VMware vSphere Performance with Intel Optane Persistent Memory in Memory Mode – Balanced Profile

Performance Study - August 10, 2020

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Table of Contents

Introduction

Intel® Optane™ persistent memory [1] (PMem) is a memory technology that can deliver larger capacities compared to traditional DDR4 memory. Increasing the memory capacity in a VMware vSphere® [2] virtualized datacenter by operating in Memory Mode allows for an increase in VM density and a subsequent increase in application capacity and performance in the same datacenter footprint. VMware vSphere 6.7 EP10 and later versions [3] supports Intel Optane persistent memory 100 series (Intel Optane PMem) with 2nd Generation Intel® Xeon® Scalable processors in App Direct and Memory Mode.

Past experiments have found some performance challenges with Memory Mode in certain circumstances as seen in the previous whitepaper [4]. A new persistent memory BIOS profile for Memory Mode has been developed that improves performance in some of these cases. This paper explores the performance seen with this new BIOS profile using an Enterprise Java workload and a modified version of the VMware VMmark® 3.1 benchmark [5].

Balanced Profile

Intel Optane PMem coexists with traditional DDR4 DIMMs on the same memory bus. The memory controller in the 2nd Gen Intel Xeon Scalable processor arbitrates between the memory transactions coming from DRAM and Intel Optane PMem. Different arbitration profiles have been defined to determine the algorithm of when the memory controller switches between DRAM and Intel Optane PMem memory transactions. These profiles are configurable options through a persistent memory BIOS setting. Two profiles were originally developed for this: Bandwidth Optimized and Latency Optimized. A new profile called Balanced Profile has been developed to optimize Memory Mode performance.

Figure 1, below, shows how the BIOS profiles affect the memory controller policies for switching between DRAM and Intel Optane PMem requests when there are pending transactions.

- **Bandwidth (BW) Optimized:** arbitrates between DRAM and Intel Optane PMem to maximize DRAM bandwidth
- **Latency Optimized:** arbitrates between DRAM and Intel Optane PMem to minimize DRAM latency
- **Balanced Profile:** optimized for Memory Mode by allowing the controller to switch more often between DRAM and Intel Optane PMem allowing eviction transactions in DRAM to execute faster

Figure 1. This shows the profiles of DRAM and Intel Optane PMem traffic. DRAM is green; PMem is orange.

In Memory Mode, the DRAM is used as a cache while the Intel Optane PMem is used as volatile main memory to deliver DRAM-like performance, depending on the workload. The CPU memory controller will first attempt to retrieve data from the DRAM cache. When the data is present, it will return the request from the DRAM cache, similar to how DRAM access works today. When the data is not present, the request will be sent to Intel Optane PMem. The request is returned to the CPU and in parallel is sent to the DRAM cache. This extra request resulting from the cache miss, in addition to the marginally higher latency of Intel Optane PMem compared to DRAM, can negatively impact performance. Balanced Profile optimizes Memory Mode performance by balancing the thresholds between DRAM and Intel Optane PMem to improve the bandwidth for all DRAM cache misses while maintaining the performance of DRAM cache hits.

For more information, see the Intel technical brief: "Implementing Persistent Memory BIOS Optimizations for Memory Mode" [6]

Testbed Configuration

VMware recommends the use of a 1:4 ratio when using Intel Optane PMem in Memory Mode. This means that the total capacity of DRAM populated in the server should be ¼ or more of the total capacity of persistent memory. Please see VMware KB 67645 [3] for additional information. We configured our testbeds to match this recommendation by keeping the DRAM–to–Intel Optane PMem sizes within this ratio. We first tested how an Enterprise Java workload performed. Later on, we tested a workload generated using VMmark 3.1 [5]. The two testbeds were configured as shown in **Table 1**.

	Enterprise Java Workload Testbed	VMmark Testbed
Processor	1x Intel Xeon Gold 6252 Processor	2x Intel Xeon 8260L Processor
Memory	192GB DDR4 (6x 32GB DIMM) 768GB Intel Optane PMem (6x 128GB)	768GB DDR4 (12x 64GB DIMM) 3TB Intel Optane PMem (12x 256GB)

Table 1. The Enterprise Java workload testbed configuration is on the left, and the VMmark benchmarking software testbed is shown on the right.

