SAND2018-1059PE

Preliminary Assessment of Impact from Patch for Meltdown and Spectre (variants 1 & 2) on Sandia National Laboratories' HPC Production Operations Using ASC Integrated Codes

9320 CapViz & Application Readiness Teams – 2018-01-31

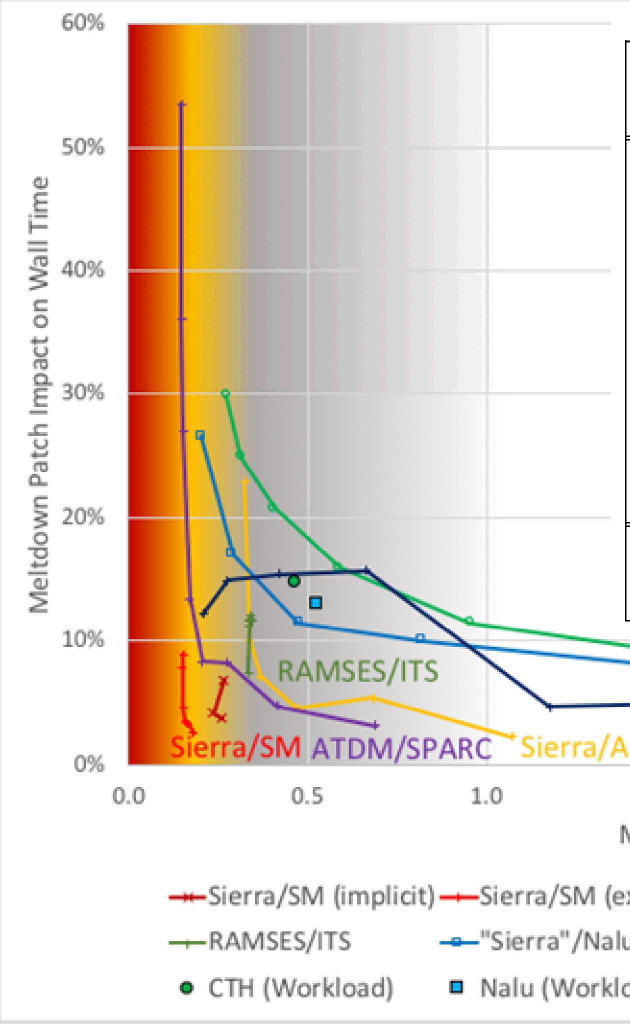
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND NO. 2017-XXXXP

Description of Testing Environment

- 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%)

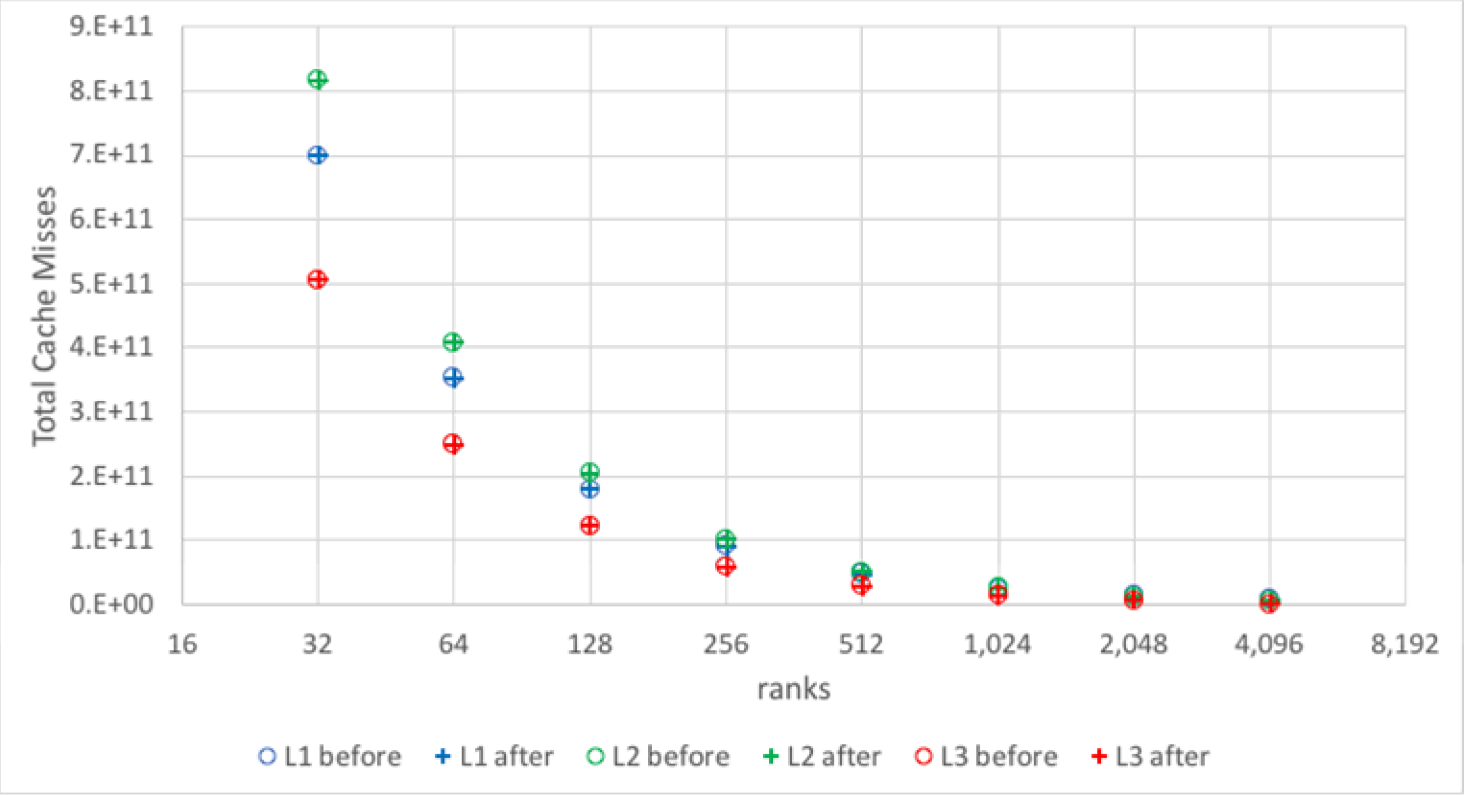
Factors Contributing to Performance Loss from Patch

