To what extent does Kernel Page Table Isolation affect performance in systems powered by Intel CPUs?

Computer Science Extended Essay

Word Count: 3998

June-Kyoo Park

Supervisor: **REDACTED PERSONAL INFO**

Table of Contents

Introduction

The central processing unit (CPU) is a critical component of a computer which handles most of the processing and execution of instructions.[1] Thus, the performance of the CPU is very crucial to many businesses and individuals that rely on computers to carry out tasks such as Google processing 2.3 million search queries per minute.[2]

In January 2018, a major security vulnerability dubbed Meltdown was revealed to the public, which allows data to be read from the computer in situations where it was thought to be impossible.[3] This means an attacker can access sensitive data such as passwords or confidential files, which poses huge threats for any computer user.

The revelation is very significant as the Meltdown vulnerability affects every Intel CPU that executes processes out of order, which is almost every processor since 1995.[4] This means that a very wide range of devices are affected, from servers of multinational companies to an average user's laptop. The vulnerability gets its name because it "[...]"melts" the security boundaries normally enforced by hardware.".[5] The attack is analogous to a teacher hiding the answer sheet to a test in a drawer, but during the test, when a student shouts "Hey, A is the correct choice for Question 1!", the teacher gets caught off-guard and accidentally reply "Yes!", before punishing the student for talking during the test.

[1] Joel Hruska on January 10, 2018 at 9:07 am Comment. (2018, January 10). What is Speculative Execution? Retrieved February 20, 2018, from https://www.extremetech.com/computing/261792-what-is-speculative-execution

[2] D'Onfro, J. (2016, March 27). Here's a reminder of just how huge Google search truly is. Retrieved February 20, 2018, from http://www.businessinsider.com/google-search-engine-facts-2016-3#first-a-trip-down-memory-lane-heres-what-googles-search-page-looked-like-back-in-1997-1

[3] CVE-2017-5754 Detail. (2018, April 1). Retrieved March 3, 2018, from https://nvd.nist.gov/vuln/detail/CVE-2017-5754

[4] Meltdown and Spectre. (2018, January). Retrieved January 14, 2018, from https://meltdownattack.com/#faq-systems-meltdown

[5] Meltdown and Spectre. (2018, January). Retrieved January 14, 2018, from https://meltdownattack.com

To mitigate the vulnerability, developers were quick to release operating system patches that all work around the same principle of Kernel Page Table Isolation (KPTI).[6]

However, there were numerous reports that varied in the extent of the affect in performance of the patch,[7] which is why this essay will test different tasks in multiple systems to explore the effect of the KPTI patch and evaluate whether the patch incurs unwanted side effects on performance.

[6] Meltdown and Spectre. (2018, January). Retrieved January 14, 2018, from https://meltdownattack.com/#faq-fix

[7] Bright, P. (2018, January 11). Here's how, and why, the Spectre and Meltdown patches will hurt performance. Retrieved January 14, 2018, from https://arstechnica.com/gadgets/2018/01/heres-how-and-why-the-spectre-and-meltdown-patches-will-hurt-performance/

Virtual Memory

Random access memory (RAM) is utilized on modern computers. RAM is a type of volatile memory which stores data randomly, allowing the CPU to retrieve files much faster than from storage devices such as hard disk drives (HDD) or solid state drives (SSD).[8] The cost of these devices go higher as transfer speed and storage space increases, which makes it economically infeasible to make RAM sizes as large as storage disk sizes.[9]

Thus, modern memory utilizes a technique called virtual memory, which provides a layer of abstraction between the allocation of memory to the software and the physical allocation on the memory device itself.[10] This allows applications to use even more memory than what it physically has through techniques such as paging, which utilizes the computer's hard disk in addition to the RAM. For example, a small Excel Spreadsheet takes up 10MB in the RAM. As the number of processes grows, the RAM will begin to be filled. When the RAM is full, it will utilize the paging technique and use the storage devices such as the HDD as "RAM". Thus, each process has a virtual memory address that is linked with the physical memory address on the RAM.[11]

When a process is being used, the virtual memory address will be translated to the physical memory address.[12]

Since translating the virtual addresses to physical addresses is a common operation, modern CPUs have a device called the Translation Lookaside Buffer (TLB) that will cache recently used page table

[8] What is RAM (random access memory)? - Definition from WhatIs.com. (n.d.). Retrieved February 20, 2018, from http://searchstorage.techtarget.com/definition/RAM-random-access-memory

[9] Hannon, T. (n.d.). What's the difference between computer memory (RAM) and hard drive storage? Retrieved February 20, 2018, from https://soundsupport.biz/2012/05/06/whats-the-difference-between-computer-memory-ram-and-hard-drive-storage/

[10] What is virtual memory? - Definition from WhatIs.com. (n.d.). Retrieved February 20, 2018, from http://searchstorage.techtarget.com/definition/virtual-memory

[11] What is virtual memory? - Definition from WhatIs.com. (n.d.). Retrieved February 20, 2018, from http://searchstorage.techtarget.com/definition/virtual-memory

[12] What is virtual memory? - Definition from WhatIs.com. (n.d.). Retrieved February 20, 2018, from http://searchstorage.techtarget.com/definition/virtual-memory

entries.[13] Page tables are data structures that store sets of mapping between virtual and physical memory addresses (memory pages - 4096 bytes for Intel CPUs).[14] It will also contain metadata such as permissions and Address Space Identifiers (ASID). ASID is an extra ID attributed to each TLB entry to so that a process switch only switches the ASID and does not flush the whole TLB.[15] Storing page tables in the TLB allows the system to utilize its CPU to perform the translation in hardware instead of the OS doing it in software, which saves time.[16]

When a process fetches a virtual address, the CPU will first access the TLB to retrieve the cached physical address. If not found in the TLB, it will try to find it in the set of page tables in the CR3 register within the CPU.[17] CR3 refers to control register 3; a register is location that can be accessed quickly by the CPU and a control register controls the actions of the CPU.[18] If a page table containing the mapping is found, it will be cached in the TLB and will be used for translation. If a page table containing the mapping is not found, a page fault will be raised to the OS, and the page fault handler will decide whether a new physical memory address will be mapped to the process or will stop the process.[19] This will slow down the processes, as the OS is taking care of the memory pages, not the hardware.

