



## MEMSTAC PERFORMANCE ON MICRON 9100 PRO NVMe SOLID STATE DRIVES

JON COKER, VP SYSTEM ARCHITECTURE

OMNITIER, INC  
ROCHESTER, MN

## TABLE OF CONTENTS

Executive summary .....	2
Introduction .....	2
Results .....	3
Uniform random workloads .....	3
Uniform read workloads .....	3
Uniform write workloads .....	4
Realistic workloads.....	5
Direct comparison to Memcached for read workloads .....	6
Hardware and test configuration .....	7
Conclusions .....	8
References.....	8
Glossary.....	9
About OmniTier Inc.....	11

## EXECUTIVE SUMMARY

MemStac is a high-performance replacement for MemCached systems, using tiered memory solutions in which the majority of the cache data resides in low-cost, NAND Flash SSDs. MemStac with Micron 9100 PRO NVMe SSDs exhibits a best-in-class tiered memory solution for high-performance key-value caches, achieving MemCached-class performance over commonly used connection counts and get/set ratios. MemStac users will see substantial cost savings using drop-in SSD-based replacements of DRAM-based MemCached systems of equal capacity, with little or no change in system performance. Alternatively, users may elect to deploy substantially larger cache size for the same cost as current DRAM solutions, providing substantial increases in system performance.

## INTRODUCTION

MemCached is a high-performance DRAM caching system used to cache large, networked storage [1]. Modern cloud datacenters extensively use low-latency DRAM caching, such as that offered by MemCached, to improve application performance. As networks and data size grow over time, a fixed cache capacity leads to excessive network traffic and database query overhead. This, in turn, slows overall application performance. With MemCached, system administrators have little option but to add or upgrade servers with additional expensive, power-hungry DRAM in order to maintain system performance requirements.

Micron's 9100 PRO NVMe SSDs are architected from the ground-up to remove legacy layers of storage interfaces, taking full advantage of the speed and parallelism of solid state nonvolatile memory. The 9100 PRO NVMe SSDs streamline efficient queuing via the NVMe protocol. This, combined with an optimized command set register interface, enables low latency and high performance. Data is delivered quickly and efficiently, with minimal burden on the host CPU. Based on industry standards, NVMe's wide adoption is driven by a strong consortium of storage technology providers and a robust ecosystem of drivers across multiple operating systems.

While DRAM-only solutions deliver excellent performance, their maximum available capacity is severely limited by cost, power, and motherboard constraints. These factors have caused an industry-wide revolution to solve the challenging performance requirements with inexpensive, denser memory technologies such as NAND Flash. The technical challenges are indeed high: , these technologies do not exhibit the native speed of DRAM; they require sometimes-cumbersome storage maintenance requirements; and finally, the memory is organized in relatively large block units (typically 4 Kbytes or larger) much like other block storage devices.

The industry is already reporting some success in this transition. Netflix reports successfully integrating Solid State Drives (SSDs) into a high-performance key/value cache using an assortment of open-source applications including MemCached [2]. Redis Labs and Intel have demonstrated greater than 3M operations per second in a read-heavy workload using a SSD/DRAM tiered-memory approach [3].

This whitepaper raises that performance bar yet again solidly into the DRAM class by employing

OmniTier’s MemStac software, NEC servers, and Micron 9100 PRO NVMe SSDs. MemStac is a fully-integrated, multi-threaded, drop-in-replacement implementation of the MemCached protocol.

