WHITEPAPER

VERSION 0.9

MARCH 26, 2018

MEMSTAC PERFORMANCE ON SAMSUNG PM1725A SOLID STATE DRIVES

JON COKER, VP SYSTEM ARCHITECTURE

OMNITIER, INC
ROCHESTER, MN

TABLE OF CONTENTS

EXECUTIVE SUMMARY

MemStac™ is a high-performance replacement for Memcached systems, using tiered memory solutions in which most of the cache information resides in low-cost, NAND Flash SSDs. MemStac with Samsung 1725a NVMe SSDs exhibits the best-in-class tiered memory solution for high-performance key-value caches, achieving and exceeding Memcached-class performance over commonly-used workloads. MemStac™ users will see substantial cost savings using drop-in SSD-based replacements of DRAM-based Memcached systems of equal capacity with no impact 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 of many kinds [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 application performance. Using standard MemCached servers, system administrators have little option but to add or upgrade servers with additional expensive, power-hungry DRAM memory to maintain system performance targets as systems requirements grow over time.

MemStac™ is a fully-integrated, multi-threaded, drop-in-replacement implementation of the Memcached protocol. Samsung 1725a SSDs were chosen for this evaluation based on their superior performance in random reads and in sequential write throughput. These SSD features lead to MemStac™ system performance at its highest level yet reported.

While DRAM-only solutions deliver excellent performance, their maximum available capacity is severely limited in large-data systems 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 low latency of DRAM; they require sometimes-cumbersome storage maintenance requirements; and finally, the memory is organized in relatively large block units (typically 4 Kbytes) much like other block storage devices.

The industry is already reporting some success in this inevitable transition to SSD-based key value caches. Netflix reports successfully integrating Solid State Drives (SSDs) into a high-performance key value cache, by 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]. Previously-reported MemStac™ results with Toshiba SSDs [4] and Micron SSDs [5] exhibited DRAM-class performance.

This whitepaper raises that performance bar yet again, significantly exceeding DRAM class

performance at typical read cache operating points, by employing OmniTier's MemStac™ software, NEC servers, and Samsung 1725a NVMe SSDs. MemStac™ showcases the industry leading performance of these Samsung devices, exhibiting the best MemStac™ system performance yet measured.

RESULTS

We showcase MemStac™ performance under a variety of conditions, 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. Implemented in key/value test systems such as *mutilate* [6] and *memtier* [7], this simple but somewhat academic benchmark exhibits the performance of the underlying storage technology without reference to real-world behavior.