[13] Paging: Faster Translations (TLBs). (n.d.). Retrieved February 20, 2018, from http://pages.cs.wisc.edu/~remzi/OSTEP/vm-tlbs.pdf

[14] Page Table Entries. (n.d.). Retrieved February 20, 2018, from http://courses.ics.hawaii.edu/ReviewICS332/morea/090_VirtualMemory/ics332_virtualmemory1.pdf

[15] Computer Science from the Bottom Up. (n.d.). Retrieved February 20, 2018, from https://www.bottomupcs.com/csbu.pdf

[16] Paging: Faster Translations (TLBs). (n.d.). Retrieved February 20, 2018, from http://pages.cs.wisc.edu/~remzi/OSTEP/vm-tlbs.pdf

[17] CPU Registers x86. (n.d.). Retrieved February 20, 2018, from https://wiki.osdev.org/CPU_Registers_x86#CR3

[18] Adams, T., & Hughes, M. C. (2018, February 28). What Is a Control Register? Retrieved March 3, 2018, from http://www.wisegeek.com/what-is-a-control-register.htm

[19] Page Table Entries. (n.d.). Retrieved February 20, 2018, from http://courses.ics.hawaii.edu/ReviewICS332/morea/090_VirtualMemory/ics332_virtualmemory1.pdf

Kernel Memory

Kernel memory is memory used by the OS's kernel. The kernel is the core of the system, and acts as a layer that connects the software to the hardware.[20] It will control drivers, translate memory addresses, and most other things in the system. Normal user processes (user mode) should not be able to access the kernel memory as it has stores sensitive data such as passwords.[21]

User Mode and Kernel Mode

Each virtual mapping or page, will contain metadata such as permissions.[22] Thus, **kernel memory is not accessible when in user mode.** If a process attempts to do so, it will trigger a page fault. If the process is running in kernel mode such as during a system call (syscall), it is allowed to access kernel memory.[23] A syscall is a request made to the kernel by a process.[24] Input/Output (I/O) operations would be an example that utilizes a lot of syscalls.[25]

Dual Mapping

If the kernel and user memory are on separate spaces (separate modes), every syscall will require a switch between the two spaces which causes page faults.[26] Since this is very slow, modern OSs will map the kernel memory directly into a process so that each process can directly access the kernel memory when needed without requiring a page table switch.[27] This prevents causing page faults for every syscall which saves valuable CPU power.

[20] Beal, V. (n.d.). What is Kernel. Retrieved from https://www.webopedia.com/TERM/K/kernel.html

[21] Diesburg, S. (n.d.). Memory Protection: Kernel and User Address Spaces. Retrieved from http://www.cs.uni.edu/~diesburg/courses/cs3430_sp14/sessions/s12/s12_memory_protection.pdf

[22] Saelee, M. (n.d.). Virtual Memory. Retrieved from http://moss.cs.iit.edu/cs351/slides/slides-vm.pdf

[23] Virtual Memory. (n.d.). Retrieved from http://csapp.cs.cmu.edu/2e/ch9-preview.pdf

[24] Virtual Memory. (n.d.). Retrieved from http://csapp.cs.cmu.edu/2e/ch9-preview.pdf

[25] CS360 Lecture notes -- Introduction to System Calls (I/O System Calls). (n.d.). Retrieved from http://web.eecs.utk.edu/~huangj/cs360/360/notes/Syscall-Intro/lecture.html

[26] Saelee, M. (n.d.). Virtual Memory. Retrieved from http://moss.cs.iit.edu/cs351/slides/slides-vm.pdf

[27] N. (n.d.). Virtual address spaces. Retrieved from https://docs.microsoft.com/en-us/windows-

Speculative Execution

Due to limitations in increasing memory speed to match CPU speeds,[28] if instruction sets are executed sequentially in a CPU, the time it takes to wait for memory to load the data from the previous instruction set gets too long, causing CPU throughput to be limited by the speed of the memory. CPU throughput is the amount of work the CPU can perform in a given period of time.[29] In order to work around the slow memory speeds, modern CPUs execute instruction sets not necessarily sequentially every time, but also out of order and speculatively.[30] That is, before the CPU is done with processing a single instruction set, another instruction set may be predicted and executed within the CPU.[31]

As mentioned before, modern OSs utilize dual mappings, where the kernel memory and user memory are on the same page table to speed up syscalls. When a user mode process tries to access kernel memory, Intel CPUs speculatively execute the access of the kernel memory even though it wasn't privileged to do so, which is a hardware fault that CPU architects failed to block.[32]

Cache Side-Channel Attack

As cache speeds are much faster than standard memory, if a particular data is accessed much quicker than other ones, it is assumed that that particular data was residing in the cache.[33] A cache

hardware/drivers/gettingstarted/virtual-address-spaces

[28] Carvalho, C. (n.d.). The Gap between Processor and Memory Speeds. Retrieved from https://pdfs.semanticscholar.org/6ebe/c8701893a6770eb0e19a0d4a732852c86256.pdf

[29] Measuring Instruction Latency and Throughput. (2008, October 20). Retrieved from https://software.intel.com/en-us/articles/measuring-instruction-latency-and-throughput

[30] Mutlu, O. (n.d.). *Lecture 12. Out of Order Execution - Carnegie Mellon - Comp. Arch. 2015*. Lecture. Retrieved from https://www.youtube.com/watch?v=P-mXr9adbCc&list=PL5PHm2jkkXmi5CxxI7b3JCL1TWybTDtKq&index=14

[31] Mutlu, O. (n.d.). *Lecture 12. Out of Order Execution - Carnegie Mellon - Comp. Arch. 2015*. Lecture. Retrieved from https://www.youtube.com/watch?v=P-mXr9adbCc&list=PL5PHm2jkkXmi5CxxI7b3JCL1TWybTDtKq&index=14

[32] Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Mangard, S., . . . Hamburg, M. (2018). Meltdown. *Meltdown*. Retrieved from https://meltdownattack.com/meltdown.pdf.

