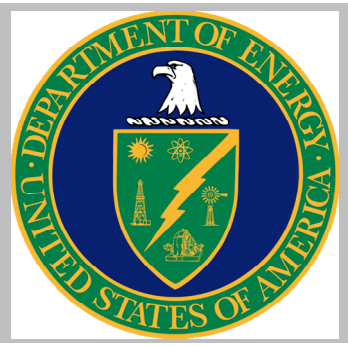
SAND2018-1199PE



Application of Performance Analysis Tools on SNL ASC Codes

9320 CapViz & Application Readiness Teams; 1300 & 1500 Production Application Development Teams

2018-02-14

U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND NO. 2018-1059 PE

Contributions

- SNL ASC FOUS FY17 L2 Milestone
  - Anthony M. Agelastos
  - Douglas M. Pase
  - Kathleen Amspaugh
  - Dennis Dinge
  - Karen Haskell
  - Lisa Ice
  - Justin Lamb
  - Mahesh Rajan
  - Ryan Shaw
  - Joel Stevenson
  - Victor Brunini
  - Jonathan Clausen
  - Martin Crawford
  - Greg Valdez
- Spectre Meltdown Analysis
  - Anthony M. Agelastos
  - Douglas M. Pase
  - Ruth Klundt
  - Steve Monk
  - Joel Stevenson
  - Justin Lamb
  - Jeff Ogden

Executive Summary

- SNL FOUS FY17 Level-2 (L2) Milestone #6019 “Application of Performance Analysis Tools on SNL ASC Codes”
  - Exercise and assess performance profiling tools
  - Engage with production code teams and use tools to assess application performance
  - Feed results to tool vendors and Tri-Lab partners to influence FY18+ CCE projects
- SNL Preliminary Assessment of Meltdown and Spectre
  - Analyze micro-benchmarks and production applications
  - Assess impact from MPI, memory allocation, I/O, and compute
  - Summarize results for broad dissemination and mitigation strategies

Application of Performance Analysis Tools on SNL ASC Codes

Milestone Specification

Description: Using currently available tools, specifically including those developed and/or provided within the CCE, characterize the performance of at least two codes (to be determined) on CTS-1 and Trinity platforms. Identify strengths and gaps in the performance analysis tool set and, in concert with application readiness support for new platforms, provide insights into possible paths for increasing platform productivity via code performance improvements.

Completion Criteria: Deliver analysis results to product owners of the targeted codes and publish usage guidelines for the application of the various tools.

Certification Method:

Professional documentation, such as a report or a set of viewgraphs with a written summary, is prepared as a record of milestone completion.

The “handoff” of the developed capability (product) to a nuclear weapons stockpile customer is documented.

Milestone NW/ASC Customers

- **Sierra/Aria**
  - Paul Crozier, Manager, 1541, Computational Thermal and Fluid Mechanics
  - Jonathan Clausen, 1516, Fluid and Reactive Processes
  - Victor Brunini, 8253, Thermal/Fluid Science & Engineering
- **RAMSES ITS**
  - Leonard Lorence, Manager, 1341, Radiation Effects Theory Department
  - Martin Crawford, 1341, Radiation Effects Theory Department
  - Greg Valdez, 1341, Radiation Effects Theory Department

Milestone Objectives

- Exercise a broad set of performance profiling and analysis tools, particularly including tools whose development has been promoted by the ASC program, to gain experience with and knowledge about the tools, strengths and weaknesses, utility on various platforms, etc. Tools included:
  - FOUS-funded: HPCToolkit, Allinea MAP, Open|SpeedShop, TAU
  - Extras: LDPXI, mpiP, Cray Perftools, Vampir

- Exercise the tools on two different SNL ASC codes, one a Sierra code (Aria, a C++ codebase) and one a RAMSES code (ITS, a Fortran codebase). The milestone generated a plethora of strong and weak scaling, trend and profile data for multiple versions and problem cases for each of the two codes.

- Exercise the tools, according to availability, on multiple platforms, particularly including CTS-architecture platform(s) and the ATS Trinity platform. Platforms used included Chama (TLCC-2 architecture), Sky Bridge (TLCC-2 architecture), Serrano (CTS-1 architecture), and Phases 1 and 2 of Mutrino (ATS-1 Trinity architecture).