Single-VM Performance with Enterprise Java Workload

We performed several experiments with an Enterprise Java workload to determine how this application behaved with a Balanced Profile. The VM memory and Java heap sizes were increased until we were at a point where the sizes were larger than the DRAM size of the system as shown in **Table 2**. As a reminder, the DRAM size is 192GB, while the Intel Optane PMem size is 768GB.

	Experiment 1	Experiment 2	Experiment 3
vCPU	16	16	8
VM Memory Size	60GB	220GB	440GB
Java Heap Size	50GB	180GB	360GB

Table 2. Single VM configurations.

We have normalized all performance scores for this workload with the performance on an identically configured system with only DRAM. As a first step, we verified that Balanced Profile did not harm performance in the expected good case (Experiment 1) which has memory sizes much smaller than the DRAM cache size. We would expect the performance not to be impacted because the VM easily fits within the DRAM cache. As seen in **Figure 2**, there is a 1% impact on performance, which we consider to be within run-to-run variation of this benchmark.

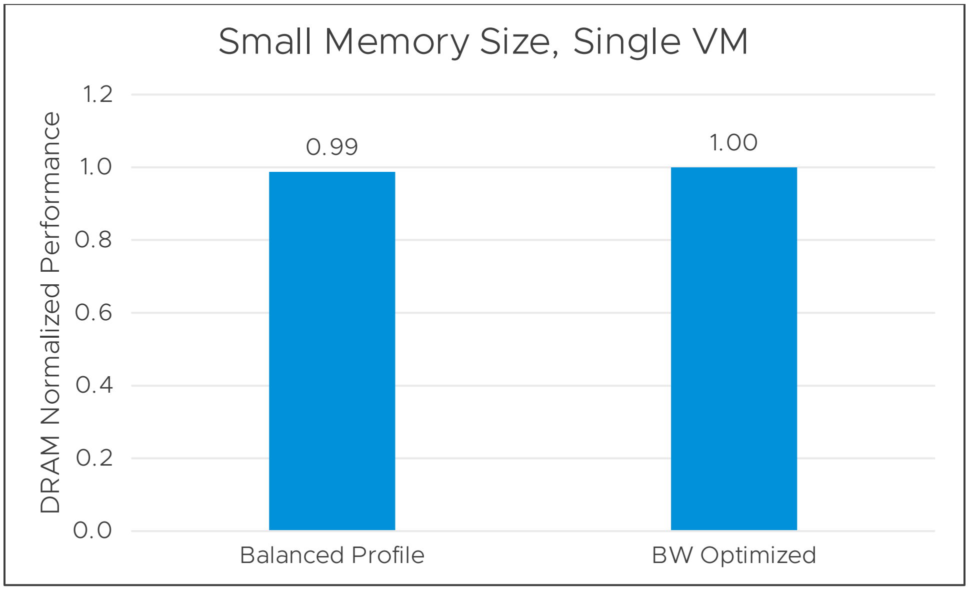

Figure 2. The balanced profile shows performance similar to the bandwidth (BW) optimized bios setting.

Moving to a larger VM memory size, we begin to see some impact on performance. Experiment 2 has the VM memory size at 220GB, slightly larger than the DRAM cache, but the Java Heap is smaller at 180GB. The Java Heap should still be able to fit in the DRAM cache. We do see a minor impact on performance of around 2% as shown in **Figure 3**, below.

