Studying Effects of Meltdown and Spectre Patches on the Performance of HPC Applications Using Application Kernel Module of XDMoD

Nikolay A. Simakov
Center for Computational Research
SUNY University at Buffalo
Buffalo, USA
nikolays@buffalo.edu

Martins D. Innus
Center for Computational Research
SUNY University at Buffalo
Buffalo, USA
minnus@buffalo.edu

Matthew D. Jones
Center for Computational Research
SUNY University at Buffalo
Buffalo, USA
jonesm@buffalo.edu

Ohad Katz
Department of Computer Science and Engineering
SUNY University at Buffalo
Buffalo, USA
ohadkatz@buffalo.edu

Joseph P. White
Center for Computational Research
SUNY University at Buffalo
Buffalo, USA
jpwhite4@buffalo.edu

Ryan Rathsam
Center for Computational Research
SUNY University at Buffalo
Buffalo, USA
ryanrath@buffalo.edu

Steven M. Gallo
Center for Computational Research
SUNY University at Buffalo
Buffalo, USA
smgallo@buffalo.edu

Robert L. DeLeon
Center for Computational Research
SUNY University at Buffalo
Buffalo, USA
rldeleon@buffalo.edu

Thomas R. Furlani
Center for Computational Research
SUNY University at Buffalo
Buffalo, USA
furlani@buffalo.edu

***Abstract*—In this work we examine how the updates addressing Meltdown and Spectre vulnerabilities impact the performance of HPC applications. To study this we use the application kernel module of XDMoD to test the performance before and after the application of the vulnerability patches. The application kernel module is designed for continuous performance monitoring of HPC systems. Here, we tested the performance difference for multiple applications and benchmarks including: NWChem, NAMD, GAMESS, ENZO, HPCC, IOR, MDTest and IMB. The results show that although some specific functions can have execution times decreased by as much as 74%, the majority of individual metrics indicate little to no decrease in performance. The real-world applications show a 2-3% decrease in performance for single node jobs and a 5-11% decrease for two node jobs. For node-counts up to 8 the degradation continues to increase reaching 27% in some cases.**

***Index Terms*—HPC, Security, Performance**

I. Introduction

The recently discovered Meltdown [1] and Spectre [2] vulnerabilities allow reading of process memory by other unauthorized processes. This poses a significant security risk on multi-user platforms including HPC resources that can result in the compromise of proprietary or sensitive information [1, 2]. Software patches released to mitigate the security vulnerabilities have the potential to significantly impact performance. According to Redhat [3] Linux OS remedies can degrade performance overall by 1-20%. In order to quantify the impact, particularly on HPC applications, we performed independent tests utilizing XDMoD's application kernel capability [4].

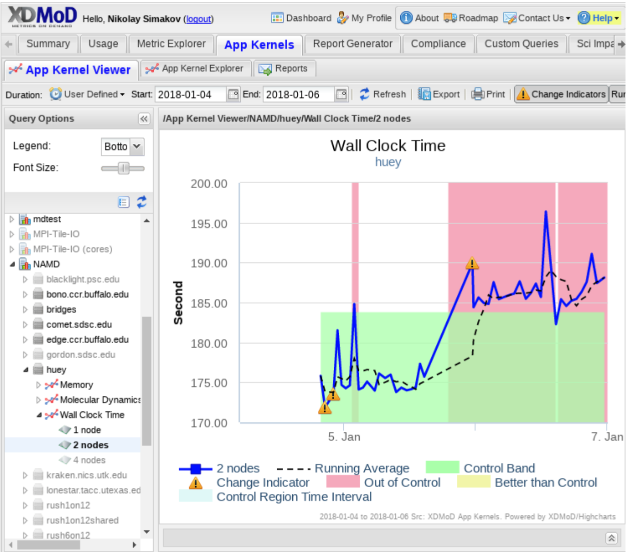
Duration: User Defined Start: 2018-01-04 End: 2018-01-06
/App Kernel Viewer/NAMD/huey/Wall Clock Time/2 nodes
Legend: Botto
2018-01-04 to 2018-01-06 Src: XDMoD App Kernels. Powered by XDMoD/Highcharts

Fig. 1. Screen-shot of the application kernel performance monitoring module of XDMod showing the change in performance of NAMD executed on 2-nodes. The module automatically calculates control regions and performs an automatic detection of performance degradation and application environment changes.