Milestone Code / Tool / Platform Table

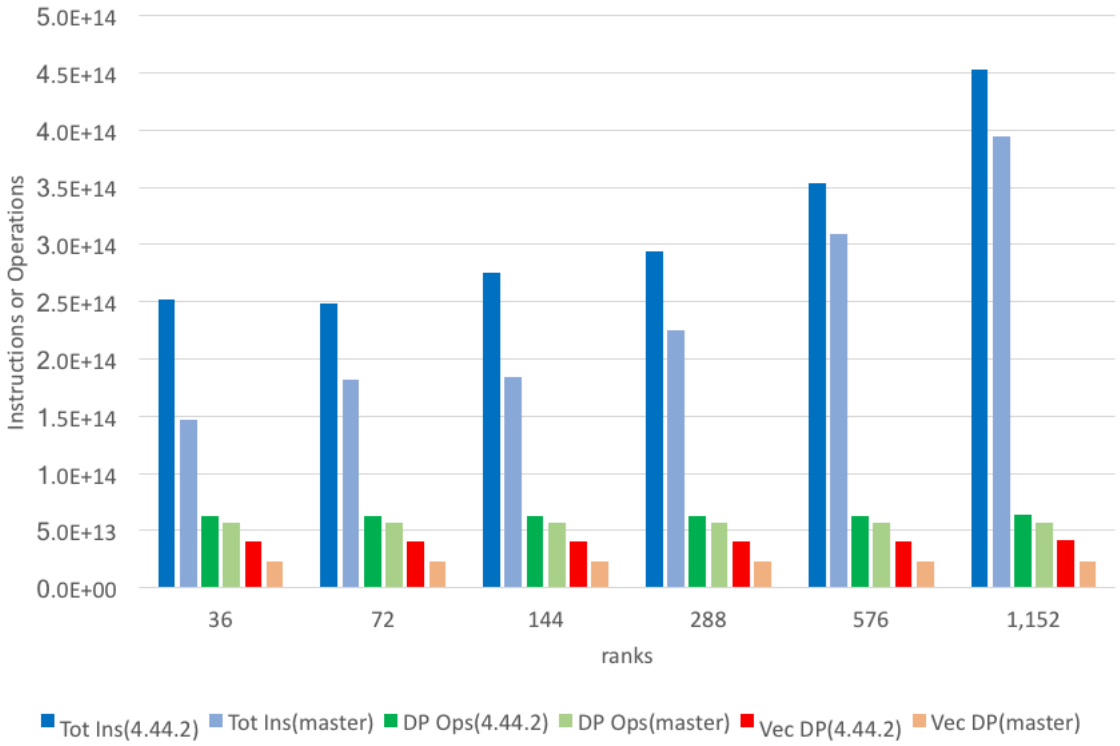
Profiler	Chama	Sky Bridge	Serrano	Mutrino Phase 1	Mutrino Phase 2
LDPXI		A,I	A,I		
mpiP	A,I	A,I	A,I		
Cray Perftools				A,I	A,I
Allinea MAP	A,I	A,I	A,I	A,I	
HPCToolkit	A	A,I	A,I		
Open|SpeedShop	A,I	A,I	A,I		
TAU			I		
Vampir			I		

A	Denotes profiler was utilized on *Sierra/Aria*
I	Denotes profiler was utilized on *ITS*
	Denotes profiler is unavailable
	Denotes profiler is installed but issues were found