UNIFORM READ WORKLOADS

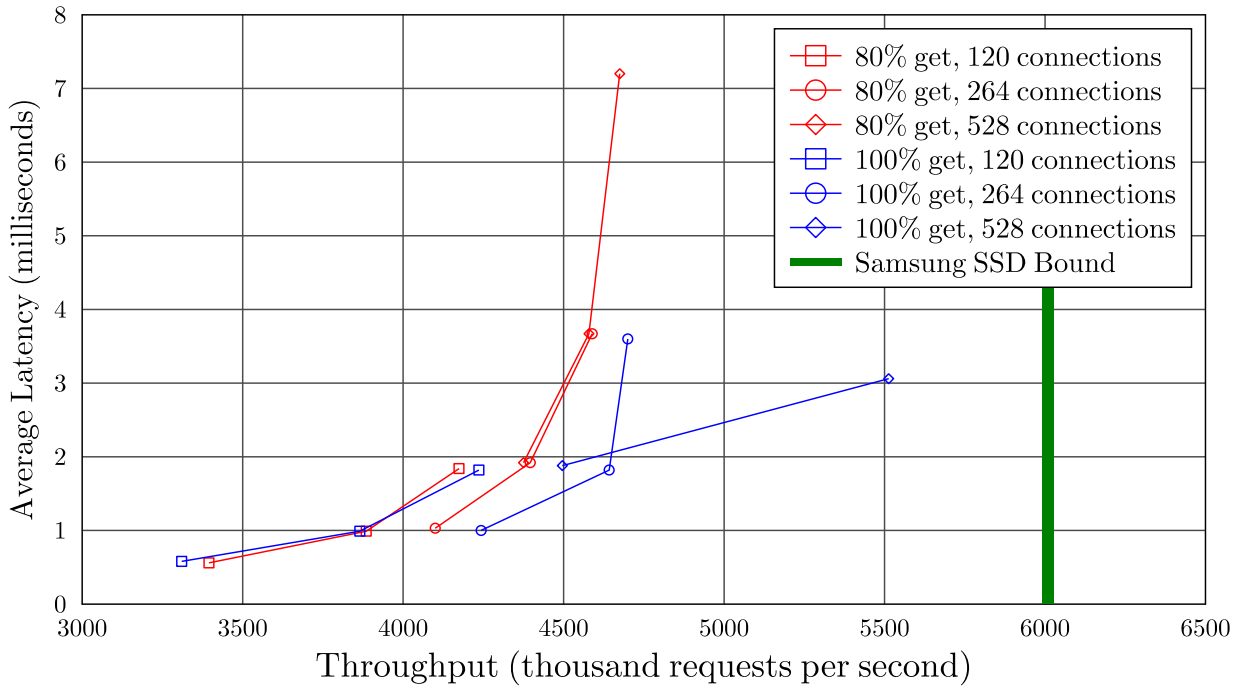

FIGURE 1: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, UNIFORM-RATE WORKLOADS AT VARIOUS TEST LOADS WITH READ-HEAVY 100% GET AND 80% GET / 20% SET RATIOS

Figure 1 shows mean latency versus throughput characteristics for four different test conditions. Each curve represents a sequence of queue depths per connection: 8, 16, and 32; the sequence is 8 and 16 for the 100% get, 528 connection case. All conditions depict read-heavy workloads, with 100% get operations shown in blue and 80% gets / 20% sets shown in red. For each of

these conditions, the total number of connections in the *mutilate* test system was set at 120, 264, and 528 connections. The connection and queue-depth-per-connection parameters are measures of the number of commands in flight. These parameters are chosen to ensure that the latency/throughput contours are well characterized in the regions of interest.

The record size is kept constant throughout this test at 100 bytes (20-byte keys and 80-byte values). Small records are used in this test to stress the command overhead of MemStac™ and the SSD devices. Performance testing with large records will simply be limited by 10Gbe network bandwidth, and therefore is not as technically challenging as performance with small records.

At a system latency target of 1 millisecond (a commonly chosen target), this MemStac™ system exceeds 4.1M operations per second at the 80%-20% mixed workload. The performance is 4.3M gets per second at the pure 100% get workload. Therefore, this MemStac™ system greatly exceeds the benchmark numbers reported by Redis Labs [7] under substantially similar latency 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 can approach the simple throughput bound for 100% uniform get operations given by the maximum random read throughput of six Samsung 1725a SSDs, with 4K random reads measured at approximately 1M IOPS per SSD.