We gratefully acknowledge the support of NSF awards OCI 1025159, ACI 1445806, and OCI 1203560.

The XD Metrics on Demand (XDMoD) tool, which is designed for the comprehensive management of HPC systems, provides users, managers, and operations staff with access to utilization data, job and system level performance data, and quality of service data for HPC resources [5]. Originally developed to provide an independent audit capability for the XSEDE program, XDMoD was later open-sourced and is widely used by university, government, and industry HPC centers [6]. The application kernel performance monitoring module of XDMoD [4] allows automatic performance monitoring of HPC resources through the periodic execution of application kernels, which are based on benchmarks or real-world applications implemented with sensible input parameters (see Fig. 1 for a web-based screen-shot).

Since the application kernels, which are computationally lightweight, are designed to run continuously on a given HPC system, they are ideal for detecting differences in application performance when system wide changes (hardware or software) are made. Accordingly, XDMoD's application kernels were employed here without modification to determine if the software patches that mitigate the Meltdown and Spectre vulnerabilities significantly impact performance.

II. Related Work

There are a large number of articles published on-line related to the performance impacts of Meltdown and Spectre fixes. A large portion of them are commentary and analysis based solely on vendors press-releases without further work. The vendors press-release most often presents only a summary on impact levels, without detailed information on individual benchmarks or applications [3, 7, 8]. There were also a large number of individuals which performed some tests and published them as a comments on forums and blogs. Among independent work major attention was given to performance of single computer and consumer level software [9–12].

In this work, we present a detailed analysis of the performance impact on several benchmarks and real-word applications with a focus on the HPC environment and scientific computation.

III. Methods

A. Selected Application Kernels

The following XDMoD application kernels were chosen for this test: NAMD [13], NWChem [14], GAMESS [15, 16], ENZO [17, 18], HPC Challenge Benchmark suite (HPCC) [19] (which includes memory bandwidth micro-benchmark STREAM [20] and the NASA parallel benchmarks (NPB)[21]), interconnect/MPI benchmarks (IMB) [22, 23], IOR [24] and MDTest [25]. The first two are based on widely used scientific applications and the others are based on commonly deployed benchmarks. The application kernels were executed on one or two nodes with 8 and 16 cores respectively. Some scaling up to 8 nodes was also done. For more details on application kernels refer to [4]. Application kernel input files can be found at https://github.com/ubccr/akrr/tree/master/akrr/appker_repo/inputs.

IOR and MDTest were executed on the parallel file system (GPFS) as well as the local file system. In order to differentiate between the two file systems, we use a ".local" suffix in the reported results when the local file system is used (e.g. IOR.local).

B. System

The benchmarks were performed on two clusters with similar hardware at the Center for Computational Research (CCR), SUNY, University at Buffalo. One is a test cluster, a few nodes separated from production cluster, for developmental purposes, and the other is the production cluster itself. The compute nodes are eight-core nodes with two Intel L5520 CPUs and 24GiB RAM, connected by QDR Mellanox Infiniband. The nodes have access to a 3 PB IBM GPFS storage system shared with other HPC resources in CCR. The operating system was CentOS Linux - 7.4.1708.

The application kernels were executed on different time-frames. On the test cluster, application kernel were executed multiple times on within a few days before and after the update. On the production cluster, we are using results from our continuous performance monitoring project, where we run application kernels on a daily basis. The data presented here for the production cluster was acquired about a month before patch application and two months after.

C. Patches

To address the Meltdown and Spectre vulnerabilities a new kernel and linux-firmware were installed following Red Hat Security Advisory [26, 27]. Specifically, kernel-3.10.0-693.5.2.el7.x86_64 was updated with kernel-3.10.0-693.11.6.el7.x86_64. The updates fixes CVE-2017-5753 (Spectre Variant 1), CVE-2017-5715 (Spectre Variant 2) and CVE-2017-5754 (Meltdown Variant 3) vulnerabilities.