Some Noteworthy Highlights - Aria

Sierra/Aria

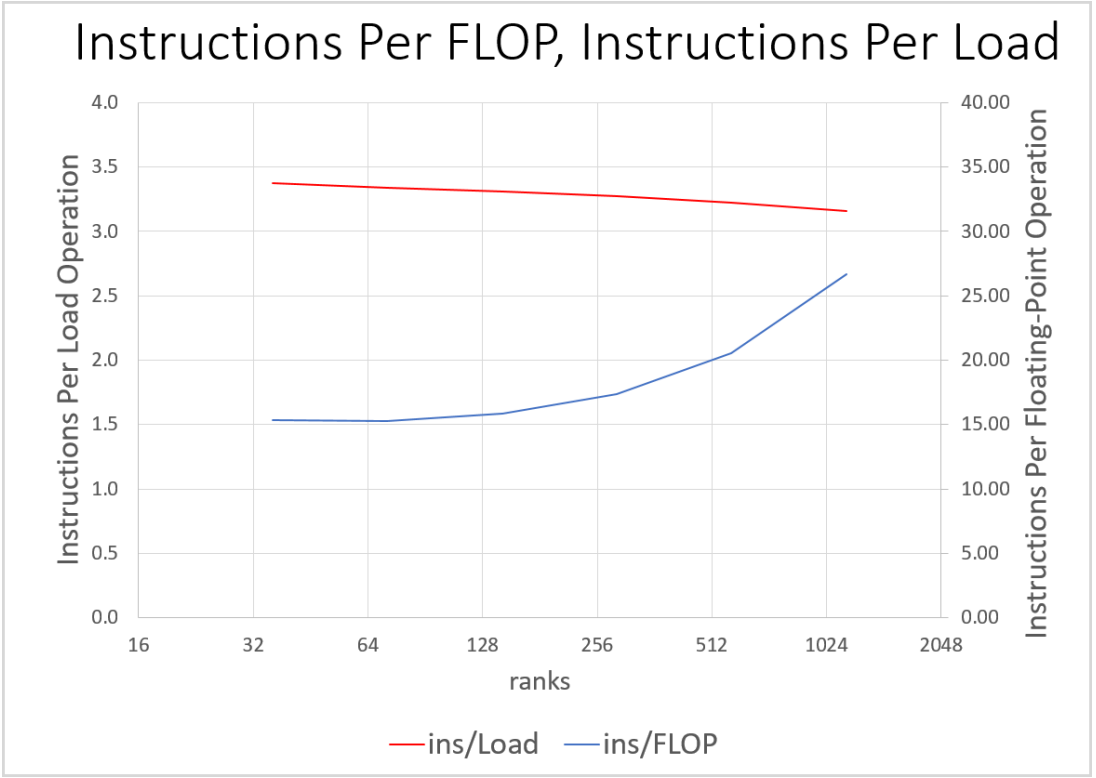
- Many of the metrics got worse, but the wall time improved from 4.44.2 to master. This occurs because version master significantly reduces the number of total instructions needed to compute a solution. The best optimization of all is eliminating work that does not have to be done. Notice also that as the number of ranks increases, the floating-point work (i.e., the “useful” work) stays constant.



Some Noteworthy Highlights - ITS

ITS

- This figure shows the high amount of non-floating-point work that ITS performs. Since ITS has a very high floating point vectorization rate (when it does do a FLOP, it's almost always vectorized which is good) and excellent cache utilization, it is this figure which I think we may want to dig deeper on when starting to look at optimal threading strategies for NGP.



Milestone Impact and Path Forward

Impact

- A wealth of knowledge and experience gained with the various tools that included identification of problems, an improved understanding of feature sets, enhanced usage documentation, and insights that may influence future ASC and FOUS program-element tool-development activities.
- Results from a large number and variety of performance analysis runs with the two target codes, available to the code teams, together with specific instructions for how to make use of the tools with the target codes.
- A 284-page SAND Report (SAND2017-9933) was generated that contains results and steps utilized to generate profile data.

Path forward

- Follow-up activities with the Sierra/Aria and/or ITS code teams to explore deep dives of interest.
- Working with other SNL codes and code teams to apply tools in support of application performance analysis.
- Sharing of results with tool vendors/providers where appropriate.
- Sharing of results with NNSA tri-lab counterparts (e.g., in support of the Common Computing Environment [CCE]), the DOE high performance computing community, and beyond.