UNIFORM WRITE WORKLOADS

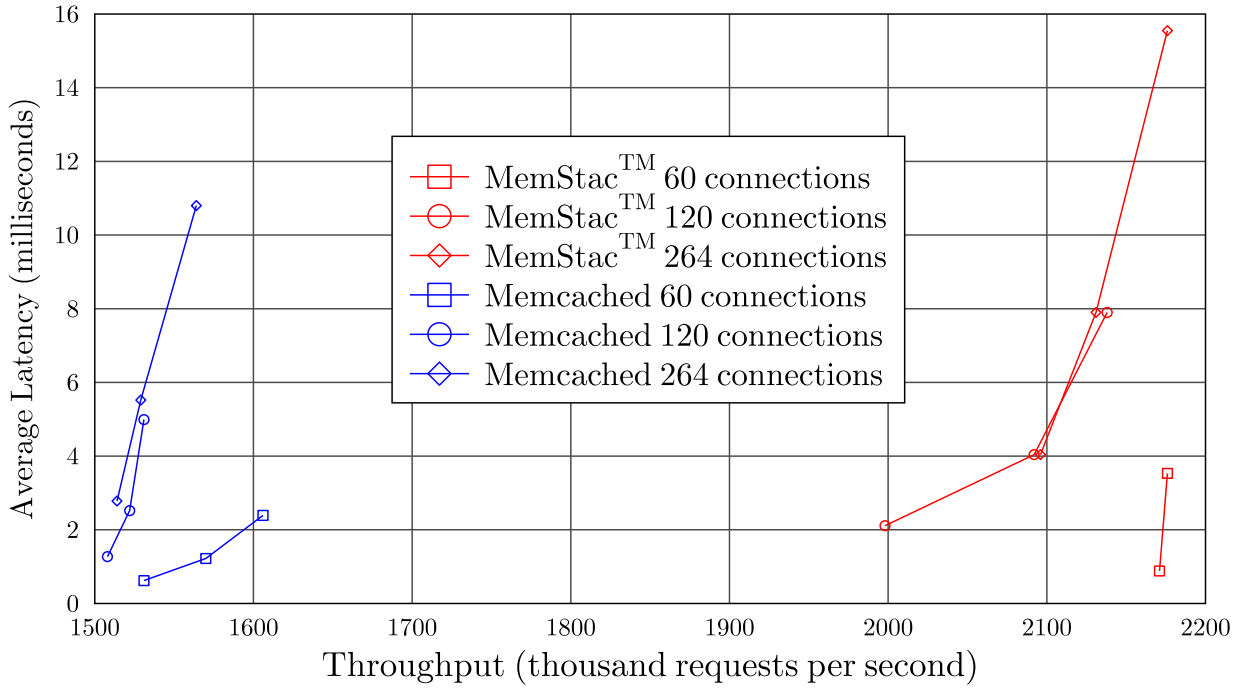

FIGURE 2: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, UNIFORM WRITE WORKLOAD AT VARIOUS TEST LOADS WITH 100% SET OPERATIONS

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. The record size is kept constant throughout this test at 100 bytes (20-byte keys and 80-byte values).

Despite the SSD technology's general reputation of compromised write performance, the MemStac™ system with NVMe SSDs exceeds 2.1M sets per second at one millisecond mean latency, significantly exceeding Memcached's 1.5M sets per second at one millisecond latency. Because of OmniTier's proprietary data handling algorithms, MemStac's write performance can exceed the expected random-write SSD performance bound.

REALISTIC READ 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 [8] to identify MemStac's performance in this critical arena. 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 4.3M 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 4.3M operations per second to 5.3M operations per second. These results are the state of the art for SSD-based key-value systems.

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 10% of the total cache capacity. The best configuration of a MemStac™ installation can depend on system requirements, expected workload distributions, and cost optimizations.

