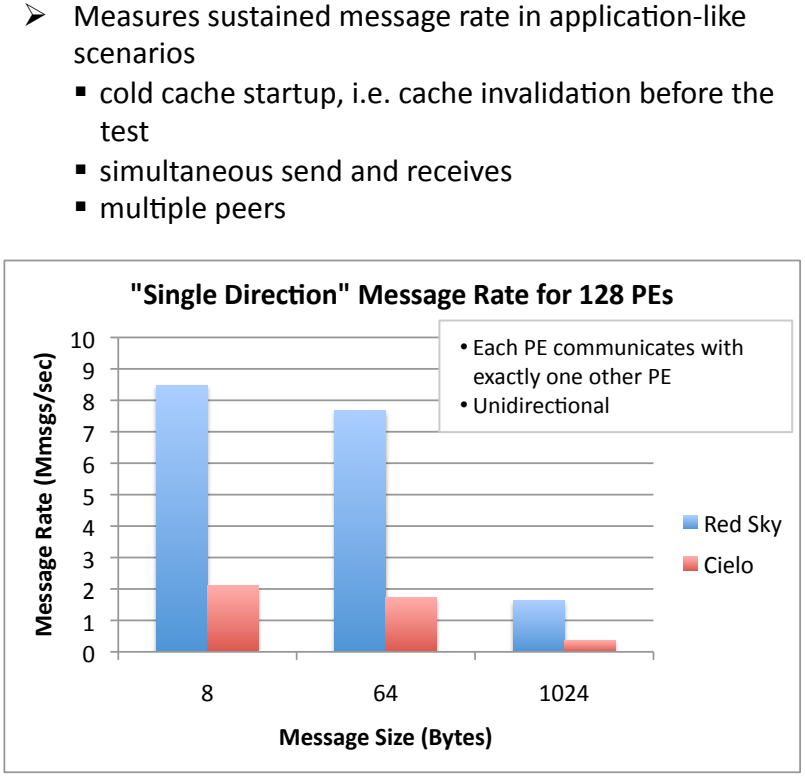
MPI Messaging Rate: Streaming Ping-Pong

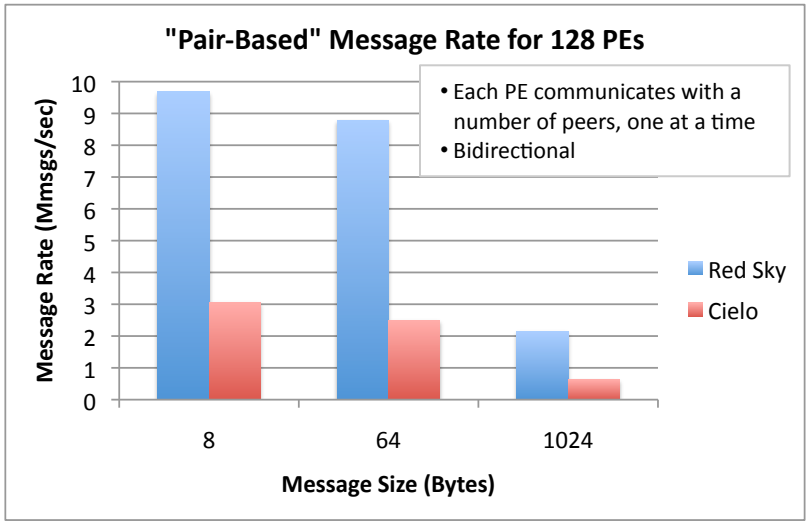
- Streaming bandwidth using Sandia mpi_bw microbenchmark
- Measures bandwidth by posting multiple outstanding sends/receives, as opposed to ping-pong with is a single send/receive
- The earlier and steeper the rise the more capable the network interface is of processing multiple outstanding message transfers
- Mpi_bw is similar to SMB Msg_rate Pre-Post metric but uses more ideal conditions
- However, it is included as it is instructive to view with the traditional Ping-Pong BW curves

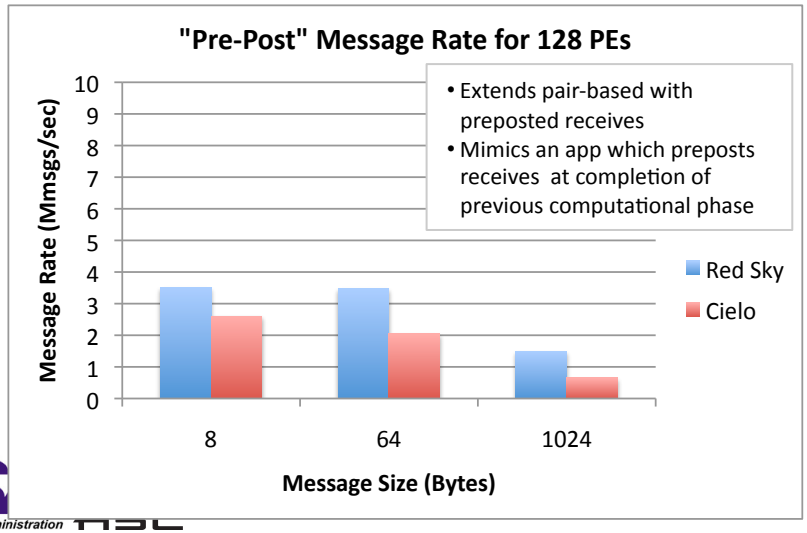


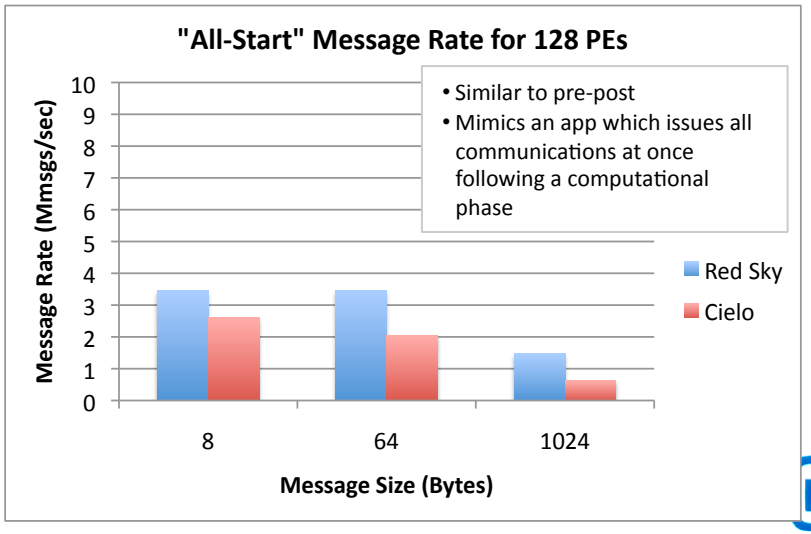
MPI Message Rate: SMB msg_rate

- Measures sustained message rate in application-like scenarios
  - cold cache startup, i.e. cache invalidation before the test
  - simultaneous send and receives
  - multiple peers
- Red Sky: 16 nodes, 8 PEs/node
- Cielo: 8 nodes, 16 PEs/node
- Measured result is aggregate message rate per node
  - All cores on a node with varying message size