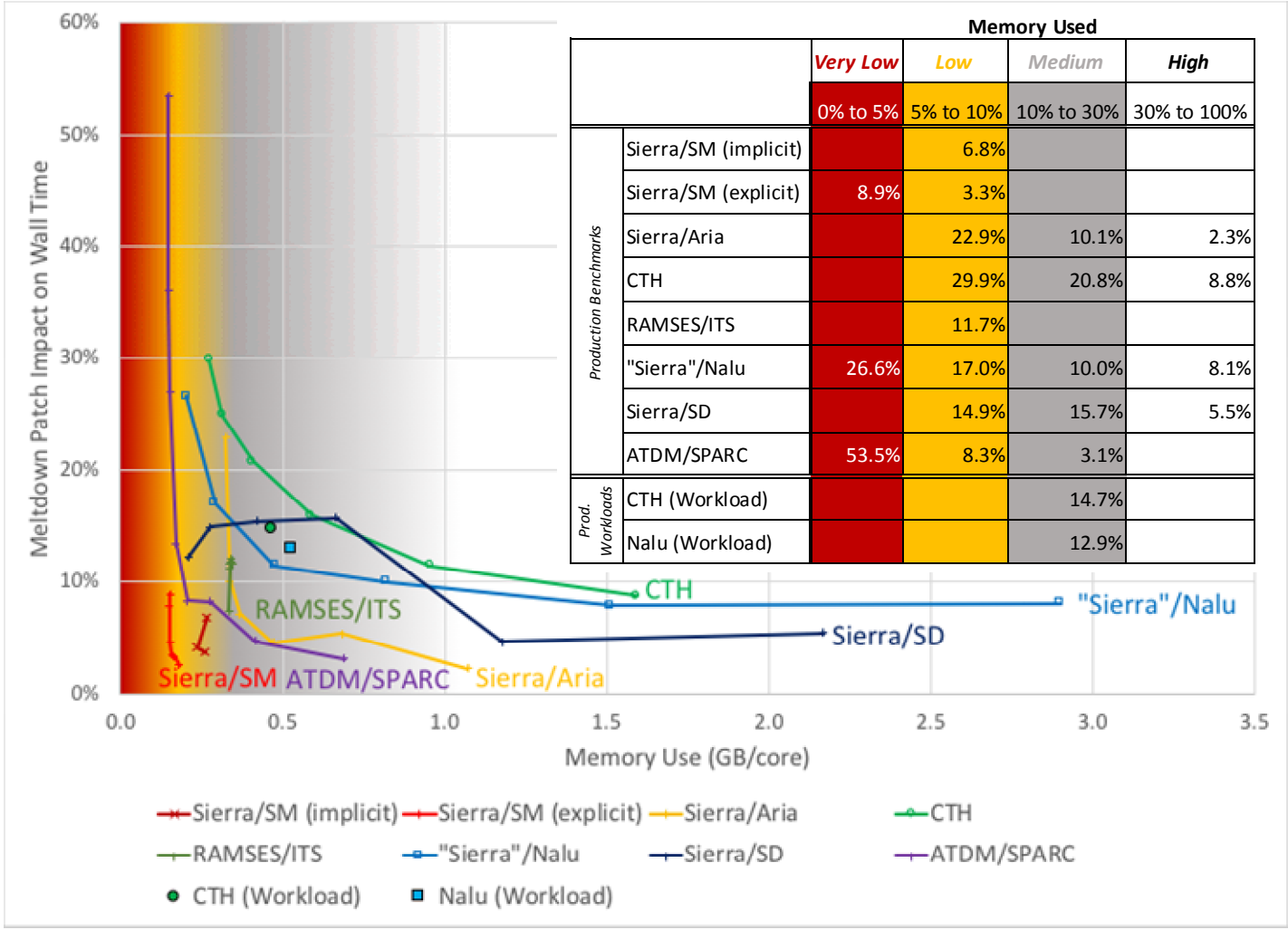
Description of Testing Environment for Meltdown/Spectre-1,2 Patch

Sandia National Laboratories

- Testing and patches were performed and applied, respectively, upon Eclipse.
- Eclipse is a new Commodity Technology System (i.e., CTS-1) at Sandia National Laboratories.
- The following patches applied to Eclipse during testing included all mitigations originally available for:
  - CVE-2017-5753 (variant #1/Spectre) addresses bounds-checking exploit during branching (kernel patch, always enabled)
  - CVE-2017-5715 (variant #2/Spectre) addresses indirect branching poisoning attack (kernel patch + microcode, enable/disable provided)
  - CVE-2017-5754 (variant #3/Meltdown) addresses speculative cache loading (kernel patch, enable/disable provided)
- Testing with mitigations disabled were with all flags disabled (i.e., Spectre variant 2 and Meltdown).

Wall Clock Time Impact from Meltdown/Spectre-1,2 Patch

- A broad range of applications representing SNL production-relevant work was used.
- Memory use per core was found to be the principal factor affecting runtime efficiency for SNL codes (see slide 6).
  - ***The vast majority of SNL production HPC applications use more than 5% of the available memory per core.***
- Apps were run at modest scale under 3 test conditions:
  - Before the patch was installed
  - After the patch was installed
  - After the patch was installed but with mitigations disabled
- Performance was the same without the patch and with the patch mitigations disabled.
  - This enables installation of the patch with the ability to revert the performance loss when necessary.
- A small increase in run-to-run variability was observed which is likely due to factors unrelated to the patch.
- Large ensemble studies that have ***very low*** memory requirements (e.g., UQ) may exhibit *significant* increase in wall clock time.