[33] Yarom, Y., & Falkner, K. (n.d.). FLUSH RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack. Retrieved from https://eprint.iacr.org/2013/448.pdf.

side-channel attack refers to an attacker compromising a system by analyzing the time taken to access certain data.[34]

An example would be the Flush+Reload cache side-channel attack. This side-channel attack works by flushing the contents of the cache, and then measuring the time taken to reload the data (hence the name Flush+Reload).[35] If the reloading of data took a very short time, it means this particular data was reloaded into the cache by another process after the flushing of the cache.[36] The Meltdown Attack utilizes this particular attack.[37]

Meltdown Attack

How it works:[38]

1. Allocate Array of size 256*4096 (256 pages since 4096 bytes is size of one page)
2. Flush cache
3. Read 1 byte from Kernel Memory
4. Multiply the byte from Kernel Memory by 4096 (page size, so that one byte takes up one page) and store it in Array
5. Iterate through Array (256 pages) and measure time taken to load each page
6. The location (0 to 255) of the page that loaded the fastest is the value of the byte from kernel memory that was read in Step 3.

[34] Yarom, Y., & Falkner, K. (n.d.). FLUSH RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack. Retrieved from https://eprint.iacr.org/2013/448.pdf.

[35] Yarom, Y., & Falkner, K. (n.d.). FLUSH RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack. Retrieved from https://eprint.iacr.org/2013/448.pdf.

[36] Yarom, Y., & Falkner, K. (n.d.). FLUSH RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack. Retrieved from https://eprint.iacr.org/2013/448.pdf.

[37] Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Mangard, S., . . . Hamburg, M. (2018). Meltdown. *Meltdown*. Retrieved from https://meltdownattack.com/meltdown.pdf.

[38] Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Mangard, S., . . . Hamburg, M. (2018). Meltdown. *Meltdown*. Retrieved from https://meltdownattack.com/meltdown.pdf.

Step 3 should not be possible, as the process is running in User space and has not been authorized to access kernel memory. At step 3, the attacker does not know the value of this byte. Nevertheless, in Intel CPUs, this instruction will be executed speculatively, as mentioned in the Speculative Execution section. After the CPU realizes it has made a mistake (raise an exception), it will attempt to roll back the executions done in Step 3 and 4. However, the cache is not flushed when this happens, which allows the attacker to utilize the Flush+Reload attack in Step 5 and 6 to determine the value of the byte from kernel memory.

Due to step 2, only the one page that contains Kernel Memory data in step 4 will be cached; the other 255 pages in the array will not be cached, which makes Step 6 possible.

The attacker can iterate through Step 1 to 6 as many times as they want until they get all the data they need.

Current Mitigation

The most straightforward and current mitigation is to separate the User memory and Kernel memory for each process, so that a process running in User mode is not able to speculatively execute a process that requires access to the Kernel memory (Making Step 2 from the previous section impossible). This is called Kernel Page Table Isolation, or KPTI. Since January 2018, operating systems such as Linux, Windows, and OS X have been patched to implement some form of KPTI.

Performance Degradation

After the patch, when a process normally running in User mode executes a syscall, a page fault will occur because the kernel memory now exists in a separate page table. As mentioned earlier, a page fault will hinder performance as the memory translation is carried on in the OS software, not in the CPU hardware.

A page fault tells the CPU that the entries in the TLB did not contain what the process was looking for, which causes the CPU to invalidate the data in the TLB, resulting in a TLB flush. This means that every execution that attempts to access kernel memory will have to flush the TLB twice - once when switching to the page tables of the kernel memory, and once when switching back to the page tables of the memory in user mode.

Experimentation

Testing the impact of the KPTI patch:

In order to effectively compare the performance difference of systems before and after the patch, the components of the system are kept constant throughout: 4GB RAM, 250GB SSD, and the Intel Pentium E6800 processor. SSD is the acronym for solid state drive which utilizes flash memory, which is much faster than the traditional magnetic hard disk drives (HDD). The SATA SSD was the fastest drive available to reduce bottlenecks from a hard disk while running the benchmarking programs.

Controlling these variables will ensure that the performance change is only caused by the KPTI patch, and will make comparing results more valid - the operating system being used will be Windows 10 Pro version 10.0.16299, with and without the KPTI Patch (Windows Update KB4056892).[39]

Method of Testing:

The system will be tested using 3 benchmarking software. Since the KPTI patch interferes with how components in the CPU translates virtual memory addresses to physical memory addresses, CPU throughput with and without the patch will be tested, using the PassMark PerformanceTest.

The CPU Test in this benchmarking software tests mathematical operations, compression, encryption, and physics.[40]

To effectively test how systems utilizing intensive I/O (syscalls) such as servers are affected, CrystalDiskMark software was used.