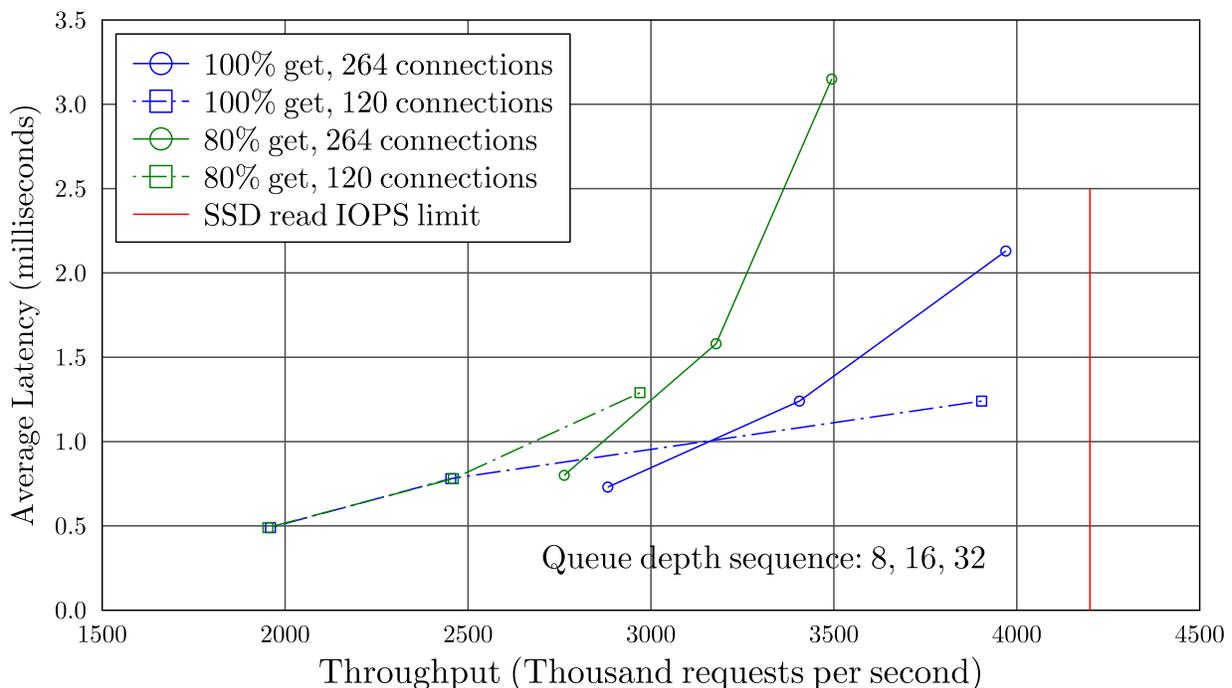
## RESULTS

We showcase MemStac performance under a variety of conditions in order to highlight the MemStac system’s robust performance.

### UNIFORM RANDOM WORKLOADS

In uniform random workloads, each key in the test is invoked with equal probability. This workload is ubiquitous in all storage-system benchmarks, even though this workload is not common in actual customer operations. In *key/value* test systems such as *mutilate* [4] and *memtier* [5], this simple benchmark exhibits the performance of the underlying storage technology ... in the case of MemStac systems, that of 9100 PRO SSDs.

### UNIFORM READ WORKLOADS



**FIGURE 1: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, UNIFORM-RATE WORKLOADS AT VARIOUS TEST LOADS WITH READ-HEAVY 100% GET AND 80% GET / 20% SET RATIOS**

Figure 1 shows mean latency versus throughput characteristics for four different test conditions. Each curve contains a sequence of 3 queue depths per connection: 8, 16, and 32. All conditions depict read-heavy workloads, with 100% get operations in one case and 80% gets / 20% sets in another. For each of these conditions, the total number of connections in the *mutilate* test system was set at either 264 connections or 120 connections. The connection and queue-depth-