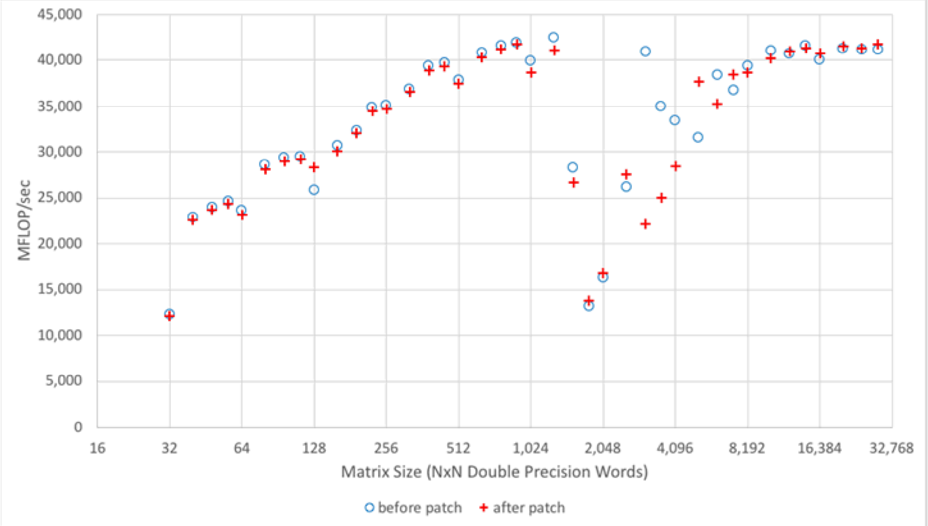
- These production workloads were created to include representative levels of file system I/O.
  - Checkpoint/restart files were output every hour.
  - In-situ visualization and results data was output frequently.
- Contributing Factors:
  - MPI time increased, more in some cases than others.
  - Both 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