		Memory Used			
		Very Low	***Low***	***Medium***	***High***
		0% to 5%	5% to 10%	10% to 30%	30% to 100%
Production Benchmarks	Sierra/SM (implicit)		6.8%		
	Sierra/SM (explicit)	8.9%	3.3%		
	Sierra/Aria		22.9%	10.1%	2.3%
	CTH		29.9%	20.8%	8.8%
	RAMSES/ITS		11.7%		
	"Sierra"/Nalu	26.6%	17.0%	10.0%	8.1%
	Sierra/SD		14.9%	15.7%	5.5%
	ATDM/SPARC	53.5%	8.3%	3.1%	
Prod. Workloads	CTH (Workload)			14.7%	
	Nalu (Workload)			12.9%	

Bottom Line: The impact ranged from **3%** to **30%** except for the “very low” region (impact up to 54%)

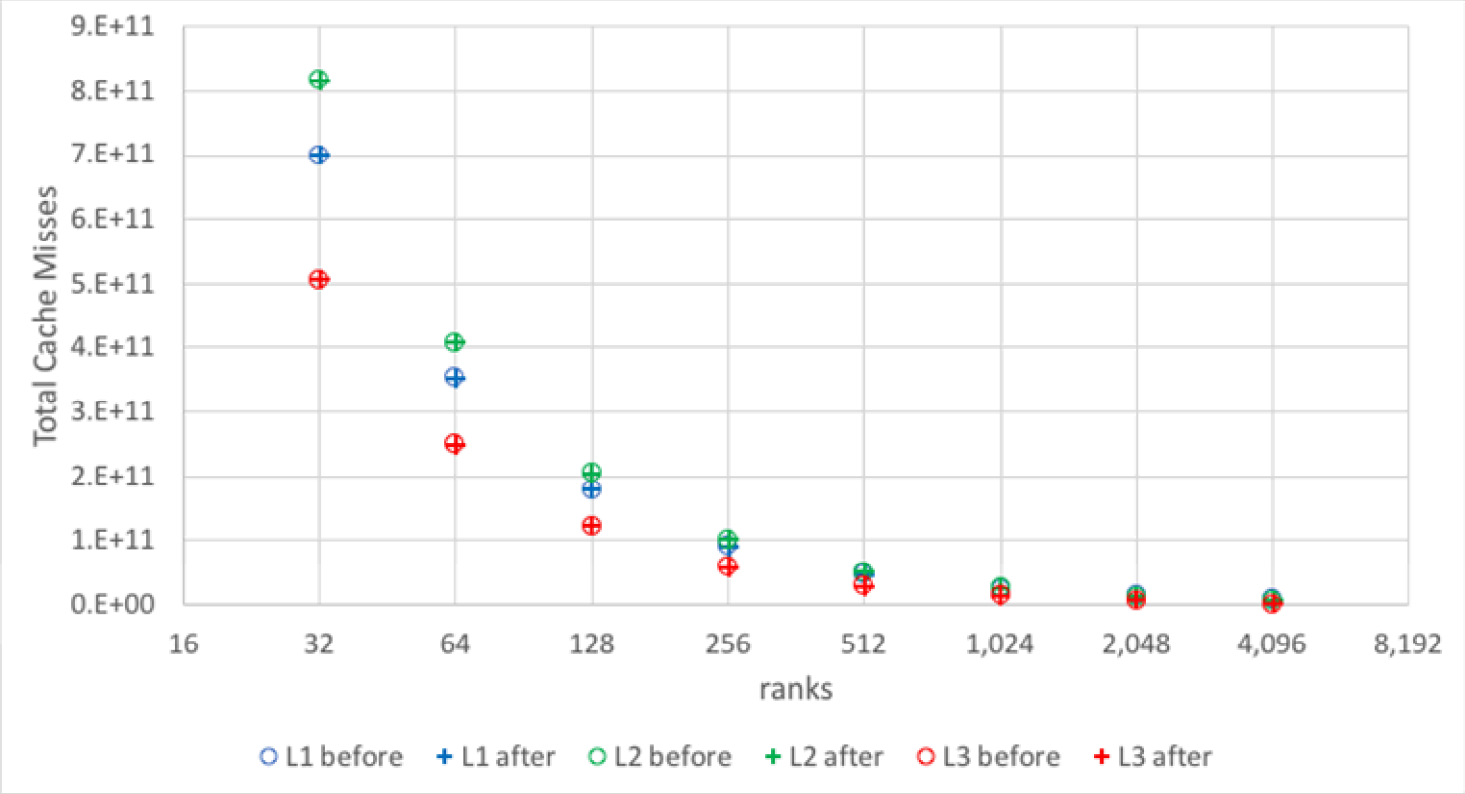
Factors Contributing to Performance Loss from Patch