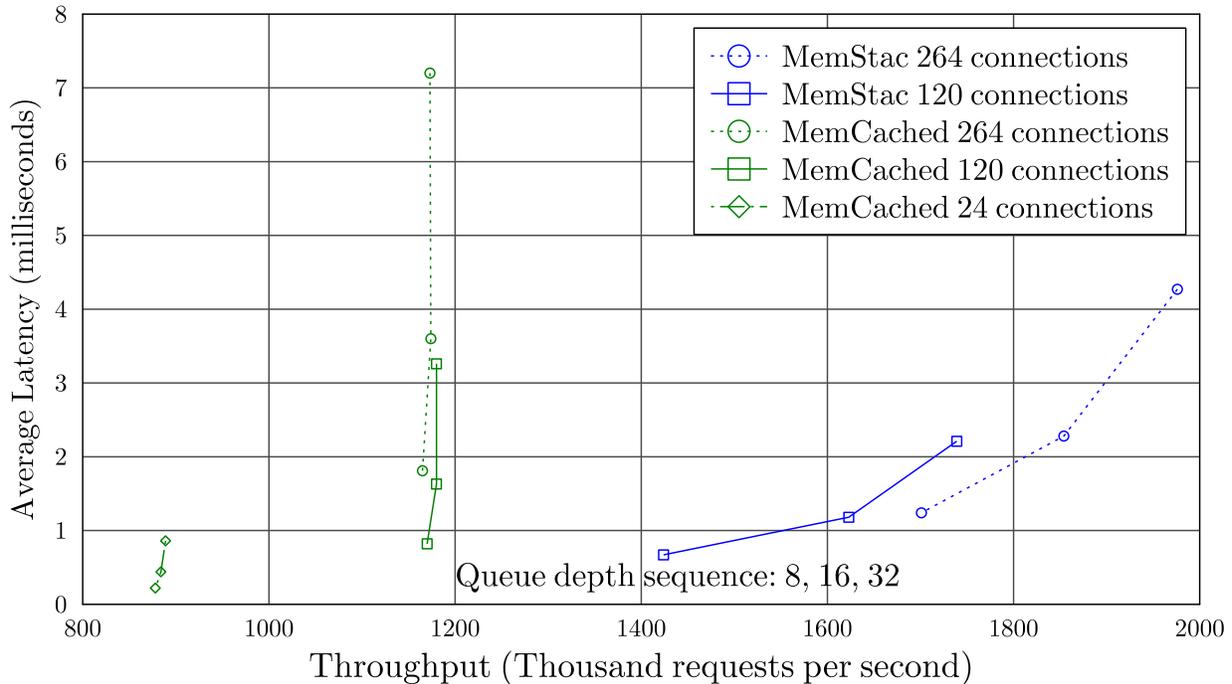
per-connection parameters are measures of the number of commands in flight, and are chosen to ensure that the latency/throughput contours are well characterized in the vicinity of the 1 ms latency target.

The record size is kept constant throughout this test at 100 bytes (20 byte keys and 80 byte values). Very small records are used in this test to stress MemStac’s command overhead. Performance testing with records much larger than 100 bytes will simply be limited by 10Gbe network bandwidth, and therefore is not as technically challenging as performance with small records.

At a system latency of 1 millisecond, the MemStac system exceeds 2.8M operations per second at the 80/20 mixed workload. The performance exceeds 3.2M gets per second at the pure 100% get workload. Therefore, the MemStac system exhibits approximately equivalent performance to the benchmark numbers reported by Redis Labs [5] under substantially similar performance requirements. A key difference exists, however: the Redis Labs demonstration required the use of non-uniform workloads and DRAM tiering technology to achieve the reported numbers. MemStac achieves these numbers *without* benefit of its DRAM tiering technology; that benefit is suppressed in a uniform random workload.

Figure 1 also indicates that the MemStac implementation efficiently approximates the simple bound for 100% uniform get operations given by the maximum random read throughput of six Micron 9100 PRO SSDs with 4K random reads measured at 700K IOPS per SSD.

### UNIFORM WRITE WORKLOADS



**FIGURE 2: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, UNIFORM WRITE WORKLOAD AT VARIOUS TEST LOADS WITH 100% SET OPERATIONS**

Write-heavy workloads are distinct from read-heavy workloads in tiered-memory cache implementations. The frequency-classification algorithms that are so effective in read-heavy workloads are not effective with unpredictable write data. Therefore, simple uniform-rate write benchmarks can effectively characterize write performance for many write workloads.

Figure 2 shows latency/throughput curves for 100% set operations for a standard MemStac installation and a similarly-configured MemCached installation for various loads, run on the same server and test hardware. Because the performance in this pure-write test is dominated by different internal mechanisms than the pure-read test (for both MemStac and MemCached), the record size is increased to a more-typical 250 bytes in these tests [6], producing a higher system stress.

Despite the SSD technology's general reputation of compromised write performance, the MemStac system with NVMe SSDs achieves 1.5M sets per second at one millisecond mean latency, significantly exceeding the maximum throughput of 1.2M sets per second (at any latency) from MemCached. Because of OmniTier's proprietary data handling algorithms, MemStac's write performance can exceed the expected random-write SSD performance bound.

## REALISTIC WORKLOADS

While workloads approximated by the uniform model do exist in practice, caching applications much more commonly exhibit read-heavy workloads in which use frequency varies widely from key to key. MemStac's data classification algorithms quickly and reliably identify the appropriate memory tier for each record, giving substantial gains in overall throughput and in latency over the uniform-rate case.