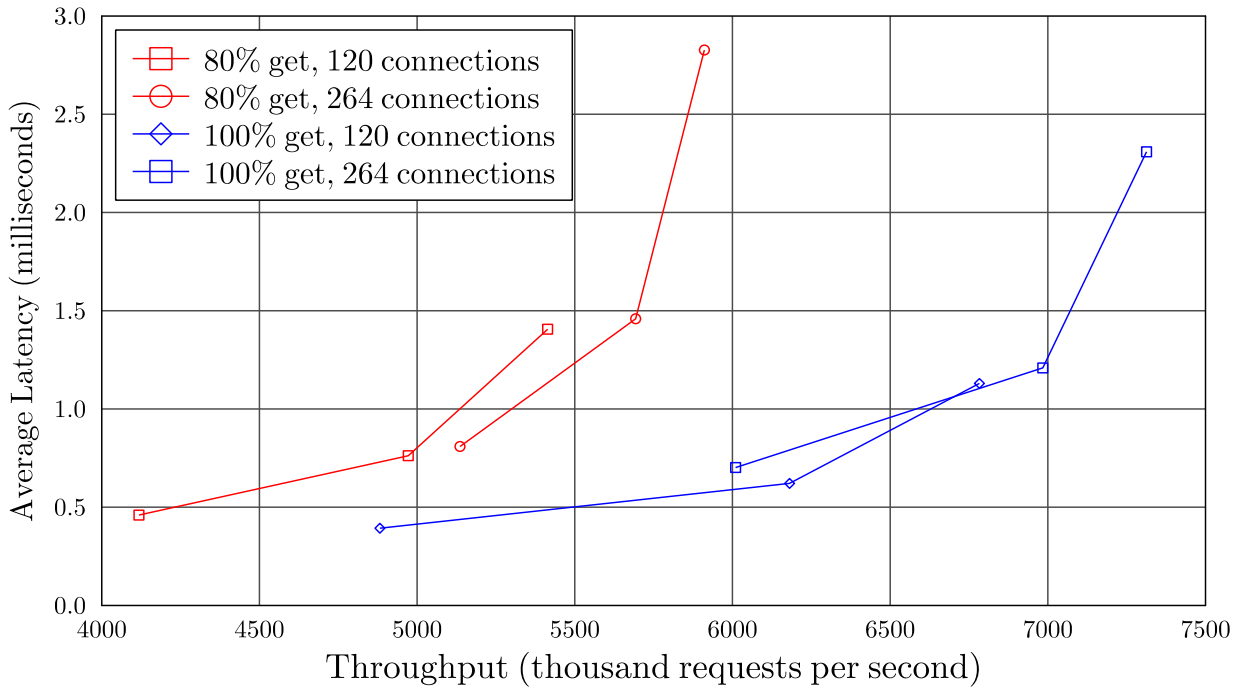

FIGURE 3: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, FACEBOOK/ETC WORKLOADS AT VARIOUS TEST LOADS WITH READ-HEAVY 100% GET AND 80% GET / 20% SET RATIOS

DIRECT COMPARISON TO MEMCACHED FOR READ WORKLOADS

The previous experiments were generally designed to put both the MemStac™ system and the Samsung 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.

We now compare the performance of MemStac™ using Samsung SSDs to that of the latest Memcached version under more realistic conditions and 100% read workload, in Figure 4. The record size is distributed in a "Facebook" workload as described in [8], with a mean size of 230 bytes. The queue depth test sequence for each test curve is 1, 8, 16, and 32. Additionally, we have limited the number of Samsung SSDs in the MemStac™ server to four (a common maximum-hardware configuration). 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 over the broad load range tested. At typical latency targets above about 0.4 uS (at the higher-load end of the spectrum), MemStac™ shows somewhat higher performance than Memcached. MemStac™ and Memcached are similarly limited to about 4.6M operations per second because of 10Gbe network throughput limits. At middle-figure latency targets between 0.2 and 0.4 uS, the performance is about equal, with the winner determined by the load detail. At latency targets below 0.2 uS (corresponding to very low load conditions), Memcached shows somewhat higher performance than MemStac™. These results show that the MemStac™ system is fully

competitive with Memcached-class read performance, and tends to exceed Memcached read performance at the higher loads typically associated with high performance read caches.

