CASE STUDY

Data Center Virtualization

Intel® Cache Acceleration Software for QEMU* Accelerates Virtual Machine Storage Workloads

Improve QEMU VM responsiveness up to 4.1x using Intel® Optane™ technology and Intel® Cache Acceleration Software[1]

At a Glance: Intel® CAS for QEMU...

- Accelerates virtualization workloads, reducing latency up to 4.1x[1]
- Caches read-heavy jobs
- Isolates workloads by providing a dedicated cache space per VM

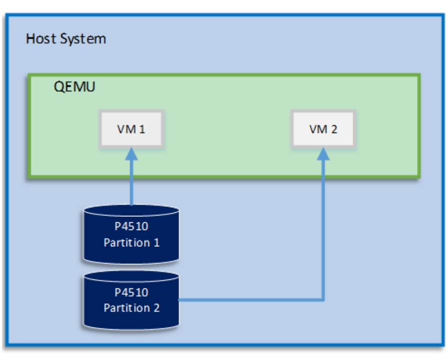

Figure 1: Example of 2 VMs in TLC Configuration

Challenge: Optimize Independent VM Workloads

QEMU is an open-source machine emulator and machine virtualizer. Multiple virtual machines (VMs) often share the same physical storage devices, however, the workload each VM processes on those storage devices is independent. This presents a clear opportunity to optimize the storage workloads on a per-VM basis.

Intel® CAS for QEMU Accelerates Workloads by Up to 4.1x[1]

Intel® CAS for QEMU is a caching solution specifically designed for QEMU. Intel® CAS for QEMU creates an instance of the caching engine on a per-VM basis, separately caching the workload specific to each VM instance. Consequently, Intel® CAS for QEMU isolates a VM by preventing one VM from dominating another VM's caching space. Two test systems with the same server hardware were utilized to test this solution. One system was configured with an Intel® SSD DC P4510 (TLC) while the second system was installed with Intel® CAS for QEMU and configured with an Intel® Optane™ SSD DC P4800X as cache, accelerating an Intel® SSD DC P4320 (QLC).

Figures 1 and 3 show the system configurations. Multiple tests were run, each launching a different number of VMs in parallel to show the performance comparison as the VM count increased. Each VM was assigned an independent drive partition and ran the FIO tool to perform random I/O with a 70:30 read/write ratio on its storage partition. Each VM was therefore running I/O independently of one-another.

Figure 2 below shows how the Intel® Optane™ SSD accelerated system consistently had lower and thus better latency than the TLC system. The acceleration benefit provided by Intel ® CAS for QEMU scaled out as the number of VMs increased reaching up to **4.1x lower read latency at 112 VMs**.[1] Figure 2 shows that the TLC system had a higher read latency which drastically increased as the number of VMs increased. The Intel® Optane™ SSD accelerated system on the other hand, had lower read latency than the TLC system and latency increased at a lower rate.

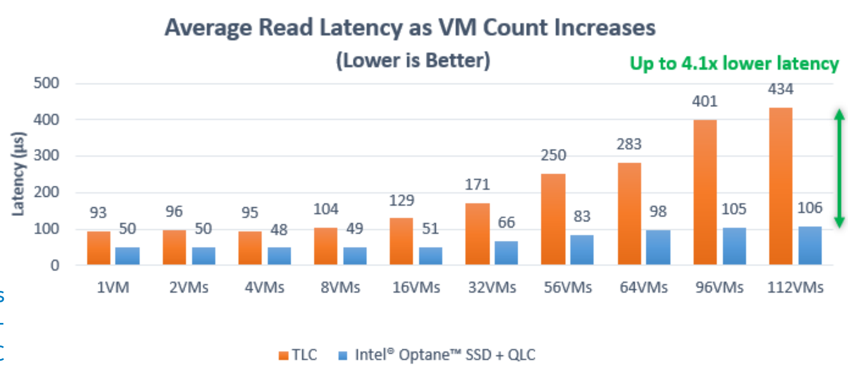

Figure 2: Average Latencies of Intel® Optane™ SSD + QLC vs TLC

Reduced Read Tail Latency Up to 2.8x[1]

Figure 3 shows Intel® CAS for QEMU working inside the emulation code stack for two VMs. As the number of VMs scale out so can Intel® CAS for QEMU, allowing many VMs with different workloads to each have independent caching optimizations. Intel® CAS for QEMU was configured in this solution in write-through mode to achieve immediate data consistency between the cache and back-end devices while accelerating read I/O by caching the hot data.