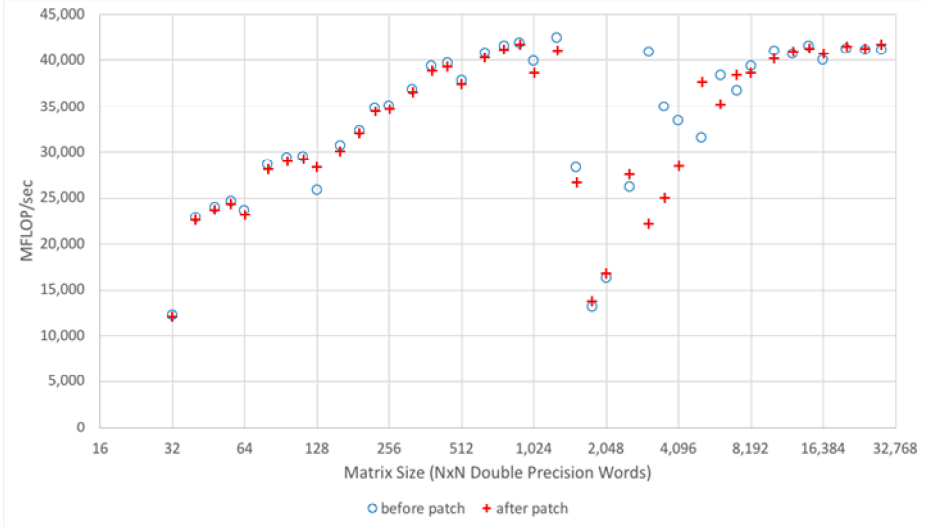
- These production workloads were created to include representative levels of file system I/O.
  - Checkpoint/restart files were output every hour.
  - In-situ visualization and results data was output frequently.
- Contributing Factors:
  - MPI time increased, more in some cases than others.
  - Both user and system time increased (i.e., not just time in the kernel).
  - File read/write time increased but was a small contributor to overall workload loss in these tests.
  - Memory allocation/deallocation/reallocation time increased but was a small contributor to overall workload loss in these tests.
- Non-contributing Factors (shown in next slide):
  - L1, L2, and L3 cache hits/misses were unchanged.
  - Pure core and memory time were unchanged.
- I/O, memory allocation, and MPI are all impacted by the patch; SNL applications spend more time in MPI than I/O and memory allocation, **therefore the dominant impact for SNL production codes was attributable to MPI**.

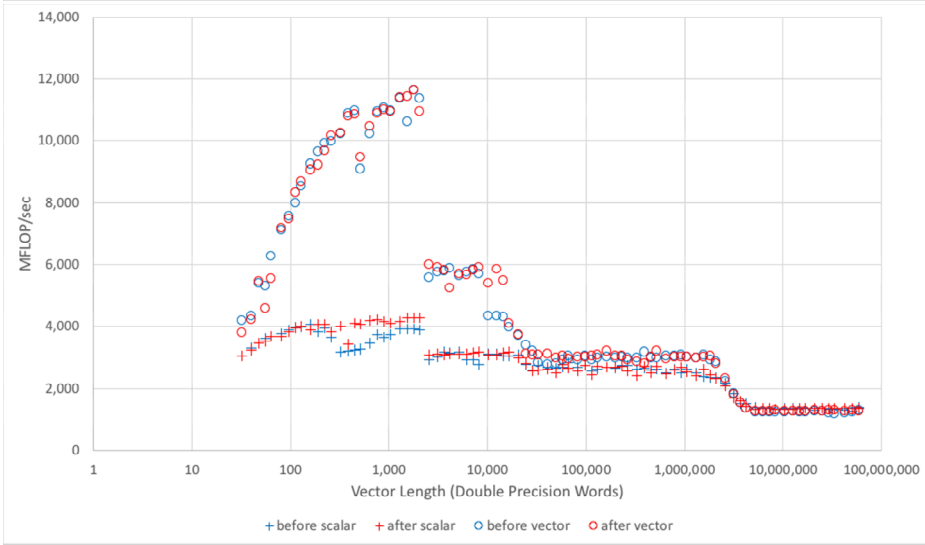
	Activity	Wall Clock Contribution	*Activity* Performance Loss	*Overall Workload* Loss
CTH Prod. Workload	MPI	76.4%	19.2%	**14.7%**
	I/O	0.1%	30.7%	**0.0%**
	Memory Allocation	0.8%	79.1%	**0.7%**
	Compute	22.7%	0.0%	**0.0%**
Nalu Prod. Workload	MPI	40.2%	32.0%	**12.9%**
	I/O	0.7%	-37.9%	**-0.3%**
	Memory Allocation	0.5%	38.5%	**0.2%**
	Compute	58.6%	0.0%	**0.0%**