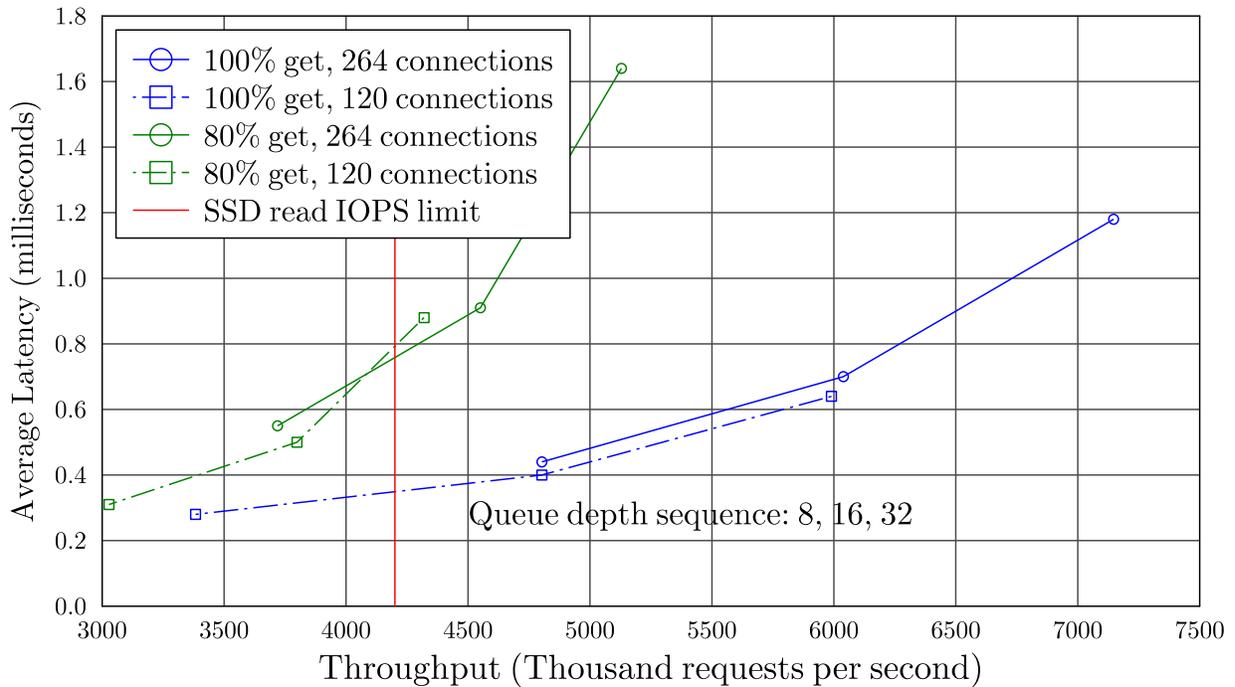
We employ an industry-standard frequency distribution identified by Facebook [6] to identify MemStac's performance in this critical arena. In particular, the ETC frequency distribution with Zipf-Mandelbrot exponential parameter  $s = 2.5$ , and offset parameter  $\beta = 0.03$ . A continuous version of the Zipf-Mandelbrot law for a continuous key index  $0 < \tau < 1$  is given by

$$p_{\tau}(\tau) = \left[ \frac{K}{\tau + \beta} \right]^s$$

for some normalizing constant,  $K$ .

Figure 3 shows substantial gain in performance using MemStac's tiering technology, significantly exceeding the SSD performance bound. The test conditions include 100-byte records and at most 1/8 of the cache capacity resident in DRAM. At a typical latency of 1 millisecond, the pure read workload improves from 3.2M gets per second in a uniform workload to 6.7M gets per second in the Zipfian case. Similarly, the 80/20 mixed workload improves from 2.8M operations per second to 4.6M operations per second.

MemStac is optimized for high performance using small fractions of total SSD cache capacity resident in DRAM. Customers may expect DRAM-class performance using MemStac, over a recommended DRAM installation range of 3% to 15% of the total cache capacity. The best configuration of a particular MemStac installation can depend on system requirements, expected workload distributions, and cost optimizations.