[39] January 3, 2018—KB4056892 (OS Build 16299.192). (2018, January 3). Retrieved from https://support.microsoft.com/en-us/help/4056892/windows-10-update-kb4056892

[40] PassMark Rating and comparable baselines. (n.d.). Retrieved from https://www.passmark.com/products/pt.htm

CrystalDiskMark measures disk read and write speeds, and will allows to test operations with varying queue depths (QD).[41] QD is the number of I/O instructions that can be handled at each time by the host bus adapter (HBA).[42] The HBA is an adapter that connects the SSD to the computer.[43] To measure performance impact for the average user, the PCMark 10 Benchmark was used. It runs a variety of tasks that is used commonly in the workplace and measures system performance during those tests (such as time taken to open a file, or average frames per second while rendering a 3D object).[44]

Hypothesis

Tasks that do not require lots of syscalls such as integer calculation or floating point calculations should not be affected by the KPTI patch. Tasks that have a lot of I/O such as reading and writing files to secondary memory will be slowed significantly by the KPTI patch. Everyday usage will not be hit much as they mostly run in user space.

[41] H. (2018, February 03). CrystalDiskMark. Retrieved from https://crystalmark.info/en/software/crystaldiskmark/

[42] What is Storage Queue Depth (QD) and why is it so important? (2016, July 26). Retrieved from https://www.settlersoman.com/what-is-storage-queue-depth-qd-and-why-is-it-so-important/

[43] What is host bus adapter (HBA)? (n.d.). Retrieved from http://searchstorage.techtarget.com/definition/host-bus-adapter

[44] PCMark 10 - The Complete Benchmark. (n.d.). Retrieved from https://www.futuremark.com/benchmarks/pcmark10

Results

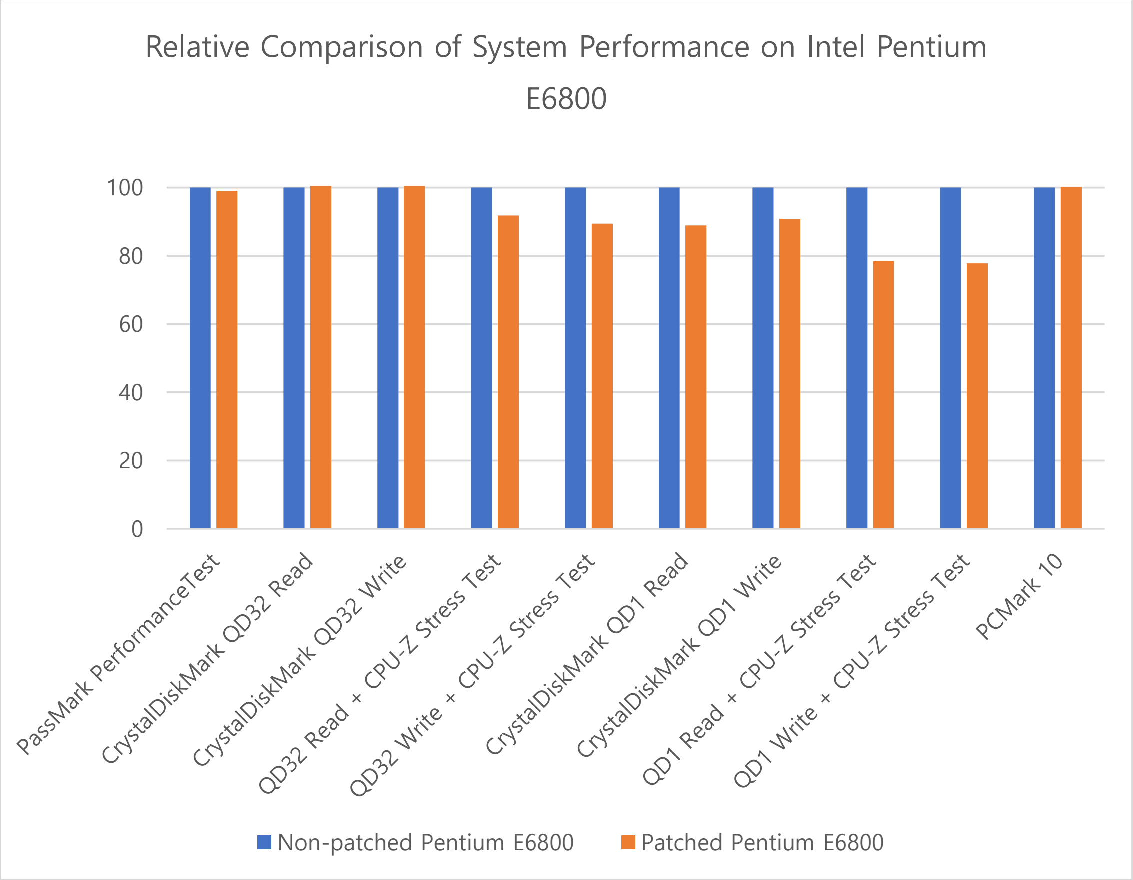
Figure 1 shows the raw data taken from the benchmarking programs with and without the patch:

Figure 1:

	Non-patched Pentium E6800	Patched Pentium E6800
PassMark PerformanceTest	2280	2258
CrystalDiskMark QD32 Read (MB/s)	42.02	42.2
CrystalDiskMark QD32 Write (MB/s)	86.74	87.18
QD32 Read + CPU-Z Stress Test (MB/s)	29.51	27.09
QD32 Write + CPU-Z Stress Test (MB/s)	34.7	31.05
CrystalDiskMark QD1 Read (MB/s)	35.98	31.99
CrystalDiskMark QD1 Write (MB/s)	52.17	47.38
QD1 Read + CPU-Z Stress Test (MB/s)	1.328	1.041
QD1 Write + CPU-Z Stress Test (MB/s)	1.318	1.025
PCMark	1673	1677

Figure 2 shows the data as a relative comparison:

Figure 2:



Conclusion of Tests

Raw CPU performance decreased by 1 percent, which is very small but still a decrease in CPU performance. However, this is probably caused by random errors that were present during the experimentation causing CPU performance to vary.