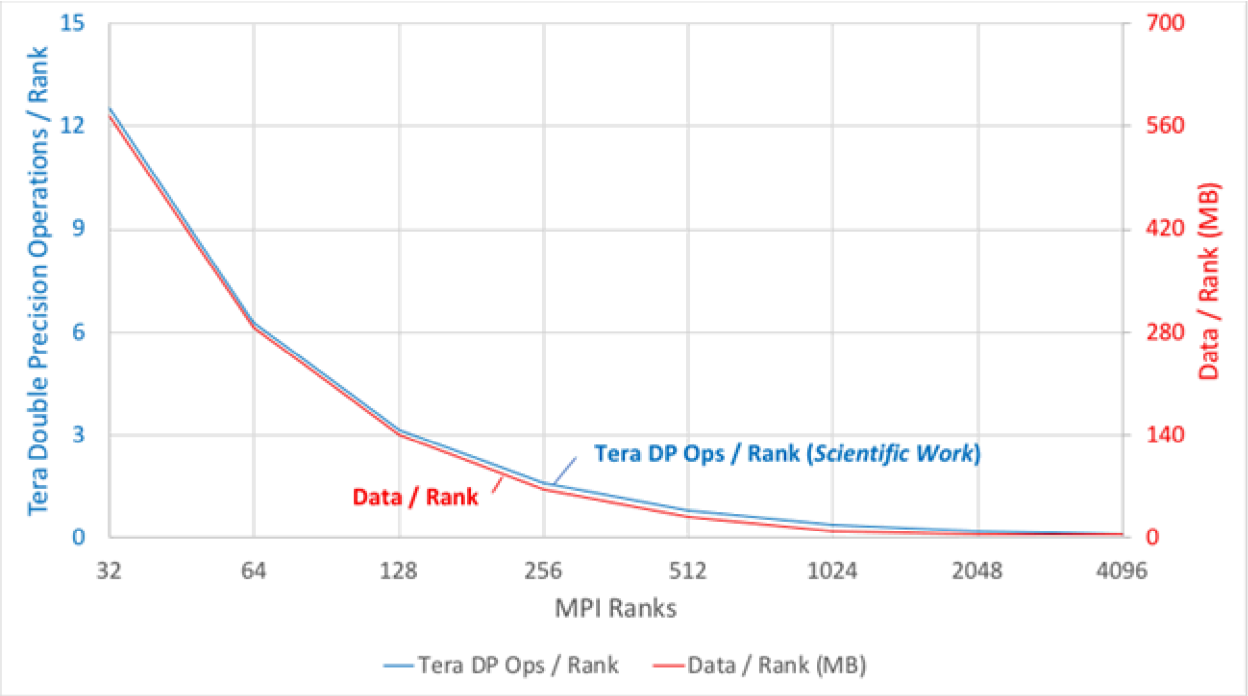


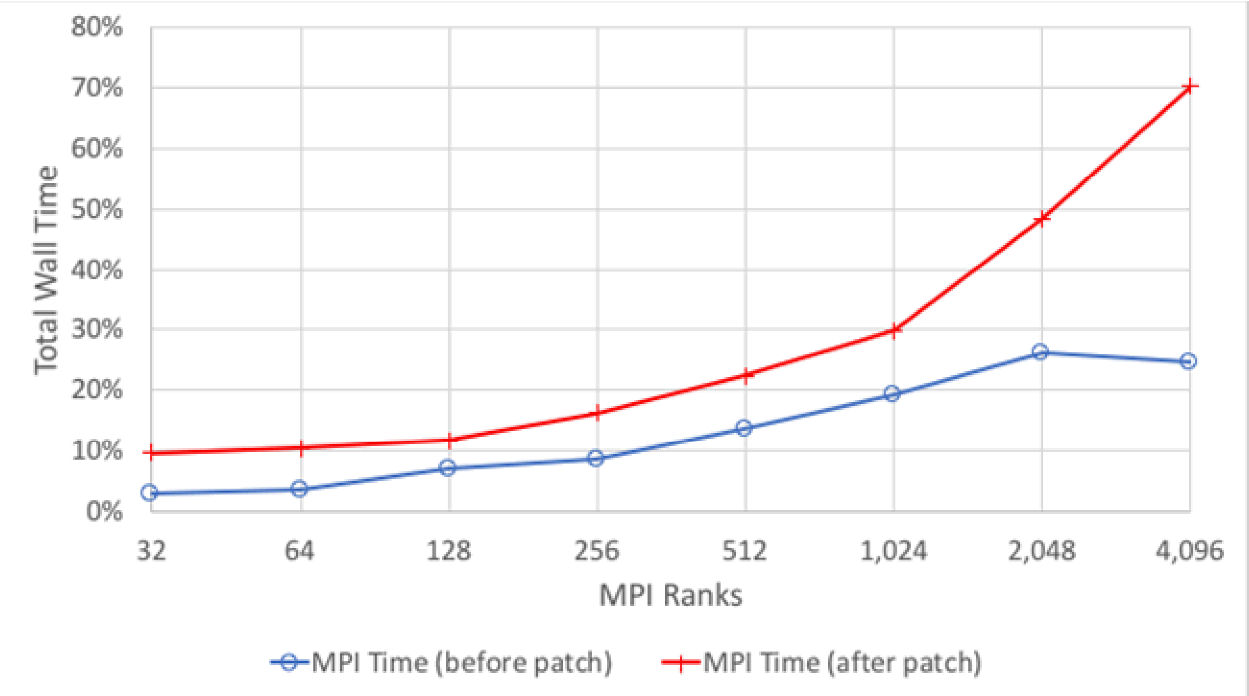


Top-left figure shows ATDM/SPARC Production Benchmark (highlighting L1, L2, L3 cache misses), top-right shows DGEMM (highlighting floating point operation rate for varying matrix sizes), and bottom-right shows DAXPY (highlighting cache and memory performance).

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.