**FIGURE 3: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, FACEBOOK/ETC WORKLOADS AT VARIOUS TEST LOADS WITH READ-HEAVY 100% GET AND 80% GET / 20% SET RATIOS**

### DIRECT COMPARISON TO MEMCACHED FOR READ WORKLOADS

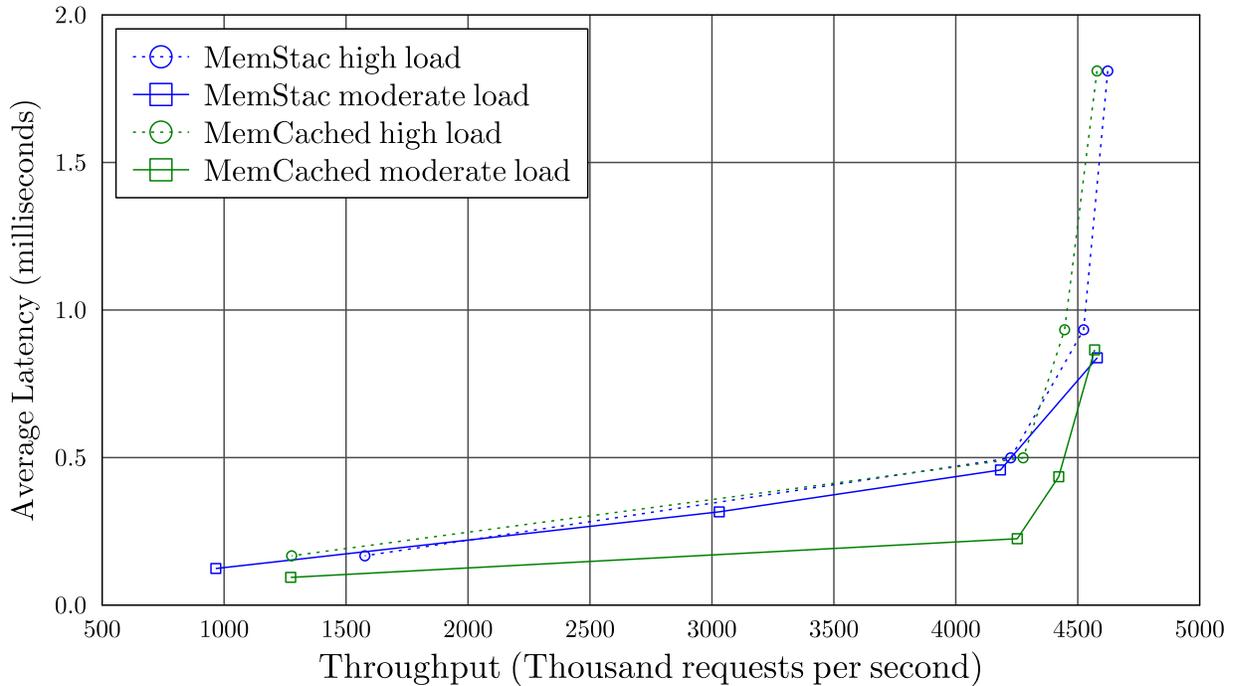
The previous experiments were generally designed to put both the MemStac system and the Micron SSDs “through their paces”, either by choosing a workload which circumvents MemStac’s data classification system, or by using small enough keys such that the experiment is not simply dominated by network limitations. These results show excellent performance in these corner cases, and in fact we have already seen that MemStac significantly exceeds the MemCached standard for write-heavy workloads.

In this section, we compare the performance of MemStac using Micron SSDs to that of MemCached under fully realistic conditions under 100% read workload. The record size is distributed in a “Facebook” workload as described in that same paper [6], with a mean size of 230 bytes. Other test conditions are the same as those described for Figure 3.

The results in Figure 4 show that MemStac system compares favorably to MemCached throughout the very broad load range tested. At the right side of the figure, both MemStac and MemCached are limited to about 4.6M operations per second under this workload because of 10Gbe network throughput limits. These results achieve the state of the art for SSD-based key-value systems, and complete the picture painting MemStac as fully consistent with MemCached-class performance.

This comparison shows MemStac equivalence over a very broad range of typically-used operating conditions. At very-low loading, corresponding to very-low-throughput conditions (far to the left of the range shown in Figure 4), the MemStac system cannot effectively utilize the native parallelization advantages of the underlying SSD technology. In such extremely-

underutilized corner cases, average DRAM latencies can be better than MemStac by on the order of tens of microseconds. For reference, this effect is about the size of the square markers in the figure.