The standalone SSD benchmark **increased performance by an average 0.5%,** which completely contradicts the hypothesis that I/O intensive tasks will be impacted heavily. After running additional tests while monitoring system resources using the Intel Extreme Tuning Utility software,[45] it was

[45] Intel® Extreme Tuning Utility (Intel® XTU). (n.d.). Retrieved from https://downloadcenter.intel.com/download/24075/Intel-Extreme-Tuning-Utility-Intel-XTU-

observed that there was an average one percent increase in CPU utilization with the patch installed. The most likely reason is that the KPTI patch increased the load on the CPU while performing the disk benchmark, which increases the power state of the CPU, allowing for faster response times when switching between user and kernel memory page tables.

While the patch may have increased performance for the standalone disk benchmark, it does not reflect real-life scenarios where the CPU is utilized at full capacity without additional processing capacity in the CPU reserved for the disk I/O operations. Thus, the identical disk benchmark was conducted again, this time with a CPU stress test running in the background. If our hypothesis that the increased CPU load improved disk I/O performance is sound, this test should show a decrease in performance as the CPU will not be able to compensate for the additional load by increasing its power state since it is already running at maximum capacity.

With the patch applied and CPU utilization at maximum, **the disk read/write performance decreased by average of 9.4 percent**, which supports the observation that the CPU load increases during I/O intensive tasks.

Thus, a more intensive disk benchmark was conducted, with a lower QD. Lowering QD will lower I/O operations of the SSD as the HBA can only handle 1 I/O operation at a time compared to the high QD of 32. However, this will reduce latency from the SSD, as the HBA can now quickly handle 1 I/O operation rather than 32 I/O operations at a time. Latency is the delay in output from a system from an input.[46] Since the latency from SSD gets lower,[47] the additional workload of the CPU with the KPTI patch will affect the SSD performance more for low QD workloads than high QD workloads.

This is reflected in the **10 percent decrease in disk performance** after the KPTI patch.

With the CPU at maximum utilization, the performance degrades even further, **at 22 percent worse than the unpatched version of the OS.**

[46] Beal, V. (n.d.). Latency. Retrieved from https://www.webopedia.com/TERM/L/latency.html

[47] Norman, L. (n.d.). Latency: The Heartbeat of a Solid State Disk. Retrieved from https://www.snia.org/sites/default/education/tutorials/2010/spring/solid/LeviNorman_Latency_The_Heartbeat_SSD.pdf

The PCMark 10 benchmark showed an **increase of 0.2 percent**, which is most likely due to variance in system resources during the test.

Extension of the test

The Pentium E6800 is nearly a decade old, being released in 2010.[48] Thus, there is a need to test performance impacts on newer CPU architectures.

Intel CPUs include a technology named the Process Context Identifier (PCID) since 2010.[49] The PCID will include another metadata for the TLB which identifies the TLB entry, much like the ASID.[50] [51] Thus, when a process requests a switch from the user memory pages to kernel memory pages, instead of flushing the entire TLB and loading the kernel memory pages, the CPU simply needs to change the ID of the TLB from the user memory to kernel memory to access it.

Thus, CPUs utilizing the PCID technology should see less of a performance hit from the KPTI patch. The newer system used for this test is composed of an Intel i5-8600K CPU, 4GB RAM and the same 250GB SSD. The software used is the same; Windows version 10.0.16299, with and without the KPTI patch. The same benchmarking tools and methods were used. The Windows KPTI patch enables the PCID technology for this CPU.

[48] Intel® Pentium® Processor E6800 (2M Cache, 3.33 GHz, 1066 FSB) Product Specifications. (n.d.). Retrieved from https://ark.intel.com/products/42811/Intel-Pentium-Processor-E6800-2M-Cache-3_33-GHz-1066-FSB

[49] Westmere Arrives. (n.d.). Retrieved from https://www.realworldtech.com/westmere/

[50] PCID is now a critical performance/security feature on x86 - Google G... (2018, January 07). Retrieved from http://archive.is/ma8Iw

[51] Re: PCID and TLB flushes (was: [GIT PULL] kdbus for 4.1-rc1). (2015, April 28). Retrieved from https://lkml.org/lkml/2015/4/28/824