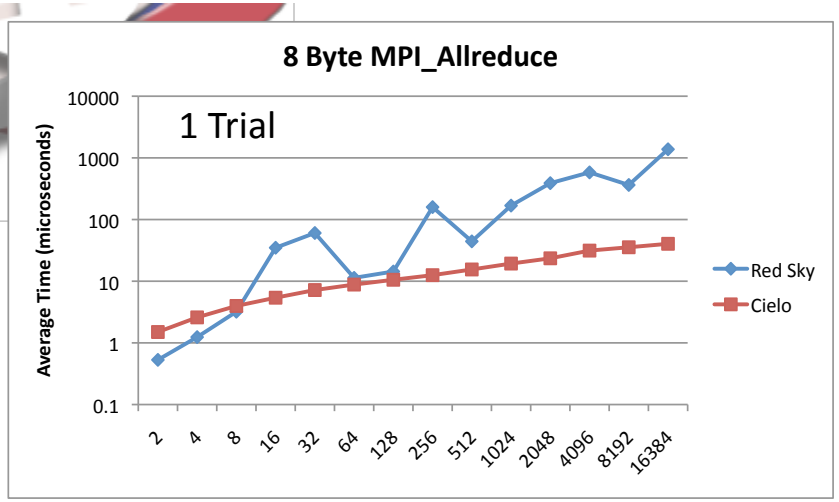


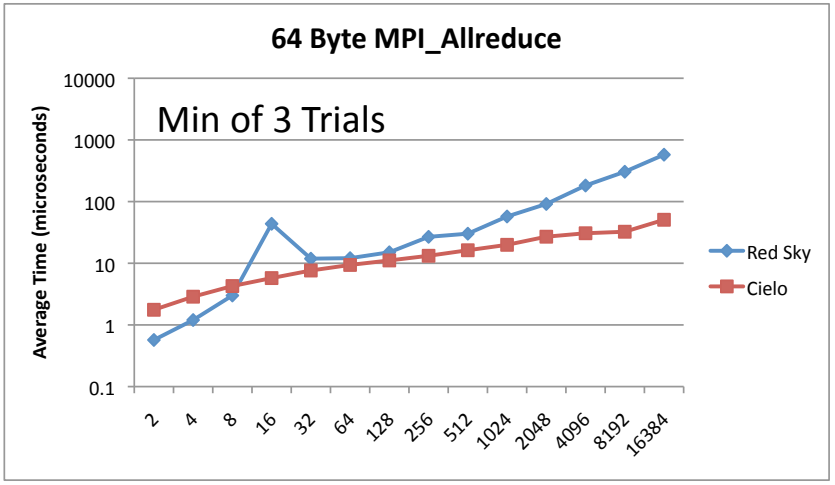


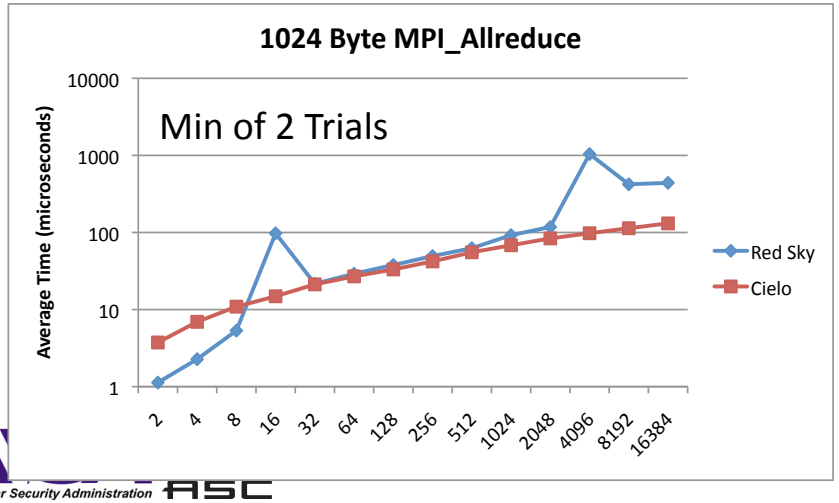


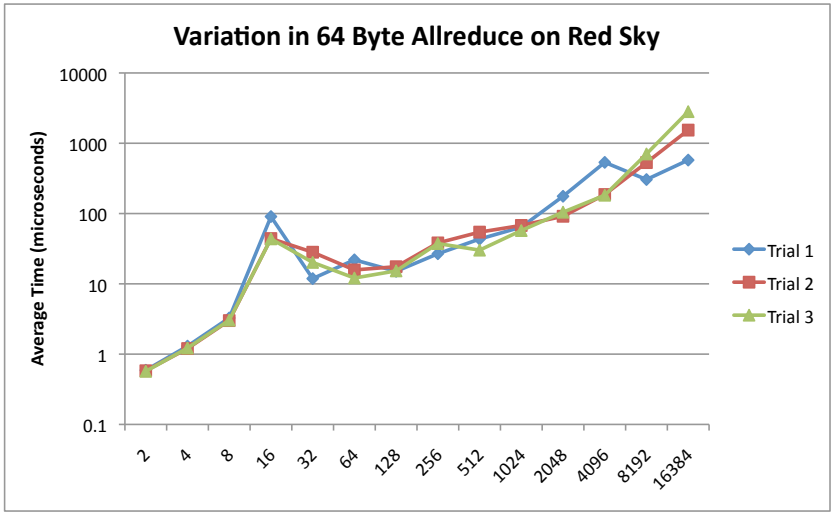


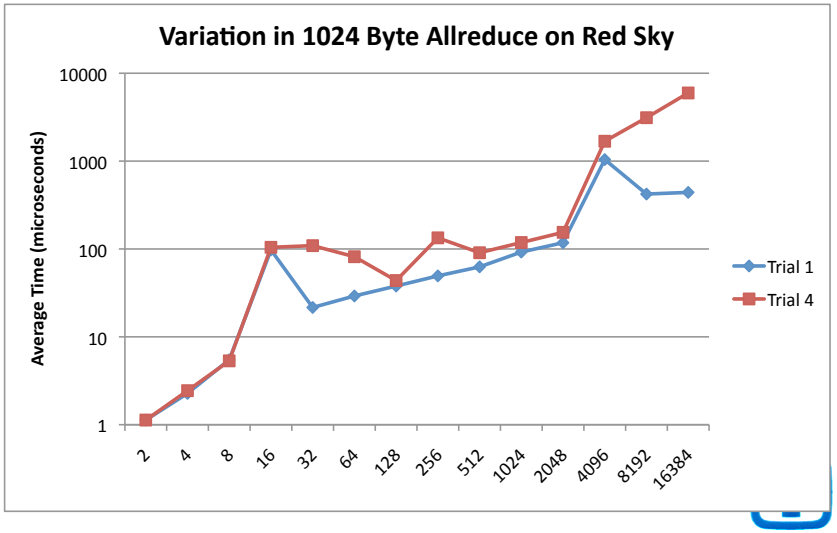
MPI Collectives: Allreduce (lower better)