**FIGURE 4: COMPARISON OF MEMSTAC WITH MICRON SSDS AGAINST THE MEMCACHED STANDARD FOR 100% READ WORKLOAD**

A classic 2014 study by the author of *mutilate* test system put maximum read-heavy MemCached performance at up to 3M gets per second at millisecond average latency [4], which is exceeded here. MemCached performance is expected to improve over time with improving hardware and software algorithms, as is that of MemStac. Yet, MemStac’s performance remains competitive in its first implementation, even with relatively small amounts of DRAM.

## HARDWARE AND TEST CONFIGURATION

The hardware configuration for the servers used in this demonstration are as shown in Table 1.

<b>Server</b>	NEC 2-socket Intel Xeon CPU E5-2699 V4, 2.2Ghz, 22 cores per socket
<b>Operating system</b>	Ubuntu Server 14.04, Linux 4.4
<b>Solid State Drives</b>	1.2TB Micron 9100 PCIe NVMe SSDs; sequential read 2.8GB/s ; sequential write 1.3 GB/s; 4K random read 700K IOPS; 4K random write 180K IOPS [7] [8]; six drives per server.
<b>NVMe driver</b>	Standard Linux NVMe driver V1.0 for Linux 4.4

<b>Network Interface Card</b>	10GbE Intel X550T (rev 01)
<b>NIC driver</b>	ixgbe 4.2.1-k
<b>10GbE network switch</b>	Oracle ES2-64 10/40Gbps Ethernet Switch

TABLE 1: NOMINAL HARDWARE CONFIGURATION

All servers are installed with two instances of the Engineering Sample release of MemStac software. All Memcached results are taken with two instances of MemCached software, similarly configured.

Performance measurements are aggregated across the MemStac instances running on the server. The reported numbers always represent steady-state, fully-sustainable MemStac system performance.

## CONCLUSIONS

MemStac, paired with Micron 9100 PRO SSDs, exhibits a best-in-class tiered memory solution for high-performance key-value caches. MemStac achieves MemCached-class performance over a wide variety of test conditions, including a benchmark condition in which its tiering-technology benefit is suppressed. In real-world workloads, MemStac provides excellent performance. Users will see substantial cost savings using drop-in replacements to DRAM-based MemCached installations with no perceptible change in performance. Alternatively, users may elect to buy substantially more cache size for the same price as DRAM solutions, providing substantial increases in application performance.

## REFERENCES

- [1] "Memcached - a distributed memory object caching system," 2016. [Online]. Available: <http://memcached.org>.
- [2] "Application data caching using SSDs," 25 May 2016. [Online]. Available: <http://techblog.netflix.com/2016/05/application-data-caching-using-ssds.html>.
- [3] K. Ouaknine and F. Ober, "Redis on Flash with Intel NVMe SSDs: a high performance benchmark".
- [4] Leverich, "Mutilate: a high-performance memcached load generator," [Online]. Available: <https://github.com/leverich/mutilate>.
- [5] Redis Labs, "A High-Throughput Benchmarking Tool for Redis & MemCached," 2013. [Online]. Available: [https://redislabs.com/blog/memtier\\_benchmark-a-high-throughput-benchmarking-tool-for-redis-memcached#.V\\_1Vh-ArLIU](https://redislabs.com/blog/memtier_benchmark-a-high-throughput-benchmarking-tool-for-redis-memcached#.V_1Vh-ArLIU).
- [6] Atikoglu, "Workload analysis for a large-scale key-value store," in *SIGMETRICS '12*, 2012.

- [7] Micron, Inc. , "9100 Pro 1.2TB datasheet," [Online]. Available: [https://www.micron.com/~media/documents/products/datasheet/ssd/it/9100\\_hhhl\\_u\\_2\\_pcie\\_ssd.pdf](https://www.micron.com/~media/documents/products/datasheet/ssd/it/9100_hhhl_u_2_pcie_ssd.pdf).
- [8] Micron, Inc. , "Micron 9100 PCIe NVMe SSD Product Brief," [Online]. Available: [https://www.micron.com/~media/documents/products/product-flyer/9100\\_ssd\\_product\\_brief.pdf?la=en](https://www.micron.com/~media/documents/products/product-flyer/9100_ssd_product_brief.pdf?la=en).
- [9] J. Leverich and C. Kozyrakis, "Reconciling high server utilization and sub-millisecond quality of service," in *EuroSys '14*, Amsterdam, 2014.