Results

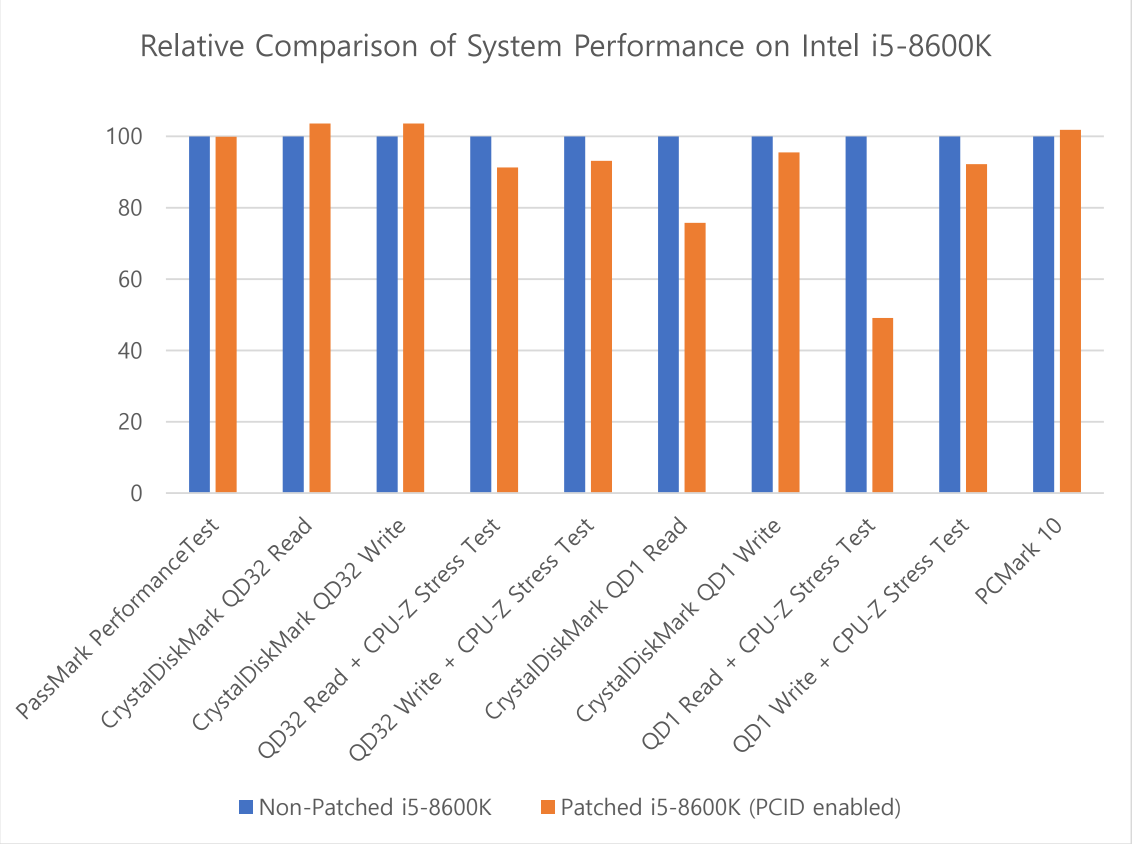
Figure 3 shows the raw data taken from the benchmarking programs with and without the patch:

Figure 3:

	Non-patched i5-8600K	Patched i5-8600K
PassMark PerformanceTest	13199	13182
CrystalDiskMark QD32 Read (MB/s)	253.13	262.3
CrystalDiskMark QD32 Write (MB/s)	231.3	239.63
QD32 Read + CPU-Z Stress Test (MB/s)	39.18	35.75
QD32 Write + CPU-Z Stress Test (MB/s)	31.05	28.92
CrystalDiskMark QD1 Read (MB/s)	50.46	38.23
CrystalDiskMark QD1 Write (MB/s)	119.1	113.7
QD1 Read + CPU-Z Stress Test (MB/s)	3.327	1.632
QD1 Write + CPU-Z Stress Test (MB/s)	2.069	1.908
PCMark	4595	4679

Figure 4 shows the data as a relative comparison:

Figure 4:



Conclusion of Testing

A similar trend as the Pentium E6800 system can be observed: The **raw CPU throughput was not affected**, while the **high queue depth disk performance increased by 4 percent.**

The high QD disk benchmark with CPU stress decreased by 7.8 percent, which is lower than the 9.4 percent decrease in the Pentium E6800 system. This is most likely due to the PCID technology allowing page table switching without flushing the entire TLB, which the Pentium E6800 system lacks.

However, for the low QD benchmarks, file reading seems to be more taxing on the CPU, exhibiting 24 percent decrease in performance without the stress test and a 51 percent decrease in performance with the CPU utilization at maximum. This is much higher than the 11 percent and 22 percent performance hit respectively on the Pentium E6800 system. This unexpected outcome may be explained by the higher performance capacity of the i5-8600K, causing the stress test to affect the disk benchmark at a higher percentage.

File writing exhibited expected results, the **standalone disk benchmark being affected by 5 percent and the addition of the CPU stress test lowering performance by 8 percent.** This is lower than the 10 and 22 percent performance decrease, respectively, on the older CPU, probably due to the utilization of the PCID technology.

The PCMark 10 benchmark showed an increase of 2 percent, which is also most likely due to variance in system resources during the test like the Pentium E6800 system.

Performance Increase After Applying the KPTI Patch

Although the increase in I/O performance was small at lower than 4 percent, it is still intriguing why this can happen. The identical experimentation was conducted 18 times, never showing a decrease in high queue depth read/write speeds. As mentioned earlier, it is likely that the CPU is being utilized more, but also because that the SSD is the bottleneck even after the KPTI patch for high QD operations.

Analysis

A short term mitigation for the performance loss is to utilize the Process-Context Identifier (PCID) in modern CPUs.

However, operating systems currently support PCID only on CPUs that also utilize a technology called Invalidate-PCID (INVPCID).[52] The reason for this is because software developers found it easier to enable PCID with INVPCID capable CPUs in the limited timeframe to release the patch, not

[52] Bright, P. (2018, January 11). Here's how, and why, the Spectre and Meltdown patches will hurt performance. Retrieved January 14, 2018, from https://arstechnica.com/gadgets/2018/01/heres-how-and-why-the-spectre-and-meltdown-patches-will-hurt-performance/

because PCID could not be enabled without INVPCID.[53] [54] [55] Thus, even though PCID has been available in CPUs for a few years already, only a few generations of them, since 2013, are able to utilize the technology on different operating systems.

It also means that businesses and individuals using CPUs that do not support PCID will have to upgrade to a newer system, which is an economical burden.

However, as shown through in the results, not having PCID may not be a problem if users were did not have any problems before the patch, as the performance did not decrease by large percentages.

Limitations of the Test

It was mentioned that the SSD could be the bottleneck, and online sources seem to support this claim. When testing with much faster NVMe SSDs, the performance was found to have decreased, in all disk benchmarks.[56]

However, it is impossible to use the NVMe SSD on the Pentium E6800, as the motherboard does not support the interface.