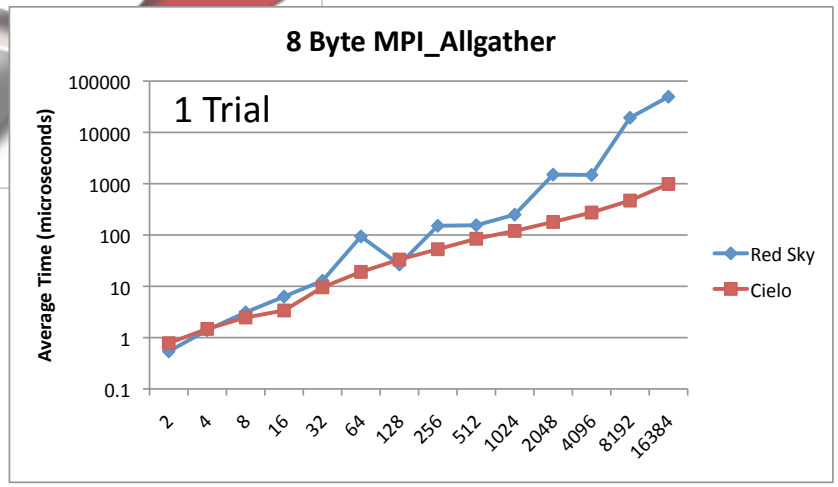


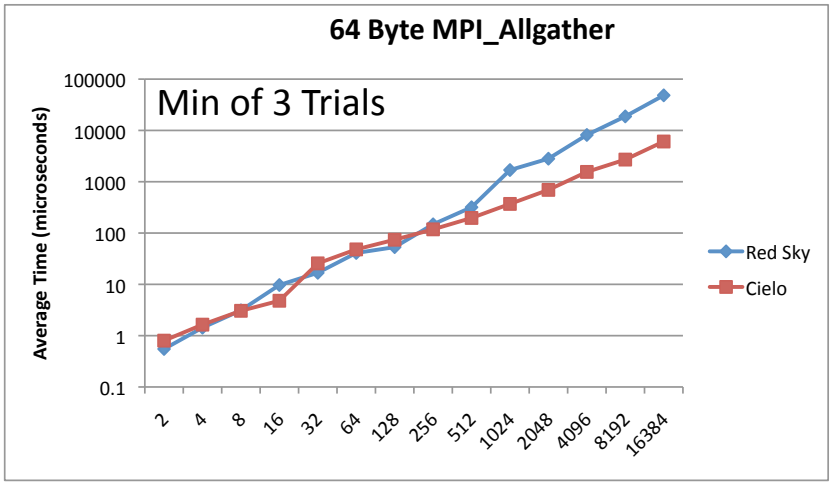


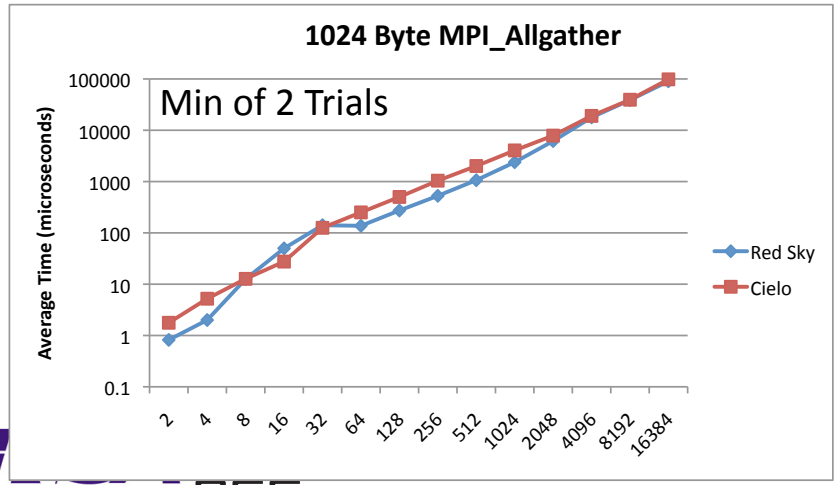
National Nuclear Security Administration ASC

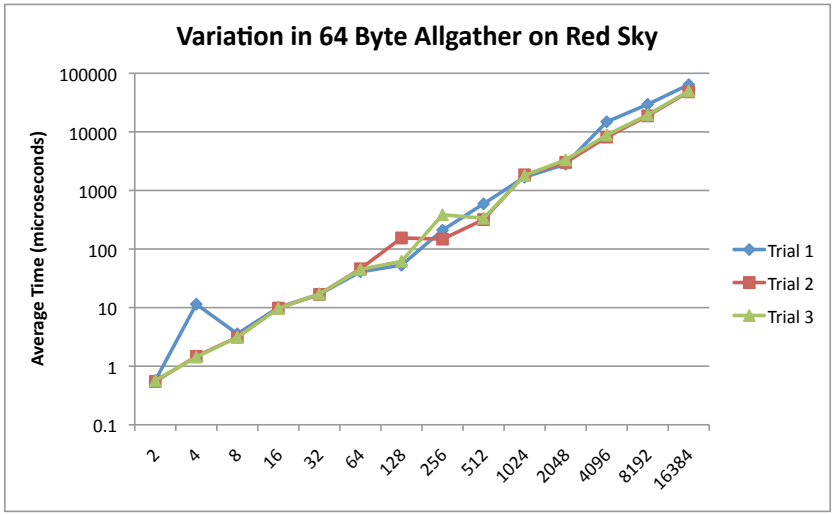
Sandia National Laboratories

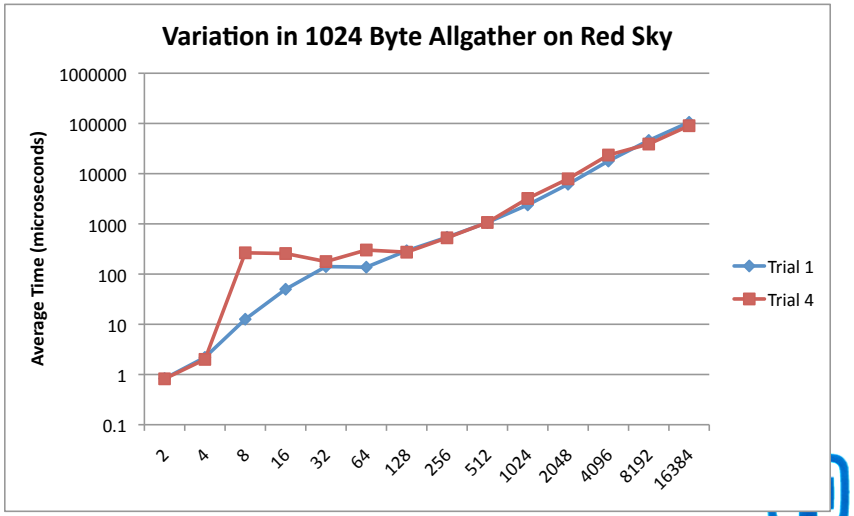
MPI Collectives: Allgather (lower better)