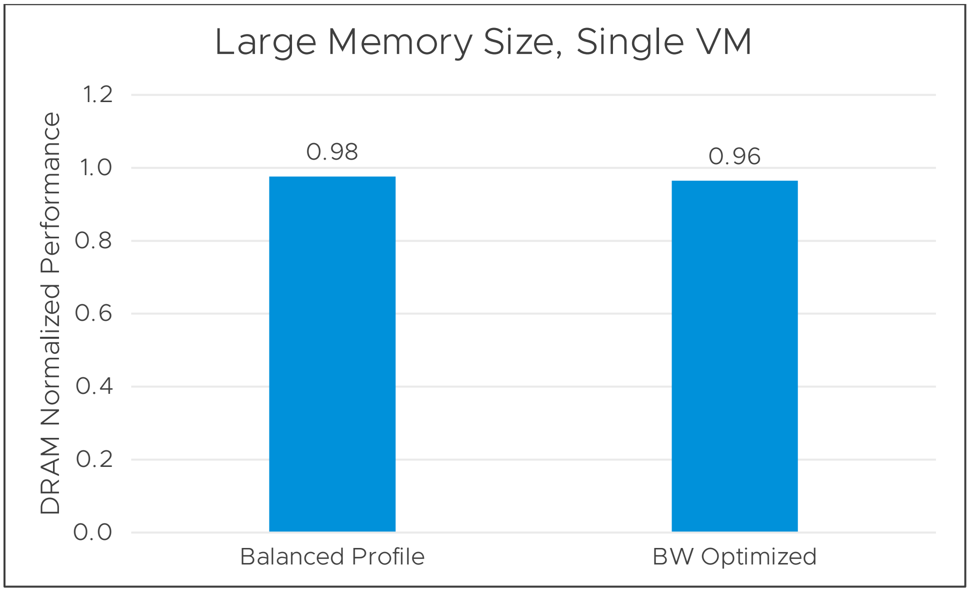

Figure 3. Compared to the balance profile, the bandwidth optimized bios setting has a minor impact of 2% on performance

Finally, we move to a VM that is larger than the DRAM cache size. This experiment uses a VM memory size of 440GB with a Java heap size of 360GB, both much larger than the DRAM cache size of 192GB. We expect the performance of this experiment to be much lower than the DRAM because much of the memory access will need to go directly to Intel Optane PMem, which has higher latency than DRAM. As we see in **Figure 4** below, the performance is lower; however, the Balanced Profile delivers a significant impact on performance with a 1.75x improvement over the Bandwidth Optimized setting.

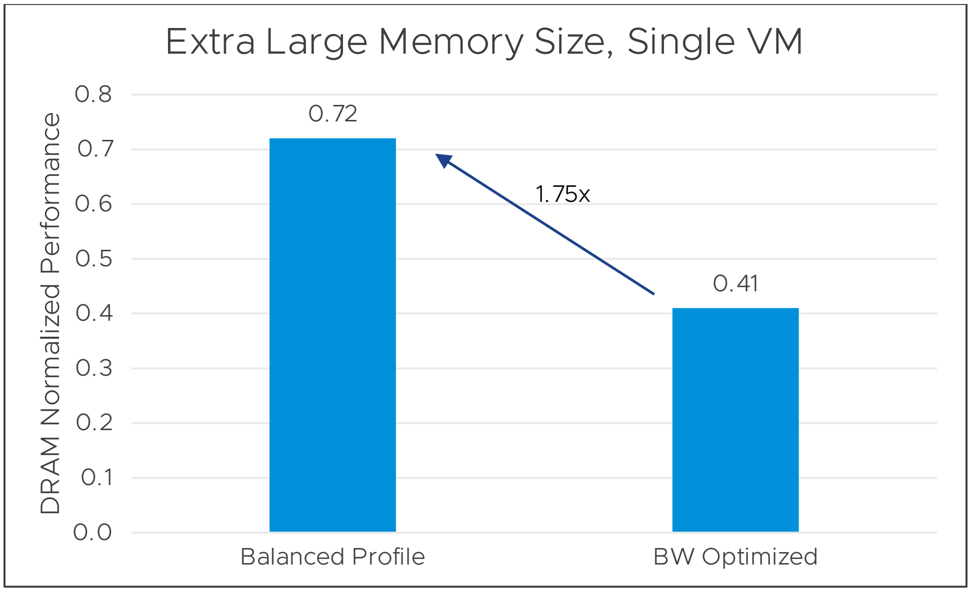

Figure 4. On a VM that is larger than the DRAM cache size, the balanced profile gives 1.75x performance over bandwidth optimized.

These single VM experiments have shown us that the Balanced Profile does not have a negative impact on the performance of a single VM over the Bandwidth Optimized BIOS setting. Additionally, we found a significant improvement in performance when the VM was larger than the DRAM cache size with the Balanced Profile.

Multi-VM Performance with Enterprise Java Workload

We have previously seen some significant performance impact with multi-VM performance in Memory Mode. Balanced Profile attempts to alleviate this issue. This experiment used the VM configurations described below in **Table 3**, with the same Enterprise Java workload as the single VM experiments.

	Small VM	Large VM
vCPU	8	8
VM Memory Size	40GB	80GB
Java Heap Size	30GB	60GB
# of VMs	16	8

Table 3. Enterprise Java Workload with multiple VMs in small and large configurations.