## GLOSSARY

<b>Cache</b>	A relatively-fast and relatively-smaller memory system serving the most frequently-used subset of another, relatively-slower and relatively-larger memory system. Caches are generally allowed to evict previously-stored data in order to make room for new data. Cache data does not persist across power-off events.
<b>Connection</b>	A high-level client/server TCP/IP programming construction by which applications communicate over a network socket, identified by a network address and port number.
<b>Core</b>	A relatively-independent processing unit on a processor chip. Modern processors exhibit multiple cores.
<b>DRAM</b>	Dynamic Random Access Memory. A high-performance, expensive electronic-memory technology always used in processor-based machines.
<b>Get</b>	A MemCached read command, in which the key is provided by the client, and the previously-written value is returned by the server.
<b>ETC</b>	The name of a commonly-tested Zipfian workload which emulates real-world MemCached traffic.
<b>IOPS</b>	IOs Per Second, a throughput measure. An <i>IO</i> is an input/output operation. Pronounced "eye-ops."
<b>Key/Value</b>	An organizing principle of some storage devices, in which the information in a variable-length <i>value</i> is identified, stored, and retrieved by reference to a variable-length <i>key</i> .
<b>KV Store</b>	A machine similar to a key-value cache, except that internal evictions are disallowed, and that data must be persistent across power-off events.
<b>Latency</b>	A measure of the time taken from the start to the end of an operation. Latencies can be reported as distributions or as statistics of the distribution, e.g., the latency mean or maximum.

<b>MemCached</b>	A simple-protocol, high-performance network standard for key-value caches, implemented in open-source software, and employing a single DRAM memory tier.
<b>MemStac</b>	A tiered-memory implementation of a MemCached server produced by OmniTier, Inc., using DRAM and SSD memory tiers.
<b>Mutilate</b>	A MemCached client designed to test throughput and latency statistics of a MemCached server.
<b>NAND Flash</b>	A electronic, non-volatile memory technology configured in strings of dense not-AND logic gates implemented in CMOS silicon.
<b>Queue depth</b>	In mutilate test clients, the <i>queue depth</i> is the number of commands in flight to the MemCached server within a TCP/IP connection.
<b>Record</b>	Refers to a key-value pair thought of as a unit.
<b>Request</b>	A generic MemCached command.
<b>Set</b>	A MemCached write command, in which both key and value are provided by the client.
<b>Socket</b>	A physical location on a server's motherboard which accepts a processor module. A server may exhibit multiple sockets. <i>[Note: TCP/IP protocols use the same word to describe a network connection].</i>
<b>SSD</b>	A Solid State Drive, typically employing NAND Flash technology. A SSD is not actually a drive. Compare to <b>HDD</b> (hard disk drive).
<b>Throughput</b>	A measure of information flow in a storage or transmission system. Typical units of measure include bytes per second, operations per second, gets per second, or requests per second.
<b>Tier</b>	A layer of memory technology in a modern storage server. Storage servers utilizing multiple tiers may be referred to as <i>tiered-memory servers</i> or simply as <i>tiered servers</i> .
<b>Uniform</b>	A description of a particular workload distribution in which test keys are selected randomly but with uniform probability from key to key.
<b>Workload</b>	A description of the sequence of commands presented to a storage device. A <i>pure workload</i> contains commands of only one type. A <i>mixed workload</i> may contain commands of multiple types, such as read and write commands.
<b>Zipfian</b>	A description of a particular workload distribution in which test keys are selected randomly but with variable frequency from key to key, as described by a <i>Zipfian</i> distribution of parameters $s$ and $\beta$ .

## ABOUT OMNITIER INC.

OmniTier Inc., founded in 2015, is a developer of high-performance, function-optimized solutions for modern datacenter infrastructure applications, including data caching, NoSQL, and real-time analytics using novel memory-management architectures. Its leadership team has a track record of delivering many “industry firsts” in data storage and access across different media types. The company currently has offices in Santa Clara, California, and Rochester, Minnesota.