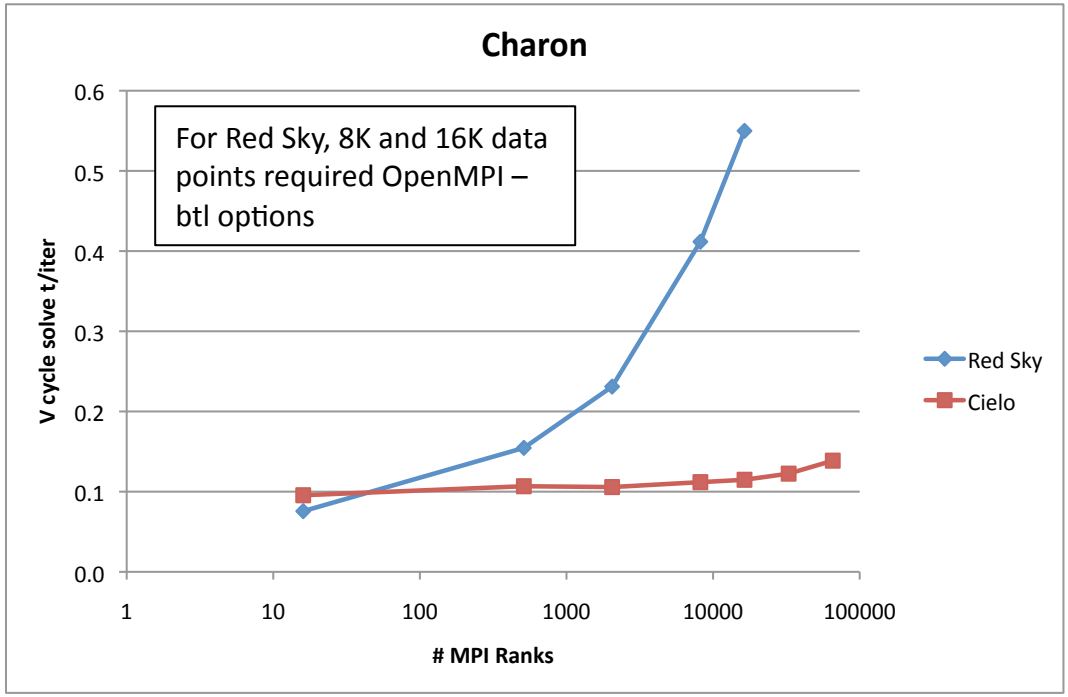




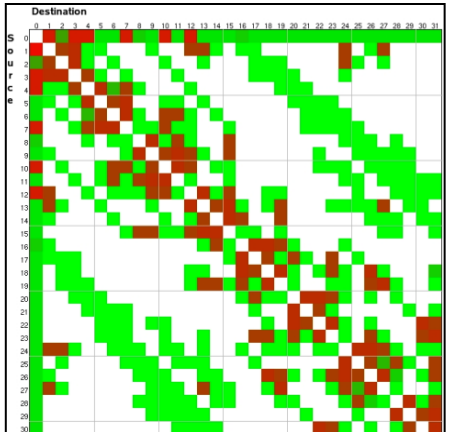
Acceptance Applications

Lab	Code	Fortran	Python	C	C++	MPI	OpenMP	Description
SNL	RAMSES/ Charon			X	X	X		A transport reaction code to simulate the performance of semiconductor devices under irradiation
SNL	CTH	X		X		X		Explicit, multi-material shock hydrodynamics code
LANL	xNOBEL	X		X		X		Continuous Adaptive Mesh Refinement (CAMR) code: Hydrodynamics with adaption and high-explosive burn modeling
LANL	SAGE	X		X		X		Multi-dimensional multi-material Eulerian hydrodynamics code with adaptive mesh refinement.
LLNL	AMG2006			X		X	X	Algebraic Multi-Grid linear system solver for unstructured mesh physics packages
LLNL	UMT2006	X	X	X	X	X	X	Single physics package code. Unstructured-Mesh deterministic radiation Transport.

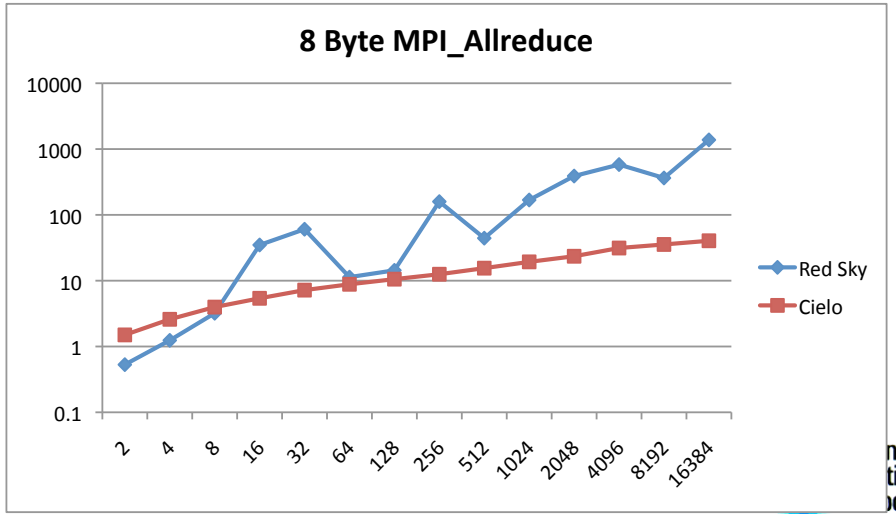
Cielo 6x application: SNL's Charon; Weak Scaling; (lower better)