The Microsoft Windows operating system does not allow PCID to be disabled for the i5-8600K after the patch. If PCID can be disabled, a more accurate experimentation can be conducted in regards to how PCID mitigates the performance loss arising from the KPTI patch.

[53] Hansen, D. (n.d.). [PATCH] x86/mm/kaiser: remove no-INVPCID user ASID flushing. Retrieved from https://www.mail-archive.com/linux-kernel@vger.kernel.org/msg1546880.html

[54] [RFC 00/13] x86/mm: PCID and INVPCID. (n.d.). Retrieved from https://groups.google.com/forum/#!topic/fa.linux.kernel/IV8D8p9uR9g

[55] A. (2014, November 21). Improve Performance for Separating Kernel and User Address Space with Process-Context Identifiers (PCIDs). Retrieved from http://hypervsir.blogspot.com/2014/11/improve-performance-for-separating.html

[56] Malventano, A. (2018, January 5). Meltdown's Impact on Storage Performance - Really an Issue? Retrieved from https://www.pcper.com/news/Storage/Meltdowns-Impact-Storage-Performance-Really-Issue

Conclusion

Answering the question “To what extent does Kernel Page Table Isolation affect performance in systems powered by Intel CPUs?” depends on many factors.

For the purpose of the essay, performance was measured in terms of CPU throughput, disk read/write speeds, and combinations of different tasks.

Workloads may vary from user to user, and the age of the system matters as well, shown by the relatively low performance hits in many cases for the PCID enabled system.

As the performance hit is imperceivable for everyday usage shown through the PCMark 10 benchmark, average users should install the patch. The PCMark 10 benchmark also did not show any decrease in graphics performance, which means even some system resource intensive programs such as games can run without any drop in quality or performance.

The results show that the KPTI patch affects I/O intensive systems such as servers the most, with the worst-case scenario in the experimentation exhibiting a 51 percent reduction in disk performance. The CPU not being able to find the assigned kernel memory slows down processes first because it cannot access memory as fast as before the patch was applied, and also due to the page table isolation causing a page fault, flushing the TLB twice per syscall, which is taxing on the CPU. However, since servers are most likely to contain sensitive data for many individuals, it is recommended that they install the patch as well. The high impact on disk performance can be mitigated by optimizing software code so that the disk processes high QD workloads more often than low QD workloads. As the number of I/O operations the HBA can process is increased, there will be increased latency from the SSD, but this latency allows for less load on the CPU, which should consequently improve performance after the patch, as shown in the results above.

The reason the Meltdown vulnerability was possible is because CPU architects did not consider that out of order speculative execution could be used to access kernel memory, and only implemented the memory protection sequentially to maximize CPU performance. It is insane that such a small inconsideration during the design process can affect millions of users around the world. This calls for CPU redesigns that are not affected by Meltdown.

Future CPU designs should make sure that kernel memory should not be accessed sequentially. However, accessing kernel memory speculatively is not allowed and will cause exceptions status quo anyhow, as mentioned before. This means, it is acceptable for CPUs to speculatively access restricted memory, as long as it is not able to process the physical data in the memory.

There are many ways to mitigate the Meltdown vulnerability without the need of KPTI:

A future design could be to overwrite all values in the kernel memory to pseudo-data in case of a failed authorization. This means that even if the CPU accesses the kernel memory speculatively, the data is meaningless. This will allow the kernel memory addresses to stay on the same page tables as user memory addresses, which prevents the need for KPTI.

Or, prevention may be better than cure:

An authorization gate can be added in front of the TLB such that the kernel pages cannot be accessed sequentially or speculatively at all while in user mode.

In conclusion, the patch should be installed if sensitive data exists on a system, since it is predicted that the performance loss will gradually decrease over time as developers continue to optimize their code in response to the patch.

[Word Count : 3998]

Bibliography

A. (2014, November 21). Improve Performance for Separating Kernel and User Address Space with Process-Context Identifiers (PCIDs). Retrieved from http://hypervsir.blogspot.com/2014/11/improve-performance-for-separating.html

Adams, T., & Hughes, M. C. (2018, February 28). What Is a Control Register? Retrieved March 3, 2018, from http://www.wisegeek.com/what-is-a-control-register.htm

Beal, V. (n.d.). What is Kernel. Retrieved from https://www.webopedia.com/TERM/K/kernel.html

Beal, V. (n.d.). Latency. Retrieved from https://www.webopedia.com/TERM/L/latency.html

Bright, P. (2018, January 11). Here's how, and why, the Spectre and Meltdown patches will hurt performance. Retrieved January 14, 2018, from https://arstechnica.com/gadgets/2018/01/heres-how-and-why-the-spectre-and-meltdown-patches-will-hurt-performance/

Carvalho, C. (n.d.). The Gap between Processor and Memory Speeds. Retrieved from https://pdfs.semanticscholar.org/6ebe/c8701893a6770eb0e19a0d4a732852c86256.pdf

Computer Science from the Bottom Up. (n.d.). Retrieved February 20, 2018, from https://www.bottomupcs.com/csbu.pdf

CPU Registers x86. (n.d.). Retrieved February 20, 2018, from https://wiki.osdev.org/CPU_Registers_x86#CR3

CS360 Lecture notes -- Introduction to System Calls (I/O System Calls). (n.d.). Retrieved from http://web.eecs.utk.edu/~huangj/cs360/360/notes/Syscall-Intro/lecture.html

CVE-2017-5754 Detail. (2018, April 1). Retrieved March 3, 2018, from https://nvd.nist.gov/vuln/detail/CVE-2017-5754