D. Comparison of the Results

The tests were run prior to and after application of the vulnerability updates. The "before" tests include approximately 20 runs for most of the application kernels. The "after" tests include approximately 50 runs for most application kernels. The comparison of before and after distributions were evaluated using the Wilcoxon (Mann-Whitney) two sample, two sided, test with α parameter equal to 0.05. That is, we consider the means of two distributions to be different if the probability that such test results could be obtained from equal distributions is less than or equal to 0.05.

IV. Results and Discussion

Table I and Fig. 2 show the change in walltime before and after the patches for the suite of application kernels employed in this study. For the compute intensive applications (NAMD, NWChem and HPCC), the performance degradation is around 2-3% for parallel single node jobs. However it increases to 5-11% for the case of two nodes.

IOR and MDTest benchmarks measure the performance of the file system. As discussed in the introduction, we tested

TABLE I

CHANGE IN WALLTIME UPON PATCH APPLICATION.

Application	Number of Nodes	Difference, %[1]	Are the means different?[2]	Before Patch Application			After Patch Application		
				Mean, Seconds	Standard Deviation, Seconds	Number of Runs	Mean, Seconds	Standard Deviation, Seconds	Number of Runs
System: Test Cluster									
NAMD	1	3.3	Y	306.6	1.44	24	316.8	3.05	51
NAMD	2	6.9	Y	175.4	2.78	22	187.6	2.91	50
NWChem	1	2.6	Y	77.8	1.91	23	79.8	1.13	54
NWChem	2	11.4	Y	58.4	1.05	21	65.1	4.38	50
HPCC	1	2.2	Y	304.1	6.39	23	310.8	4.86	51
HPCC	2	5.3	Y	345.1	5.41	22	363.4	8.16	50
IMB	2	4.1	Y	14.8	0.54	21	15.4	1.47	50
IOR	1	3.7	Y	188.5	9.41	21	195.4	11.78	51
IOR	2	0.7	N	371.1	12.23	22	373.8	16.42	50
IOR.local	1	2	N	462.8	16.37	12	471.8	19.12	50
MDTest	1	22.9	Y	30.5	3.17	21	37.5	4.00	50
MDTest	2	9.1	Y	166.7	3.60	23	181.9	2.92	50
MDTest.local	1	68	Y	3.8	0.62	12	6.3	2.40	50
System: Production Cluster									
NAMD	1	3.1	Y	308.6	2.94	22	318.1	4.42	102
NAMD	2	5.7	Y	179.5	3.93	17	189.8	3.98	67
NAMD	4	11.3	Y	106.1	1.93	11	118.1	4.05	44
NAMD	8	15	Y	76.2	1.42	7	87.6	3.34	35
NWChem	1	4.1	Y	78.0	1.08	25	81.2	1.88	83
NWChem	2	9.8	Y	60.5	1.63	20	66.5	3.33	84
NWChem	4	21.6	Y	41.0	1.23	14	49.8	8.61	79
NWChem	8	27.2	Y	32.2	2.30	15	40.9	4.37	102
GAMESS	1	1	Y	286.4	1.75	23	289.4	1.88	73
GAMESS	2	5.7	Y	171.7	43.09	19	181.5	55.45	50
GAMESS	4	-6.7	Y	102.7	46.45	16	95.8	18.14	38
GAMESS	8	-35.7	N	104.9	122.19	16	67.4	38.13	26
ENZO	1	2.1	Y	4863.7	106.67	21	4964.2	100.78	57
ENZO	2	7.9	Y	3378.6	119.99	22	3644.8	144.55	44
ENZO	4	11.2	Y	2305.2	229.46	17	2562.6	216.20	37
ENZO	8	27.8	Y	1893.6	101.62	12	2419.2	388.65	27

[1] Differences are calculated as the new mean value minus the old mean value divided by the average of the two means. A larger difference indicates poorer performance after the patch.