All VMs were first powered on, then two VMs ran the Enterprise Java workload simultaneously. We repeated this process until we ran with 8 pairs of VMs. We used two configurations for these experiments. The first (Small VM) configures VMs so that active memory will consume around 30% of the 192GB DRAM cache when run in a pair. The second (Large VM) configures the VMs so that active memory will consume around 60% of the 192GB DRAM cache when run in a pair.

The VM-pair performance is within 1-3% of each other and can be attributed to normal benchmark run-to-run variations. Ideally, we would expect to have exactly 2x the performance of a single VM, which would be represented as a 2 in the figures below.

The small VM experiment uses a VM configured to consume around 30% of the DRAM cache when run as a pair. We see the results in **Figure 5**, below.

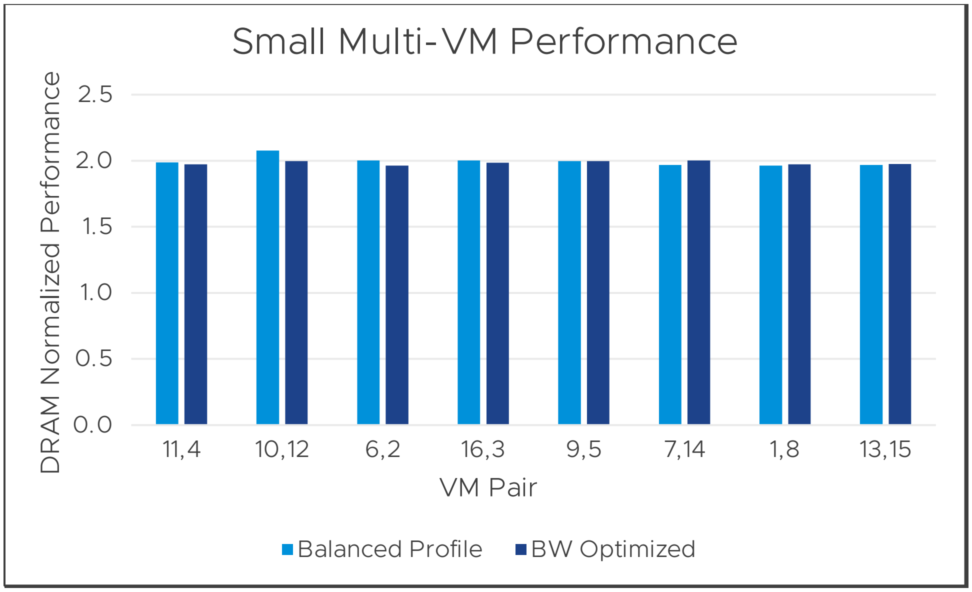

Figure 5. We compared pairs of small VMs and all showed similar performance

Generally, we see very good performance across all pairs. The small amount of active memory will allow most memory accesses to be from the DRAM cache. The small performance differences could be attributed to randomness of the memory placement of each VM.

The large VM experiment configured the VMs to have active memory use of around 60% of the DRAM cache per VM pair. Looking at **Figure 6**, below, we see performance degradation with VM pairs 5 and 7.