Bottom Line: SNL apps that spend a higher fraction of time in MPI will see the largest impacts

Factors Not Contributing to Performance Loss



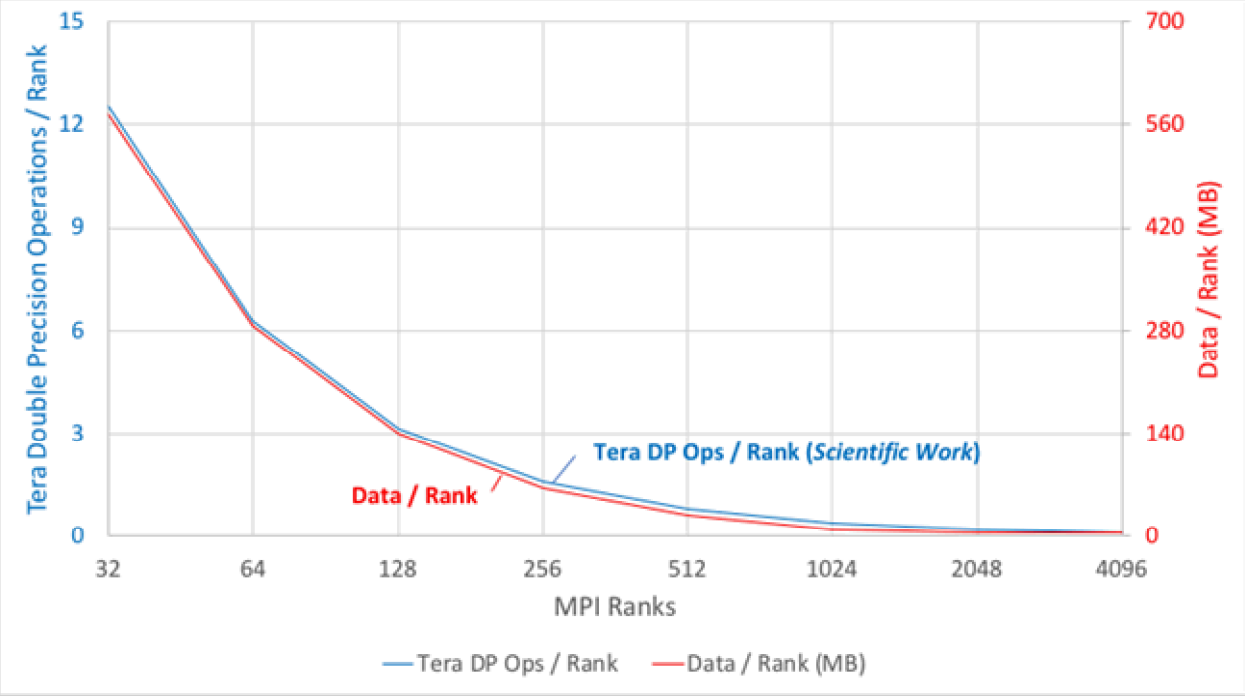


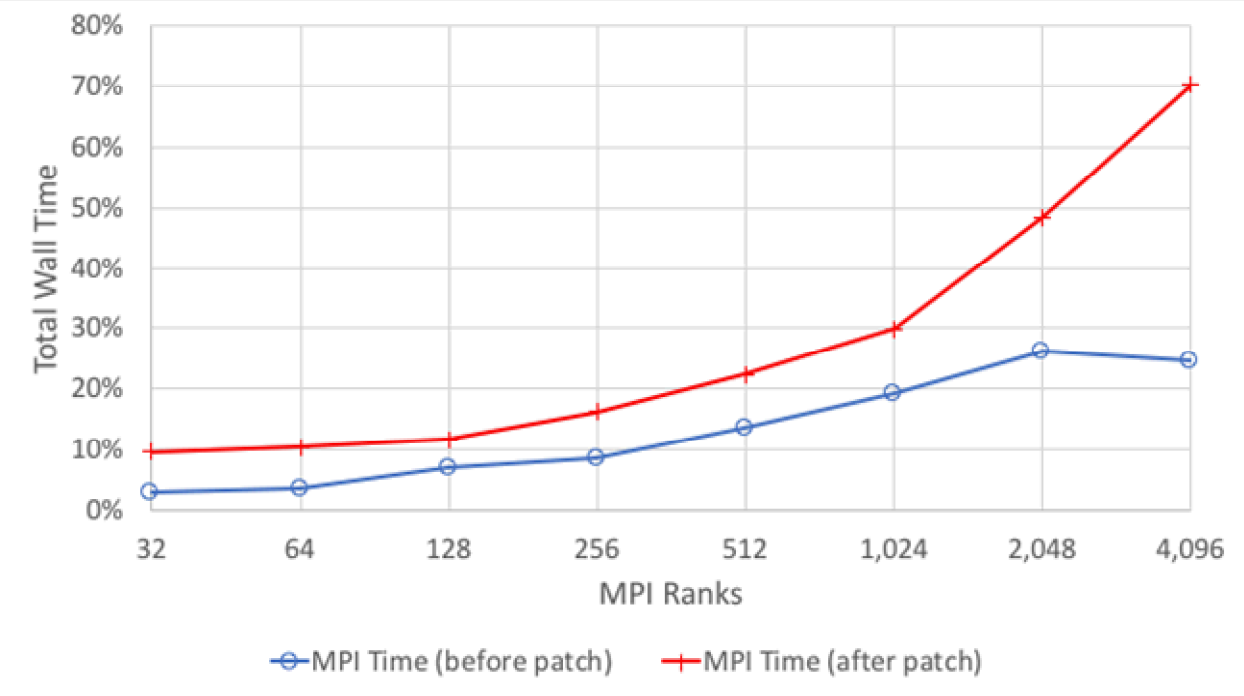


Top-left figure shows ATDM/SPARC Production Benchmark (highlighting L1, L2, L3 cache misses), top-right shows DGEMM (highlighting floating point operation rate for varying matrix sizes), and bottom-right shows DAXPY (highlighting cache and memory performance).

Bottom Line: Patch Does Not Impact Cache, Core, or Memory Performance

Explanation for Memory Use Per Core as Principal Factor for SNL Codes





- The strong scaling example shown here is the ATDM/SPARC Production Benchmark.
- The amount of scientific computation work performed is directly proportional to the amount of memory allocated (not including application binary).

- Relative MPI time increases as the amount of scientific work decreases.
- For fixed scientific work, e.g., strong scaling shown above, increasing the number of MPI ranks reduces the amount of scientific work per rank which increases the relative overhead and, as a result, the impact of the patch.

Bottom Line: Patch Impact for SNL is Inversely Proportional to Memory Per Rank

Description of Applications Used for this Study

- Micro-benchmarks
  - **DAXPY**, part of Netlib/LAPACK/BLAS-1 and many other numerical libraries, is "**D**ouble precision result of **A** times **X**(i) **P**lus **Y**(i)." We used this to quantify cache and memory performance.
  - **DGEMM**, part of Netlib/LAPACK/BLAS-3 and many other numerical libraries, is "**D**ouble precision **GE**neral **M**atrix-matrix **M**ultiplication." We used this to highlight floating point operation rate for varying matrix sizes.
- All test cases for the following Production Benchmarks and Production Workloads are developer-produced, analyst-relevant use cases that are utilized to track performance deviation in production features. A preference was given to easily scalable models.