[2] The Wilcoxon (Mann-Whitney) two sample, two sided, test with $\alpha = 0.05$ was used to determine if mean values before and after are statistically different.

both the parallel and local file systems. Tables SI.1 and SI.1 show selected results for these tests. In both cases there is a significant decrease in performance for file meta-data operations (10-20%). However, the performance degradation for read and write operations is only in the range of 0-3%. Based on these findings, the performance degradation should be smaller for applications that use a small number of large files versus those that use a large number of small files. Data processing applications may therefore be particularly sensitive to the patches employed to mitigate the vulnerabilities.

The IMB test shows that most reported metrics are degraded by more than by 2% (See Supplementary Information).

The HPCC benchmark performs various tests from linear algebra, fast Fourier transformation (FFT) and memory manipulation. Interestingly the simple arrays manipulations (STREAM tests: arrays addition, copying and scaling) are actually faster in the case of two nodes (See Supplementary Information). However FFT, matrix manipulation and matrix transposition get slower. The surprising performance improvement in STREAM tests might be due to other changes in the kernel. Regardless of the cause, this improvement does not transfer to matrix manipulation and matrix transposition, which are 2% and 10% slower (on two nodes).

The performance degradation of real applications NAMD, NWChem, ENZO and GAMESS as a function of the number of nodes is shown in Figure 3. Detailed metrics are shown in Table 5 and 6 of Supplemented Information. The mean wall time for GAMESS decreased however this is due to a large number of outliers and the median value shows a small increase. Other applications namely NAMD, NWChem and ENZO do exhibit a significant degradation of their performance as the number of nodes increases. That is, more parallel, multi-node jobs suffer a greater performance degradation and the patches strongly affect the parallel scaling of these applications. Because most parallel applications rely on high-abstraction communication libraries like MPI, a KPTI friendly implementation may partially help some applications to regain their performance.

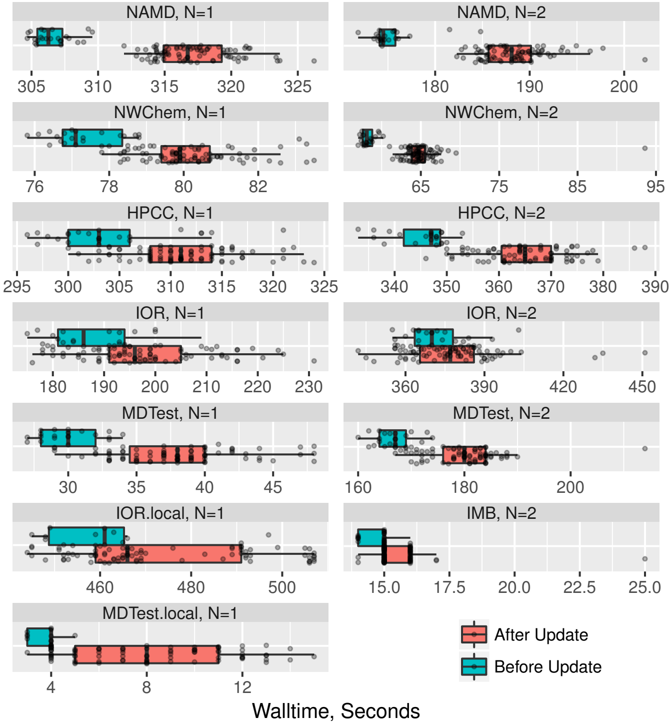

Fig. 2. Application kernel walltime comparisons before and after the updates. Box plot diagrams are used to show sample statistics. The left side of the box, the vertical line within the box and the right side of the box show first quartile, median and third quartile. In addition all measurements are plotted using round points.