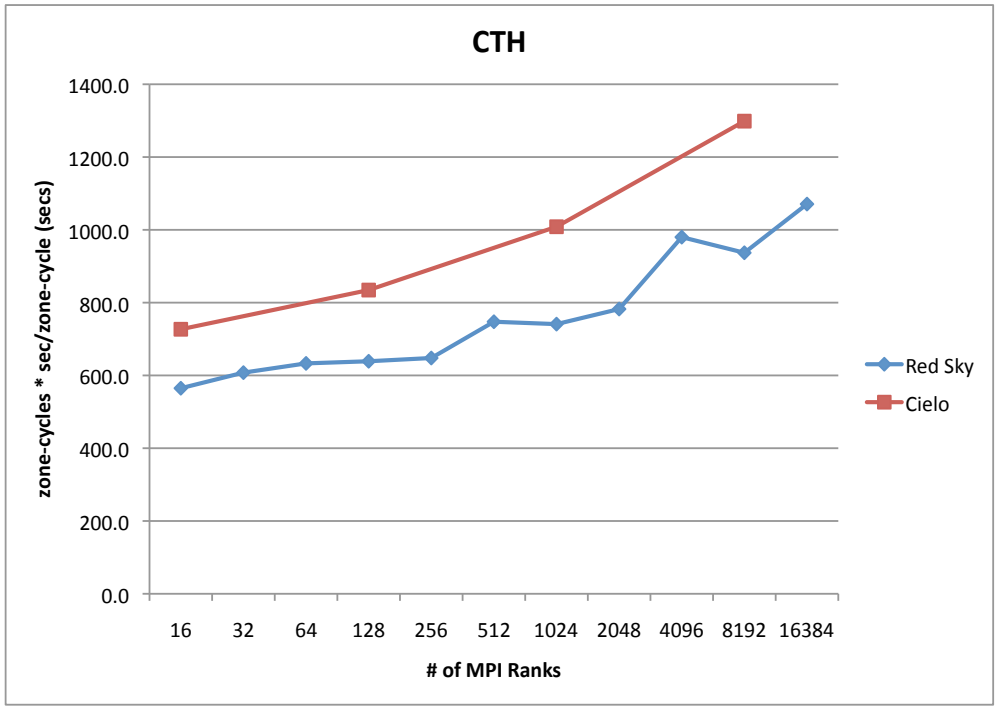
- Semiconductor device simulation code
- Finite element discretization produces a sparse, strongly coupled nonlinear system.
- A fully-coupled implicit Newton-Krylov solver is used with a multigrid preconditioner
- Performance dominated by small message transfers, avg 900 bytes, and small message Allreduce at scale



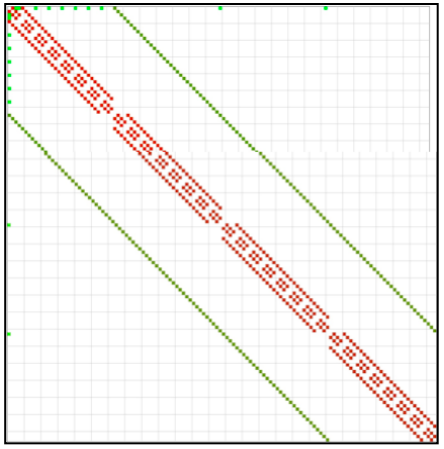
32 core message exchange CrayPat plot



Cielo 6x application: SNL's CTH; Weak Scaling; (lower better)

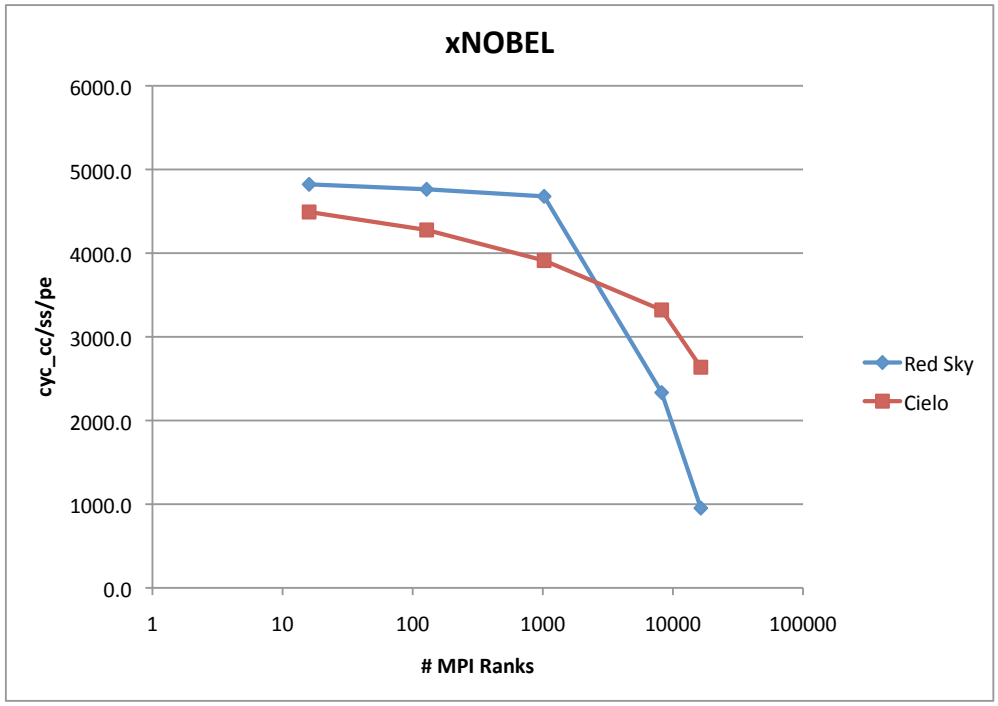


- CTH is used for two- and three-dimensional problems involving high-speed hydrodynamic flow and the dynamic deformation of solid materials
- Model: shaped-charge; cylindrical container filled with high explosive capped with a copper liner.
- Weak scaling analysis with 80x192x80 computational cells per processor.
- Processor exchanges information with up to six other processors in the domain (after 128 PEs). Dominated by large message (3MB) exchange and Allreduce

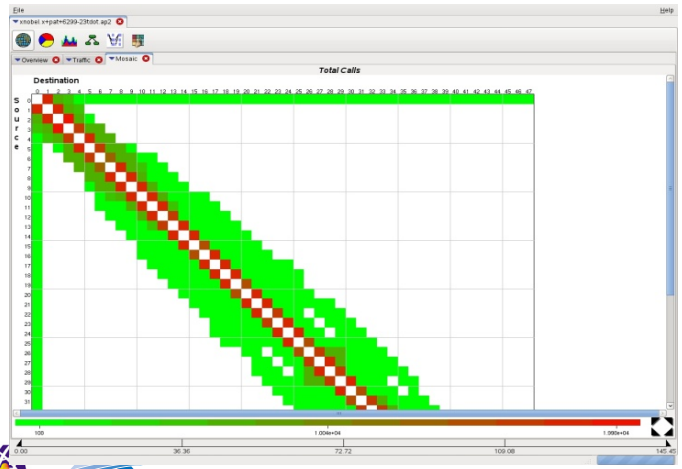


128 core message exchange CrayPat plot

Cielo 6x application: LANL's xNobel; weak scaling; (higher better)