- Sierra Code Suite
  - **Sierra/Aria** is a finite element analysis code for the solution of coupled, multiphysics problems with a focus on energy transport, species transport with reactions, electrostatics, and incompressible fluid flow.
  - **Sierra/SM** is a 3-D, nonlinear, structural, statics and dynamics code with explicit and implicit time integration.
  - **Sierra/SD** is a massively parallel code for structural dynamics finite element analysis.
- Nalu
  - **Nalu** is a generalized, unstructured, massively parallel, low Mach CFD code built on the Sierra Toolkit and Trilinos solver stack.
- CTH
  - **CTH** is a multi-material, large deformation, strong shock wave, solid mechanics code. The code is explicit and uses finite volume difference for the numerical simulation of the high-rate response of materials to impulsive loads.
- RAMSES/ITS
  - **RAMSES/ITS** is a software package of codes that provide Monte Carlo solutions of multi-dimensional linear time-independent coupled electron/photon radiation transport problems.
- ATDM/SPARC
  - **ATDM/SPARC** is a compressible CFD code that is capable of solving aerothermal, aerodynamics, and aerostructural problems.

Conclusion(s)

- Calibrating and assessing performance tools is very important to gain confidence with their results.
  - These studies foster collaboration with the tool vendors to improve their offerings.
- Developing and gathering production-relevant test cases that can be strong- and weak-scaled is very important to understand broad impact upon the user community.
  - These studies foster collaboration with the application developer and analyst communities.
- Tools that support batch profiling with minimal-to-no impact on production builds are preferred for ensemble studies.
  - Reliable and scriptable processes enable rapid, large-scale ensemble performance studies.