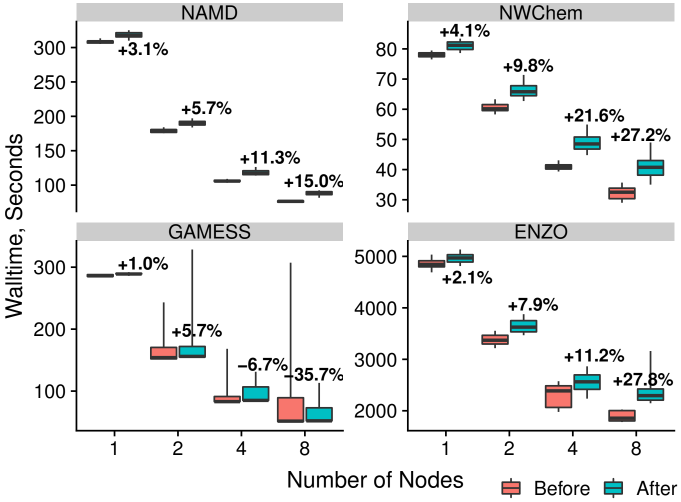

Fig. 3. Performance change of real-world application based Application kernels on the UB-HPC cluster, as measured during continuous application kernel performance monitoring. The data labels show the percent difference of the mean wall time before and after patch application.

V. Conclusions

In addition to continuous HPC system performance monitoring for quality of service, the application kernel module of XDMoD also allows rapid benchmarking of HPC systems to identify the impact of a particular change on the system.

The studied security patches have a significant degrading effect on multiple metrics, most notably MPI calls, memory copying and file metadata operations. Many other metrics show little to no change.

Overall, scientific applications executed on a single node have a moderate decrease in the performance around 2-4%. However, the performance degradation grows with node count reaching 27% in one case. This most likely is caused by increasing the number of network related system calls. Hopefully, this can be addressed in the compiler and MPI libraries.

VI. Acknowledgements

We gratefully acknowledge the support of NSF awards OCI 1025159, ACI 1445806, and OCI 1203560.

VII. Supplemental Information

Supplemental information is available at https://github.com/HPCSYSPROS/Workshop18/tree/master/Studying_Effects_of_Meltdown_and_Spectre_Patches_on_the_Performance_of_HPC_Applications_Using_Application_Kernel_Module_of_XDMoD

References

1. Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Mangard, S., Kocher, P., Genkin, D., Yarom, Y. & Hamburg, M. Meltdown. *ArXiv e-prints.* arXiv: 1801.01207 (Jan. 2018).
2. Kocher, P., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M. & Yarom, Y. Spectre Attacks: Exploiting Speculative Execution. *ArXiv e-prints.* arXiv: 1801.01203 (Jan. 2018).
3. *Speculative Execution Exploit Performance Impacts - Describing the performance impacts to security patches for CVE-2017-5754 CVE-2017-5753 and CVE-2017-5715* https://access.redhat.com/articles/3307751. Retrieved 09-21-2018.
4. Simakov, N. A., White, J. P., DeLeon, R. L., Ghadersohi, A., Furlani, T. R., Jones, M. D., Gallo, S. M. & Patra, A. K. Application kernels: HPC resources performance monitoring and variance analysis. *Concurrency and Computation: Practice and Experience* **27.** CPE-14-0402.R1, 5238–5260. ISSN: 1532-0634 (2015).
5. Furlani, T. R., Schneider, B. I., Jones, M. D., Towns, J., Hart, D. L., Gallo, S. M., DeLeon, R. L., Lu, C., Ghadersohi, A., Gentner, R. J., Patra, A. K., Laszewski, G., Wang, F., Palmer, J. T. & Simakov, N. *Using XDMoD to facilitate XSEDE operations, planning and analysis* in *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery (XSEDE '13)* (2013), 8. doi:10.1145/2484762.2484763.