The amount of locality in the data plus the 70:30 read/write I/O ratio of the test allowed for the improvement of tail latencies at P80 as well. Figure 4 below shows the tail read latency benefit of the Intel® Optane™ SSD accelerated system at P80 compared to the TLC system using the same test scenario described earlier. The acceleration benefit provided by Intel® CAS for QEMU scaled out as the number of VMs increased reaching up to **2.8x lower read tail latency at P80 with 112 VMs**.[1] Figure 4 shows that 80 percent of all I/Os had faster I/O response time on the Intel® Optane™ SSD accelerated system than on the TLC system even as the number of VMs increased.[1]

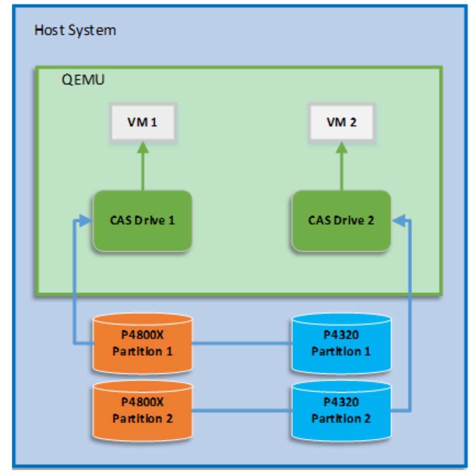

Figure 3: Example of two VMs Running with Intel® CAS QEMU

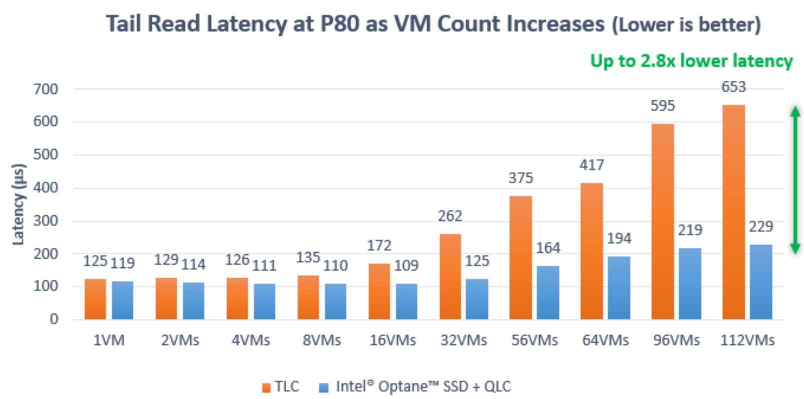

Figure 4: QoS Improvements with Intel® Optane™ SSD + QLC vs TLC

Conclusion

Intel® CAS for QEMU accelerates VM workloads by caching read heavy jobs and helps isolate VMs by providing a dedicating caching space per VM instance. In this solution the read latency was lowered by up to 4.1x using an Intel® Optane™ SSD accelerated system[1.] This solution demonstrates how creating a caching space per VMs accelerates overall I/O despite the use of shared physical resources. A lower latency means better responsiveness to VM end-users resulting in enhancements to latency-sensitive applications.

Learn More

Solutions with Intel® CAS: http://www.intel.com/cas

1. System configuration: Server model: SYS-6029U-TR4T; MB: X11DPU; CPU: Intel® Xeon® Platinum 8180 CPU @ 2.50GHz, 28C/56T, 38.5 MB L3 Cache, Turbo, HT (205W); Mem: 8x32GB Hynix HMA84GR7AFR4N-VK DIMMs (256GB), DDR4-2666; NICs: 4x Embedded Intel X710/X557 10GbE LAN; BIOS Version: 1.10; Host Operating System: Red Hat Enterprise Linux Server release 7.5; Kernel Version: 3.10.0-862.11.6.el7.x86_64; Guest Operating System: CentOS Generic Cloud release 7.5; Kernel Version: 3.10.0-862.3.2.el7.x86_64; QEMU release v2.1.2; FIO release v3.1; TLC config: (1) Intel® SSD DC P4510 8TB with each VM given a 15GiB partition from this drive; QLC Config: (1) Intel® SSD DC P4320 7.68TB and (1) Intel® Optane™ SSD DC P4800X 1.5TB with each VM given a 15GiB partition from P4320 and 12GiB partition from P4800X drives; Intel CAS Software Setup: Version 03.07.00, Intel CAS for QEMU patch version 3.7.0, P4800X partition used for caching, P4320 partition used as a core drive, caching mode was set to write-through; Workload: FIO running inside guest VM, result is the average of 10 trials on preconditioned drives and after cache is warmed, each trial with: 15GiB test size, block size 4KB, time-based 50min test with 10 min ramp runtime, uniform random distribution , random readwrites, 70/30 rw mix, 1 I/O depth, 1 job; Cache warming procedure: 100% random reads from full drive capacity.

0419/FP/HC Please Recycle 338974-001