WHITEPAPER

VERSION 1.0

OCTOBER 1, 2016

MEMSTAC PERFORMANCE ON TOSHIBA 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 the majority of the cache information resides in low-cost, NAND Flash SSDs. MemStac with Toshiba SSDs exhibits a best-in-class tiered memory solution for high-performance key-value caches, achieving MemCached-class performance over a wide variety of test conditions for the first time. MemStac users will see substantial cost savings using drop-in replacements to standard MemCached systems with little or no change in system performance. Alternatively, users may elect to buy substantially more cache size for the same price 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 constant cache capacity leads to excessive networking and database querying overheads, thereby slowing application performance. Currently, system administrators have little option but to add or upgrade servers with additional expensive, power-hungry DRAM memory to maintain system performance requirements.

While DRAM-only solutions deliver excellent performance, the installed capacity is severely limited in large-data systems by cost, power, and motherboard constraints. Therefore, there exists an industry-wide revolution to meet these challenging performance requirements with inexpensive, denser memory technologies such as NAND Flash. The technical challenges are indeed high, because these technologies do not exhibit the native speed of DRAM; they require sometimes-cumbersome storage maintenance requirements, such as wear leveling; and finally, the memory is organized in relatively large block units (typically 4 Kbytes) much like older hard disk and tape technologies. As such, SSDs cannot be classified as random-access memories.

The industry is already reporting some success in this inevitable transition. 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].

This whitepaper raises that performance bar yet again solidly into the DRAM class, employing OmniTier's MemStac software, NEC servers, and Toshiba 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's function is to exhibit the performance of the underlying storage technology … in the case of MemStac systems, that of the SSDs.

UNIFORM READ WORKLOADS

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 record size is kept constant throughout this test at 100 bytes (20 byte keys and 80 byte values). Small records are used 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.

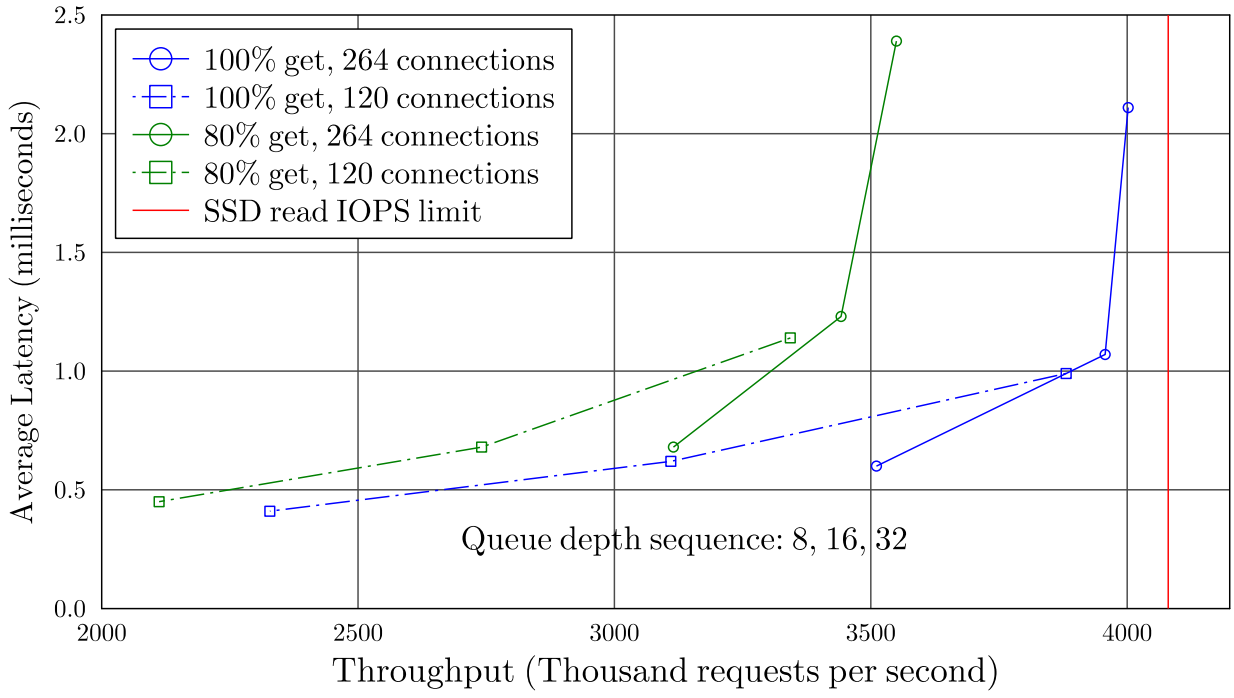

FIGURE 1: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, UNIFORM-RATE WORKLOADS AT VARIOUS TEST LOADS WITH READ-HEAVY 100% GET AND 80% GET / 20% SET RATIOS