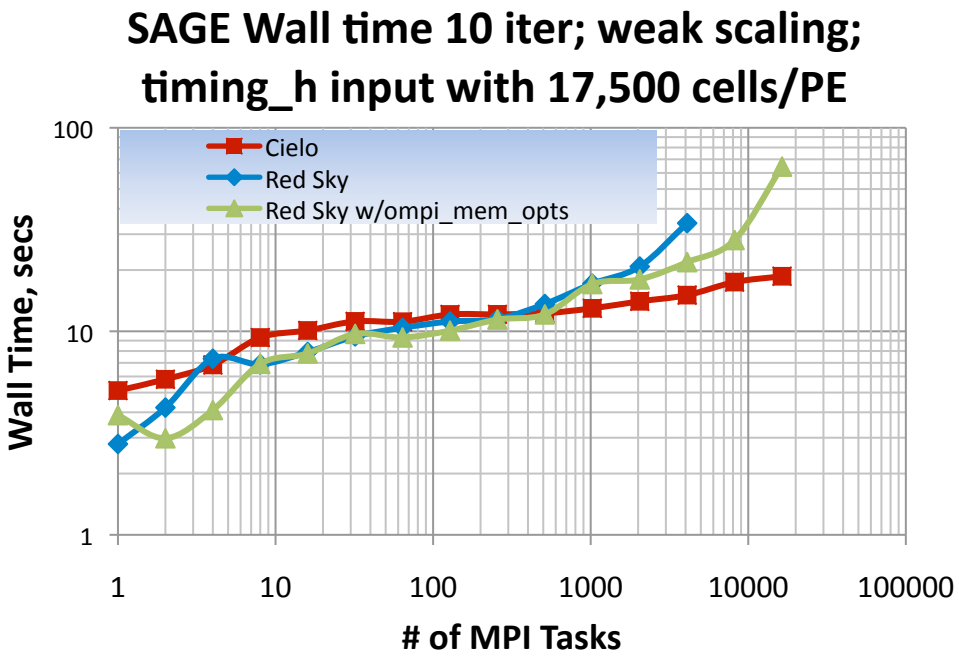


- Continuous Adaptive Mesh Refinement(CAMR) code: Hydrodynamics with adaption and high-explosive burn modeling
- Model: 3D simulation of a 105 mm shaped charge (sc301p) calculation and exercises this application just like a production simulation
- Run time dominated by MPI communication and MPI Sync: computation time small.
- MPI Barrier synchronization (2720 calls) and 4 Bytes MPI_Allreduce (4960 calls) Sync time are the dominant MPI overheads

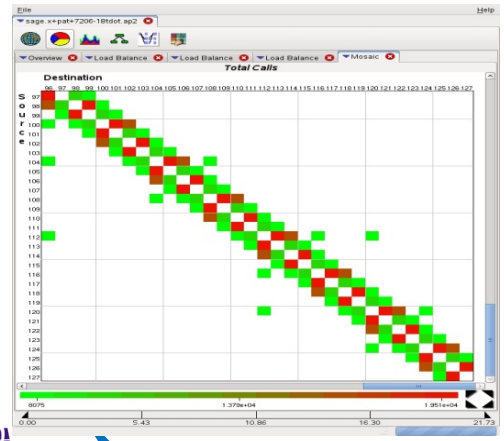


48 core message exchange CrayPat plot

Cielo 6x application: LANL's SAGE weak scaling; lower better

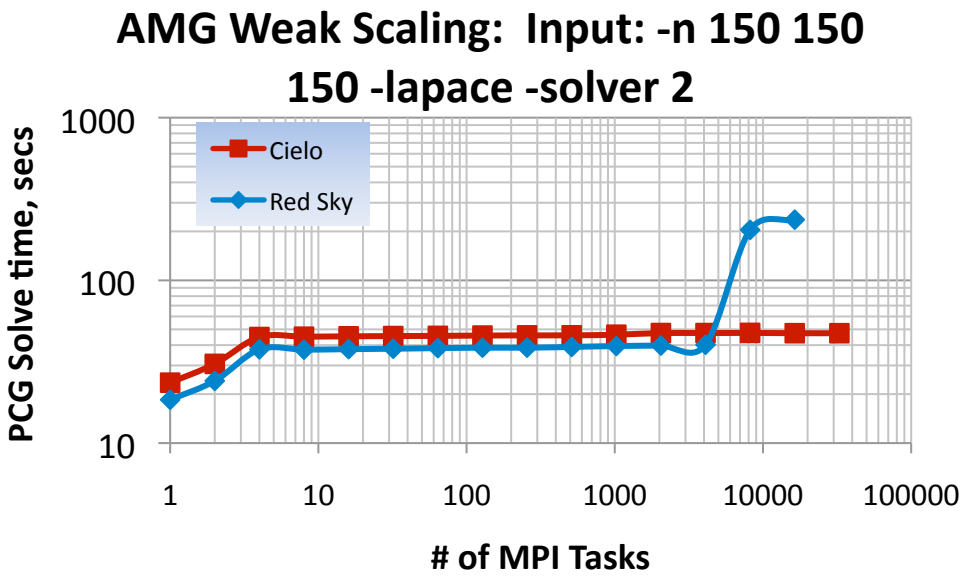


- SAGE is a LANL multi-dimensional multi-material shockwave Eulerian hydrodynamics code that uses Adaptive Mesh Refinement. In weak-scaling mode – the global problem size grows with the processor count while the problem size per socket remains constant.
- A modified 1-D data decomposition is used to partition the global 3-D mesh. This results in near-neighbor communications at small processor counts. The logical distance between communicating neighbors increases as the processor count increases. This can impact the cost of communications depending on the network topology and routing strategy employed
- MPI communication and Sync time takes approx. ¼ run time. MPI_waitall (max 16858 calls)and MPI_Recv (8 bytes, Max 16,736 calls) are the dominant MPI overhead

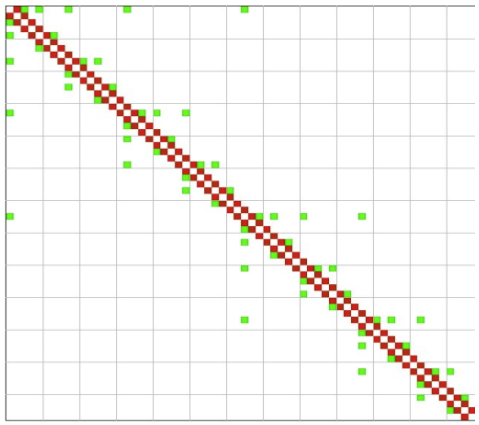


128 core message exchange CrayPat plot

Cielo 6x application: LLNL's AMG weak scaling; lower better



- Benchmark capable of both MPI and OpenMP
- But in these runs no hybrid parallelism is tested.
- The amount of data communicated between MPI tasks relative to the amount of computation is small.
- The main memory bandwidth need is large
- At scale dominated by MPI_Allreduce (sync) performance