Diesburg, S. (n.d.). Memory Protection: Kernel and User Address Spaces. Retrieved from http://www.cs.uni.edu/~diesburg/courses/cs3430_sp14/sessions/s12/s12_memory_protection.pdf

D'Onfro, J. (2016, March 27). Here's a reminder of just how huge Google search truly is. Retrieved February 20, 2018, from http://www.businessinsider.com/google-search-engine-facts-2016-3#first-a-trip-down-memory-lane-heres-what-googles-search-page-looked-like-back-in-1997-1

H. (2018, February 03). CrystalDiskMark. Retrieved from https://crystalmark.info/en/software/crystaldiskmark/

Hannon, T. (n.d.). What's the difference between computer memory (RAM) and hard drive storage? Retrieved February 20, 2018, from https://soundsupport.biz/2012/05/06/whats-the-difference-between-computer-memory-ram-and-hard-drive-storage/

Hansen, D. (n.d.). [PATCH] x86/mm/kaiser: remove no-INVPCID user ASID flushing. Retrieved from https://www.mail-archive.com/linux-kernel@vger.kernel.org/msg1546880.html

Intel® Pentium® Processor E6800 (2M Cache, 3.33 GHz, 1066 FSB) Product Specifications. (n.d.). Retrieved from https://ark.intel.com/products/42811/Intel-Pentium-Processor-E6800-2M-Cache-3_33-GHz-1066-FSB

Intel® Extreme Tuning Utility (Intel® XTU). (n.d.). Retrieved from https://downloadcenter.intel.com/download/24075/Intel-Extreme-Tuning-Utility-Intel-XTU-

January 3, 2018—KB4056892 (OS Build 16299.192). (2018, January 3). Retrieved from https://support.microsoft.com/en-us/help/4056892/windows-10-update-kb4056892

Joel Hruska on January 10, 2018 at 9:07 am Comment. (2018, January 10). What is Speculative Execution?

Retrieved February 20, 2018, from https://www.extremetech.com/computing/261792-what-is-speculative-execution

Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Mangard, S., . . . Hamburg, M. (2018). Meltdown. *Meltdown*. Retrieved from https://meltdownattack.com/meltdown.pdf.

Malventano, A. (2018, January 5). Meltdown's Impact on Storage Performance - Really an Issue? Retrieved from https://www.pcper.com/news/Storage/Meltdowns-Impact-Storage-Performance-Really-Issue

Measuring Instruction Latency and Throughput. (2008, October 20). Retrieved from https://software.intel.com/en-us/articles/measuring-instruction-latency-and-throughput

Meltdown and Spectre. (2018, January). Retrieved January 14, 2018, from https://meltdownattack.com/#faq-systems-meltdown

Meltdown and Spectre. (2018, January). Retrieved January 14, 2018, from https://meltdownattack.com/#faq-fix

Mutlu, O. (n.d.). *Lecture 12. Out of Order Execution - Carnegie Mellon - Comp. Arch. 2015*. Lecture. Retrieved from https://www.youtube.com/watch?v=P-mXr9adbCc&list=PL5PHm2jkkXmi5CxxI7b3JCL1TWybTDtKq&index=14

N. (n.d.). Virtual address spaces. Retrieved from https://docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/virtual-address-spaces

Norman, L. (n.d.). Latency: The Heartbeat of a Solid State Disk. Retrieved from https://www.snia.org/sites/default/education/tutorials/2010/spring/solid/LeviNorman_Latency_The_Heartbeat_SSD.pdf

Paging: Faster Translations (TLBs). (n.d.). Retrieved February 20, 2018, from http://pages.cs.wisc.edu/~remzi/OSTEP/vm-tlbs.pdf

Page Table Entries. (n.d.). Retrieved February 20, 2018, from http://courses.ics.hawaii.edu/ReviewICS332/morea/090_VirtualMemory/ics332_virtualmemory1.pdf

PassMark Rating and comparable baselines. (n.d.). Retrieved from https://www.passmark.com/products/pt.htm

PCID is now a critical performance/security feature on x86 - Google G... (2018, January 07). Retrieved from http://archive.is/ma8Iw

PCMark 10 - The Complete Benchmark. (n.d.). Retrieved from https://www.futuremark.com/benchmarks/pcmark10

Re: PCID and TLB flushes (was: [GIT PULL] kdbus for 4.1-rc1). (2015, April 28). Retrieved from https://lkml.org/lkml/2015/4/28/824

[RFC 00/13] x86/mm: PCID and INVPCID. (n.d.). Retrieved from https://groups.google.com/forum/#!topic/fa.linux.kernel/IV8D8p9uR9g

Saelee, M. (n.d.). Virtual Memory. Retrieved from http://moss.cs.iit.edu/cs351/slides/slides-vm.pdf

Virtual Memory. (n.d.). Retrieved from http://csapp.cs.cmu.edu/2e/ch9-preview.pdf

Westmere Arrives. (n.d.). Retrieved from https://www.realworldtech.com/westmere/

What is host bus adapter (HBA)? (n.d.). Retrieved from http://searchstorage.techtarget.com/definition/host-bus-adapter

What is RAM (random access memory)? - Definition from WhatIs.com. (n.d.). Retrieved February 20, 2018, from http://searchstorage.techtarget.com/definition/RAM-random-access-memory

What is Storage Queue Depth (QD) and why is it so important? (2016, July 26). Retrieved from https://www.settlersoman.com/what-is-storage-queue-depth-qd-and-why-is-it-so-important/

What is virtual memory? - Definition from WhatIs.com. (n.d.). Retrieved February 20, 2018, from http://searchstorage.techtarget.com/definition/virtual-memory

Yarom, Y., & Falkner, K. (n.d.). FLUSH RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack. Retrieved from https://eprint.iacr.org/2013/448.pdf.