6. Palmer, J. T., Gallo, S. M., Furlani, T. R., Jones, M. D., DeLeon, R. L., White, J. P., Simakov, N., Patra, A. K., Sperhac, J. M., Yearke, T., Rathsam, R., Innus, M., Cornelius, C. D., Browne, J. C., Barth, W. L. & Evans, R. T. Open XDMoD: A tool for the comprehensive management of high-performance computing resources. *Computing in Science and Engineering* **17,** 52–62 (2015).
7. Myerson, T. *Understanding the performance impact of Spectre and Meltdown mitigations on Windows Systems* https://cloudblogs.microsoft.com/microsoftsecure/2018/01/09/understanding-the-performance-impact-of-spectre-and-meltdown-mitigations-on-windows-systems/. Retrieved 09-21-2018.
8. Hachman, M. *Microsoft tests show Spectre patches drag down performance on older PCs* https://www.pcworld.com/article/3245742/components-processors/microsoft-tests-show-spectre-patches-drag-down-performance-on-older-pcs.html. Retrieved 09-21-2018.
9. Ung, G. M. *Here's how much the Meltdown and Spectre patches drag down older hardware* https://www.pcworld.com/article/3250645/laptop-computers/how-meltdown-and-spectre-patches-drag-down-older-hardware.html. Retrieved 09-21-2018.
10. Ganesh, T. S. *Meltdown Spectre: Analyzing Performance Impacts on Intel's NUC7i7BNH* https://www.anandtech.com/show/12566/analyzing-meltdown-spectre-perf-impact-on-intel-nuc7i7bnh. Retrieved 09-21-2018.
11. Walton, S. *Testing Windows 10 Performance Before and After the Meltdown Flaw Emergency Patch* https://www.techspot.com/article/1554-meltdown-flaw-cpu-performance-windows/. Retrieved 09-21-2018.
12. Brant, T. *Tests Show Tiny PC Performance Hit From Meltdown, Spectre Fix* https://www.pcmag.com/news/358589/tests-show-tiny-pc-performance-hit-from-meltdown-spectre-fi. Retrieved 09-21-2018.
13. J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale & K. Schulten. Scalable molecular dynamics with NAMD. *J. Comp. Chem.* **26,** 1781–1802 (2005).
14. M. Valiev, E. J. Bylaska, N. Govind, N. K. Kowalski, T. P. Straatsma, Dam, H. J. J., D. Wang, J. Nieplocha, E. Apra, T. L. Windus & Jong, W. A. NWChem: a comprehensive and scalable open-source solution for large scale molecular simulations. *Comput. Phys. Commun.* **181,** 1477 (2010).
15. M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis & J. A. Montgomery. General Atomic and Molecular Electronic Structure System. *J. Comp. Chem.* **14,** 347–1363 (1993).
16. M. S. Gordon & M. W. Schmidt. in *Theory and Applications of Computational Chemistry: the first forty years* (eds C. E. Dykstra, G. Frenking, K. S. Kim & G. E. Scuseria) 1167–1189 (Elsevier, Amsterdam, 2005).
17. Norman, M. L., Bryan, G. L., Harkness, R., Bordner, J., Reynolds, D., O'Shea, B. & Wagner, R. Simulating Cosmological Evolution with Enzo. *ArXiv e-prints.* arXiv: 0705.1556 (May 2007).
18. O'Shea, B. W., Bryan, G., Bordner, J., Norman, M. L., Abel, T., Harkness, R. & Kritsuk, A. Introducing Enzo, an AMR Cosmology Application. *ArXiv Astrophysics e-prints.* eprint: astro-ph/0403044 (Mar. 2004).
19. P. Luszczek, J. Dongarra & *et al. Introduction to the HPC Challenge Benchmark Suite, ICL Technical Report ICL-UT-05-01, University of Tennessee - Knoxville* 2005.
20. J. D. Mccalpin. *Memory Bandwidth and Machine Balance in Current High Performance Computers* IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter. Nov. 1995.
21. *http://www.nas.nasa.gov/publications/npb.html*
22. OSU Micro Benchmarks 3.3. *http://mvapich.cse.ohio-state.edu/benchmarks/ [September 1, 2011]*
23. Intel MPI Benchmarks 3.2.2. *http://software.intel.com/en-us/articles/intel-mpi-benchmarks [September 1, 2011]*
24. IOR Parallel I/O Benchmark. *http://sourceforge.net/projects/ior-sio [December 1, 2011]*
25. mdtest. *an MPI-coordinated metadata benchmark test https://sourceforge.net/projects/mdtest/ [June 30, 2016].*
26. *Red Hat Security Advisory: RHSA-2018:0007* https://access.redhat.com/errata/RHSA-2018:0007. Retrieved 09-21-2018.
27. *Red Hat Security Advisory: RHSA-2018:0014* https://access.redhat.com/errata/RHSA-2018:0014. Retrieved 09-21-2018.