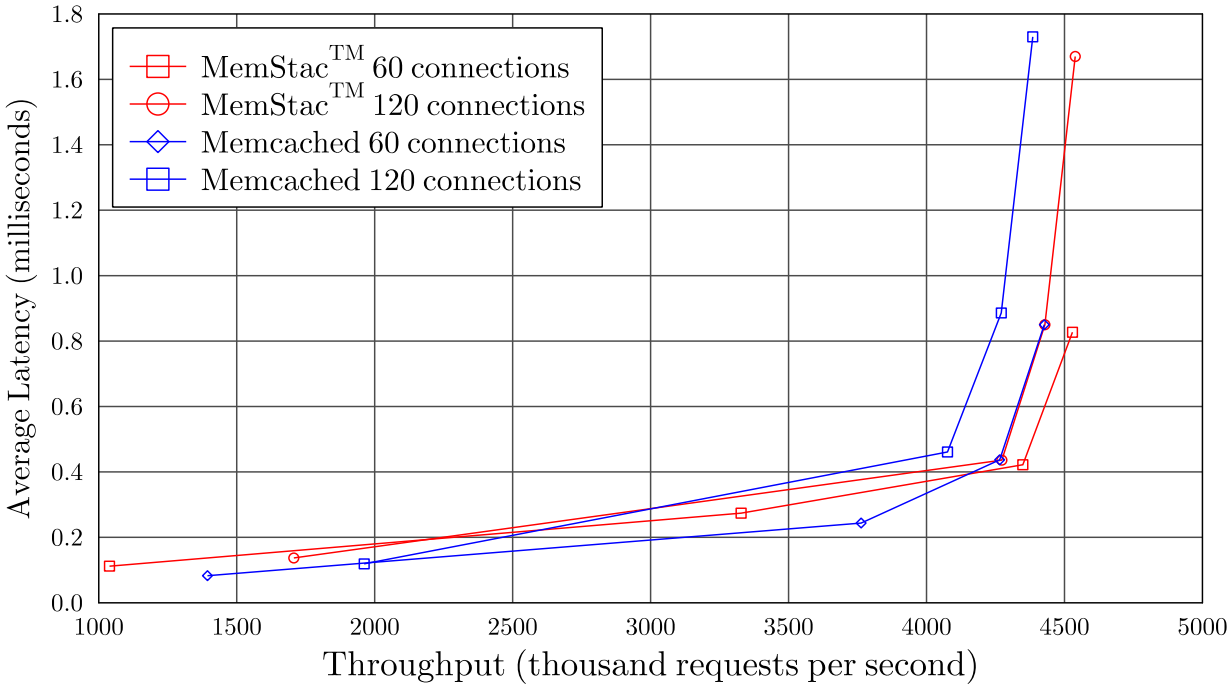

FIGURE 4: COMPARISON OF MEMSTAC™ WITH SAMSUNG SSDS AGAINST THE MEMCACHED STANDARD FOR A 100% GET REAL-WORLD CACHE WORKLOAD

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 one millisecond average latency [6], 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.

Server	NEC 2-socket Intel Xeon CPU E5-2699 V4, 2.2Ghz, 22 cores per socket
Operating system	Ubuntu Server 14.04, Linux 4.4
Solid State Drives	1.6TB Samsung 1725a NVMe SSDs; sequential read 6GB/s ; sequential write 4GB/s; 4K random read 1000K IOPS; four to six drives per server.
NVMe driver	Standard Linux NVMe driver V1.0 for Linux 4.4

Network Interface Card	10GbE Intel X550T (rev 01)
NIC driver	ixgbe 4.2.1-k
10GbE network switch	Not used (all results taken with direct 10Gbe link)
Memcached	Version 1.5.3
MemStac™	General Availability release

TABLE 1: NOMINAL HARDWARE CONFIGURATION

All servers are installed with two instances of the General Availability release of MemStac™ software. All Memcached results are taken with two instances of Memcached version 1.5.3 software, similarly configured, and results aggregated across the Memcached instances.