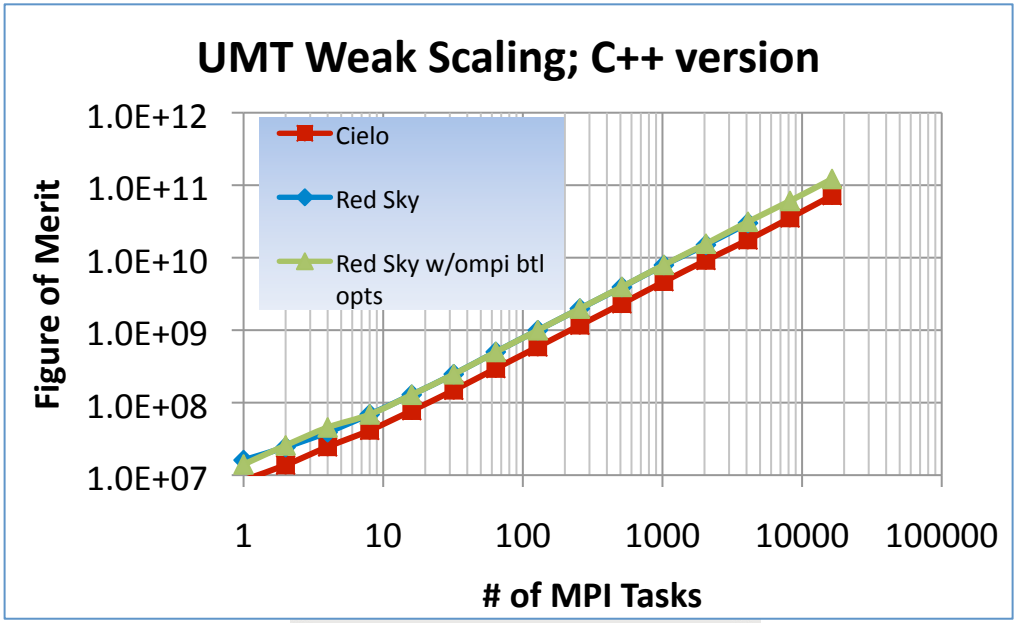


64 core message exchange CrayPat plot

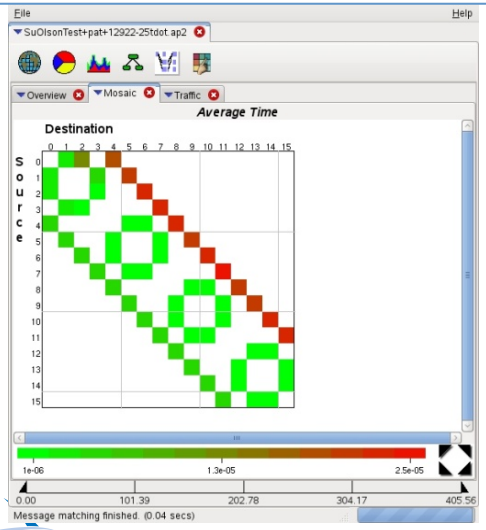
	4k Red Sky	4k Cielo	8k Red Sky	8k Cielo
Wall time (secs)	54.68	63.12	247.7	62.23
All_Reduce Comm time secs	0.779	0.034	9.01	0.033
All_Reduce Sync time (max)secs	8.23	13.66	189	13.54

National Nuclear Security Administration ASC

Cielo 6x application: LLNL's UMT weak scaling; figure of merit: larger better

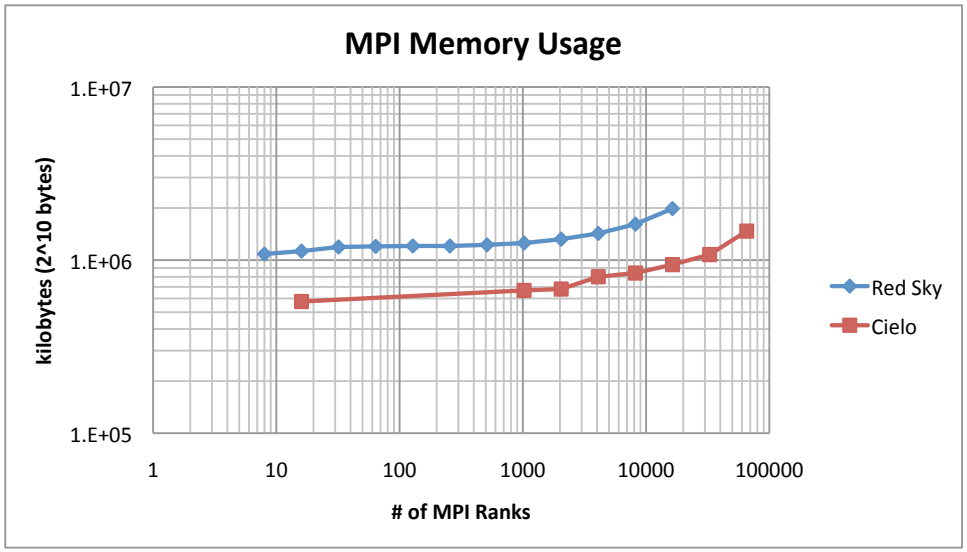


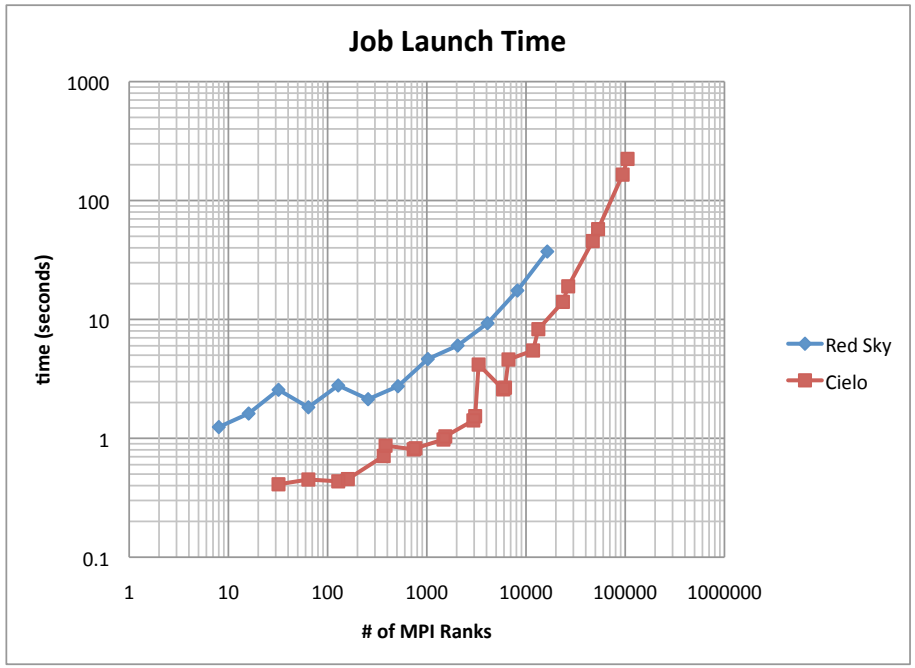
- UMT is a LLNL benchmark that provides an example of unstructured mesh deterministic particle transport
- The MPI-based parallelism uses mesh decomposition to distribute the mesh across the specified MPI tasks.
- Because the run-time of the application is overwhelmingly dominated by the kernel routines that implement the ordinate direction and energy group sweeps, UMT scales well to very large processor counts.
- Dominant MPI time is from MPI_Wait (< 8 % of run time)