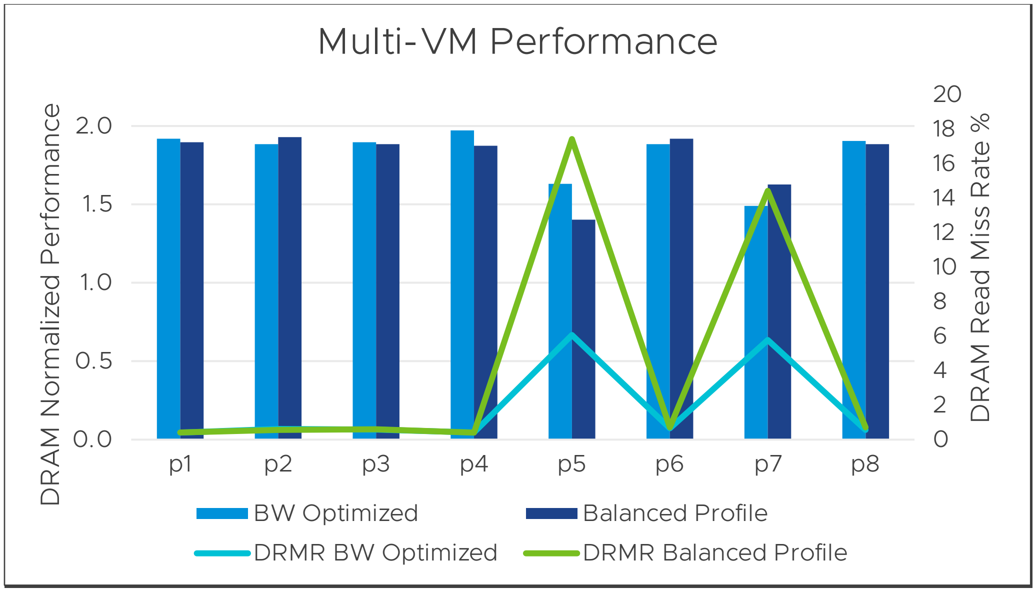

Figure 6. This shows a performance degradation with VM pairs 5 and 7.

Because there is some randomness involved where the memory is mapped with multiple VMs, we needed to look at more than just the performance. We have included the DRAM Read Miss Rate (DRMR) % in the graph to get a better understanding of the performance. This is the percent of read memory accesses that were not present in the near DRAM cache. The DRMR depends on how the memory of the paired VMs were mapped into near the DRAM cache and their access patterns. When the miss rate is very high, performance will decrease. We can see a large increase in the DRMR for pair 5 and pair 7. The Balanced Profile case has a much higher miss rate than the Bandwidth Optimized tests. Despite the larger miss rate, the performance impact is not proportional to the performance impact of the Bandwidth Optimized test. In fact, we see better performance with the Balanced Profile for pair 7, despite a higher miss rate. This shows that the Balanced Profile can sustain better performance when the miss rate increases compared to BW Optimized mode.

Mixed Workload Performance with Modified VMmark 3

Many applications are not sensitive to memory latency. We used VMmark 3.1 without infrastructure operations to examine what happens when these less-sensitive applications are pushed out of the DDR4 cache due to memory hungry "noisy neighbor" VMs. Table 4 shows the following system configuration used for this experiment.

Platform	Intel 2600WFT
Processor	2x Intel Xeon Platinum 8260L Processor
Memory	768GB DDR4 (12x64GB DIMM) / 3TB Intel Optane PMem (12x256GB)
Storage	3x Dell PM1725a 1.6TB NVMe

Table 4. Configuration of system under test

This system can support three tiles of VMmark 3.1 before it fails quality-of-service thresholds. The total memory used by a single tile is 158GB; however, it is not 100% active, so the majority of memory accesses should be fulfilled from the DDR4 cache. We used two VMs running an Enterprise Java workload to fill the DDR4 cache and force the VMmark VMs to access Intel Optane PMem directly. Each Java workload was configured with a 320GB heap; however, we limited the VMs to 8 vCPUs to reduce the impact on CPU resources. Two VMs were used so that each socket would be impacted.

We can see the performance of VMmark in **Figure 7**, below. A 3% variation in performance is normal for this benchmark. We tested four configurations. We ran VMmark 3.1 standalone with Balanced Profile and Bandwidth Optimized BIOS settings, and VMmark 3.1 with two Java workload “noisy neighbor” VMs using Balanced Profile and Bandwidth Optimized.