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

CONCLUSIONS

MemStac™, paired with Samsung PM1725a SSDs, exhibits a new best-in-class tiered memory solution for high-performance key-value caches. MemStac™ achieves Memcached-class performance over a wide variety of test conditions. In real-world workloads, the MemStac™ system provides excellent performance, *exceeding* that of the Memcached DRAM-only performance standard in both write and read workloads, at typical cache operating points. Users will realize substantial cost savings using OmniTier's drop-in replacement to DRAM-based Memcached installations, with DRAM to Flash memory cost ratios now commonly exceeding 10:1. 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] OmniTier, Inc., "MemStac Performance on Toshiba Solid State Drives," October 2016. [Online]. Available: https://omnitier.com/wp-content/uploads/2017/04/MemStac-Performance-on-Toshiba-SSDs.pdf.

[5] OmniTier, Inc., "MemStac Performance on Micron 9100 PRO NVMe Solid State Drives," January 2017. [Online]. Available: https://omnitier.com/wp-content/uploads/2017/04/MemStac-Performance-on-Toshiba-SSDs.pdf.

[6] Leverich, "Mutilate: a high-performance memcached load generator," [Online]. Available: https://github.com/leverich/mutilate.

[7] 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.

[8] Atikoglu, "Workload analysis for a large-scale key-value store," in *SIGMETRICS '12*, 2012.

[9] J. Leverich and C. Kozyrakis, "Reconciling high server utilization and sub-millisecond quality of service," in *EuroSys '14*, Amsterdam, 2014.

GLOSSARY

Cache 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.

Connection 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.

Core A relatively-independent processing unit on a processor chip. Modern processors exhibit multiple cores.

DRAM Dynamic Random Access Memory. A high-performance, expensive electronic-memory technology always used in processor-based machines.

Get A Memcached read command, in which the key is provided by the client, and the previously-written value is returned by the server.

ETC The name of a commonly-tested Zipfian workload which emulates real-world Memcached traffic.

IOPS IOs Per Second, a throughput measure. An *IO* is an input/output operation. Pronounced “eye-ops.”

Key/Value An organizing principle of some storage devices, in which the information in a variable-length *value* is identified, stored, and retrieved by reference to a variable-length *key.*

KV Store A machine similar to a key-value cache, except that internal evictions are disallowed, and that data must be persistent across power-off events.

Latency 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.

Memcached A simple-protocol, high-performance network standard for key-value caches, implemented in open-source software, and employing a single DRAM memory tier.

MemStac A tiered-memory implementation of a Memcached server produced by OmniTier, Inc., using DRAM and SSD memory tiers.

Mutilate A Memcached client designed to test throughput and latency statistics of a Memcached server.

NAND Flash A electronic, non-volatile memory technology configured in strings of dense not-AND logic gates implemented in CMOS silicon.

Queue depth In mutilate test clients, the *queue depth* is the number of commands in flight to the Memcached server within a TCP/IP connection.

Record Refers to a key-value pair thought of as a unit.

Request A generic Memcached command.

Set A Memcached write command, in which both key and value are provided by the client.

Socket A physical location on a server's motherboard which accepts a processor module. A server may exhibit multiple sockets. *[Note: TCP/IP protocols use the same word to describe a network connection].*

SSD A Solid State Drive, typically employing NAND Flash technology. A SSD is not actually a drive. Compare to **HDD** (hard disk drive).

Throughput 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.

Tier A layer of memory technology in a modern storage server. Storage servers utilizing multiple tiers may be referred to as *tiered-memory servers* or simply as *tiered servers.*

Uniform A description of a workload distribution in which test keys are selected randomly but with uniform probability from key to key.

Workload A description of the sequence of commands presented to a storage device. A *pure workload* contains commands of only one type. A *mixed workload* may contain commands of multiple types, such as read and write commands.

Zipfian A description of a workload distribution in which test keys are selected randomly but with variable frequency from key to key, as described by a *Zipfian* distribution of parameters s and β.

ABOUT OMNITIER INC.

OmniTier Inc., founded in 2015, is a developer of high-performance, tiered-memory solutions for modern datacenter infrastructure and scientific computing applications, 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 maintains offices in Milpitas, California, and Rochester, Minnesota.