16 core message exchange CrayPat plot

MPI Memory Usage & Job Launch Time

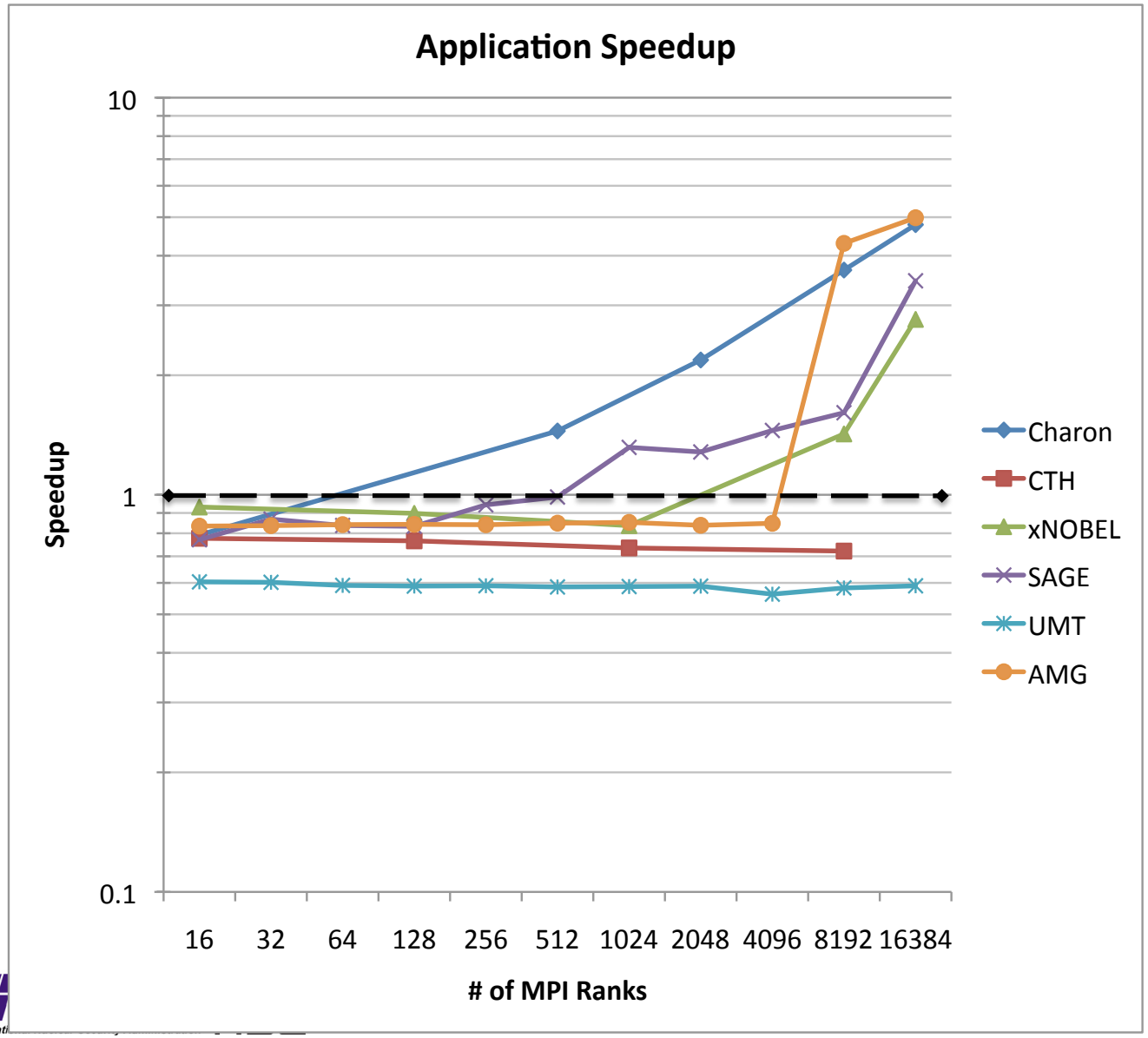




- MPI memory footprint and scaling are similar.
- Cielo's MPI has a smaller memory footprint, but not significant.
- Red Sky's footprint occupies 1/6th of the total memory at 16K Pes.

- Job launch time characteristics are similar.
- Cielo performs a little better, but not significant.
- Although launch times are small for long batch jobs, both platforms could use some improvement.

Application Speedup & Crossover



Crossover point where Cielo outperforms Red sky i.e. Speedup > 1

	# of PEs
Charon	40 to 50
CTH	none
xNOBEL	2000 to 3000
SAGE	500 to 600
UMT	none
AMG	4000 to 5000

Sandia National Laboratories

Red Sky OpenMPI "btl" Parameters and Impact

- Analysis by Brian Barrett and OpenMPI community
- Btl parameters required for large scale runs (8K and 16K)
  - --mca btl_openib_use_eager_rdma 1 --mca btl_openib_eager_limit 8192
    - Sets eager RDMA is on (but is the default already) sets cross-over for RDMA reads/ writes directly to/from memory to 8K (default is 12K). Probably no impact.
  - --mca btl_openib_receive_queues P,4096,8,6,4:P,8192,8,6,4
    - "This ones complicated …", but most significant is likely switch from shared receive queues to per-peer receive queues. Per-peer queue pairs are strictly flow-controlled at the MPI level.
  - --mca btl_openib_max_send_size 8192
    - The biggest single packet that will ever be sent using send/receive semantics.
- The Consensus
  - Brian consulted with the OpenMPI community and the general consensus was running out of QPs.
  - Brian suggests using "–mca btl_openib_receive_queues P4096,32,24,8;P8192,32,24,8"
    - Increases fragment count which may help message rate

Conclusions

- As has been historically demonstrated, first with ASCI Red vs. Cplant then Red Storm vs. TLCC, the capability platforms have a performance advantage at large scale as compared to “same generation” capacity platforms at Sandia
- Capacity platforms perform very well up to some crossover point, which has historically been in the 100’s of PEs range. Red Sky, architected for mid-range jobs, pushes the crossover to a few 1000’s of PEs for some applications and shows overall better performance for some at full scale, 16K PEs for this study.
- Red Sky scaling for large number of cores suffers from poor MPI global operations (sync time) and consequently we observed better scaling properties on Cielo
- Cielo, as per design goals for a capability system, scales well at large core counts; light weight OS and better MPI global ops performance contribute to scaling
- Red Sky QDR IB has good latency and bandwidth, but suffers from OpenMPI scalability issues (needed BTL MEMOPTS to successfully run at 8K & 16k for four of the six Cielo 6x apps)
- Run time variation on Red Sky can be substantial, depending on run-to-run job placement
- Further investigations on MVAPICH and OpenMPI (btl) options should be studied for performance tuning and feasibility of running IB at larger scales