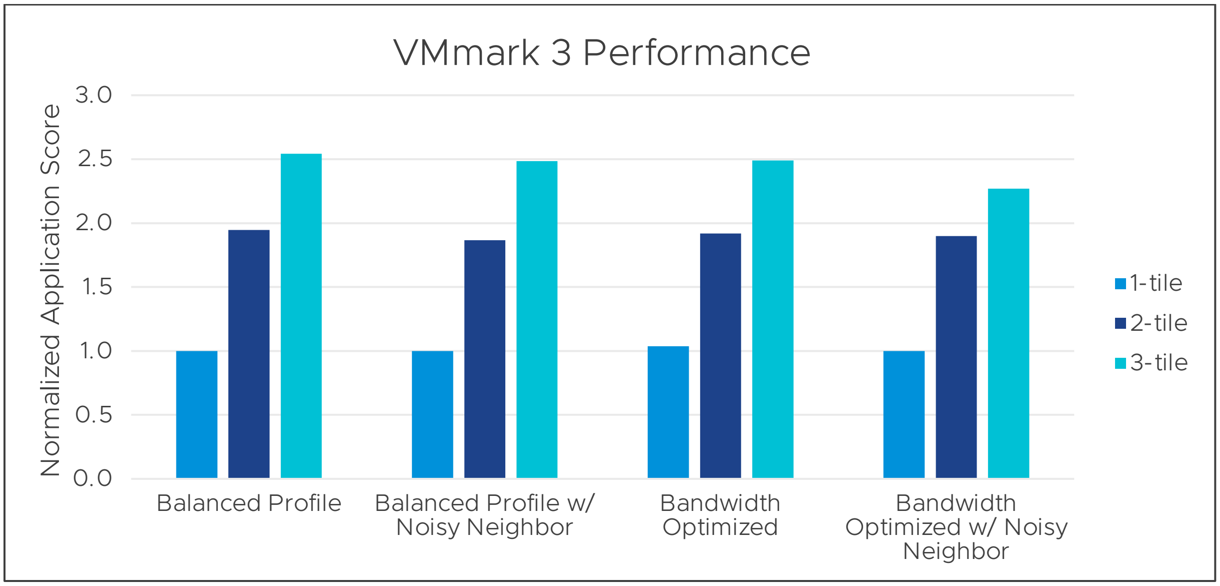

Figure 7. VMmark 3 performance

Performance is similar across all configurations except for the Bandwidth Optimized profile with the noisy neighbor VMs. The 1- and 2-tile tests do not see substantial performance differences. These configurations do not use much memory, so the majority of accesses will be from the DRAM cache. The 3-tile test, which is the most demanding configuration, sees a nearly 11% performance decrease for Bandwidth Optimized with noisy neighbor VMs. There is a much smaller 2% performance decrease for this configuration when we use the Balanced Profile.

Conclusion

We have examined the performance of Intel Optane persistent memory 100 series in Memory Mode with the new Balanced Profile against a variety of single and multi-VM workloads. Using this configuration with the VMware recommended 1:4 memory ratio, we saw no impact in performance for cases where the memory was not stressed. VMmark and Java workloads that can fit in the DRAM cache performed well using this configuration. A previous test of VDI performance, which we examined in a previous paper [4], also does not see an impact. Workloads that are larger than the cache size saw minimal performance impact with the Balanced Profile when compared to identical workloads running with the Bandwidth Optimized configuration. We also saw the performance impact when active memory was over 50% of the DRAM cache in the large-VM-pairs experiment and how Balanced Profile mitigated the impact.

Intel and VMware recommend the Balanced Profile BIOS setting for workloads using Intel Optane PMem in Memory Mode. Both companies worked with server OEM partners to release this optimization as a new persistent memory BIOS setting. Please contact your preferred OEM vendor for more information because it may not be the default setting.

References

[1] Intel Corporation. (2020) Intel Optane Persistent Memory. https://www.intel.com/content/www/us/en/architecture-and-technology/optane-dc-persistent-memory.html

[2] VMware, Inc. (2020) vSphere. https://www.vmware.com/products/vsphere.html

[3] VMware, Inc. (2020, June) vSphere Support for Intel Optane Persistent Memory (PMem) (67645). https://kb.vmware.com/s/article/67645

[4] James Zubb, Milan Potdar, Aravind Bappanadu, Atul Pandey, and Vishal Verma. (2019, July) Intel Optane DC Persistent Memory "Memory Mode" Virtualized Performance Study. https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/IntelOptaneDC-PMEM-memory-mode-perf.pdf

[5] VMware, Inc. (2020) VMmark. https://www.vmware.com/products/vmmark.html

[6] Intel Corporation. (2020, July) Implementing Persistent Memory BIOS Optimzations for Memory Mode. https://cdrdv2.intel.com/v1/dl/getContent/630736

Authors

James Zubb is a staff engineer in the Performance division of VMware. He has 10+ years of experience in performance engineering and benchmark development. He has contributed to the development of the VMmark benchmark since its initial release.

Sajjid Reza is a senior staff engineer at VCE/DPG Intel Corporation. His area of interests includes server virtualization performance analysis and optimization specifically on VMware hypervisor as well as FPGA-based acceleration solutions with VMware vSphere.