At a system latency of 1 millisecond, the MemStac system exceeds 3.2M operations per second at the 80/20 mixed workload. The performance exceeds 3.7M gets per second at the pure 100% get workload. Therefore, the MemStac system meets and exceeds 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 meets the simple bound for 100% uniform get operations given by the maximum random read throughput of six Toshiba SSDs, with 4K random reads measured at 680K IOPS per SSD.

UNIFORM WRITE WORKLOADS

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 250 bytes in these tests.

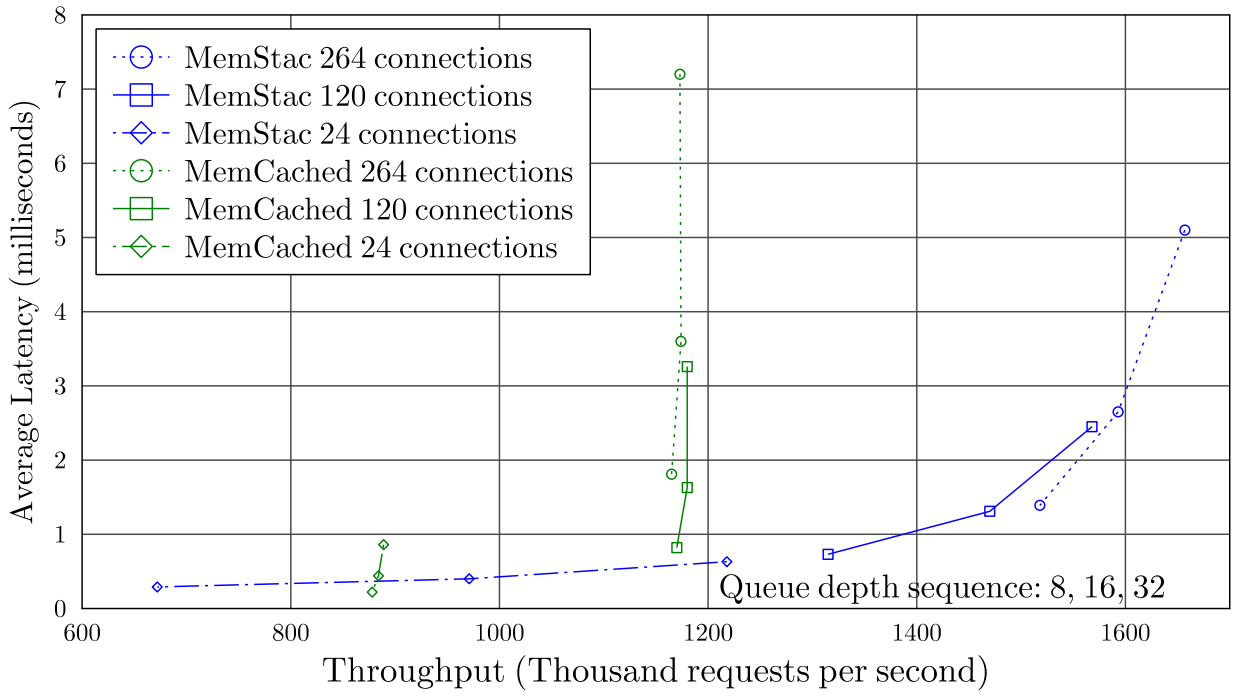

FIGURE 2: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, UNIFORM WRITE WORKLOAD AT VARIOUS TEST LOADS WITH 100% SET OPERATIONS

Despite the SSD technology's general reputation of compromised write performance, the MemStac system achieves 1.4M 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 β = 0.03. A continuous version of the Zipf-Mandelbrot law for a continuous key index 0 < τ < 1 is given by

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

for some normalizing constant *K*.

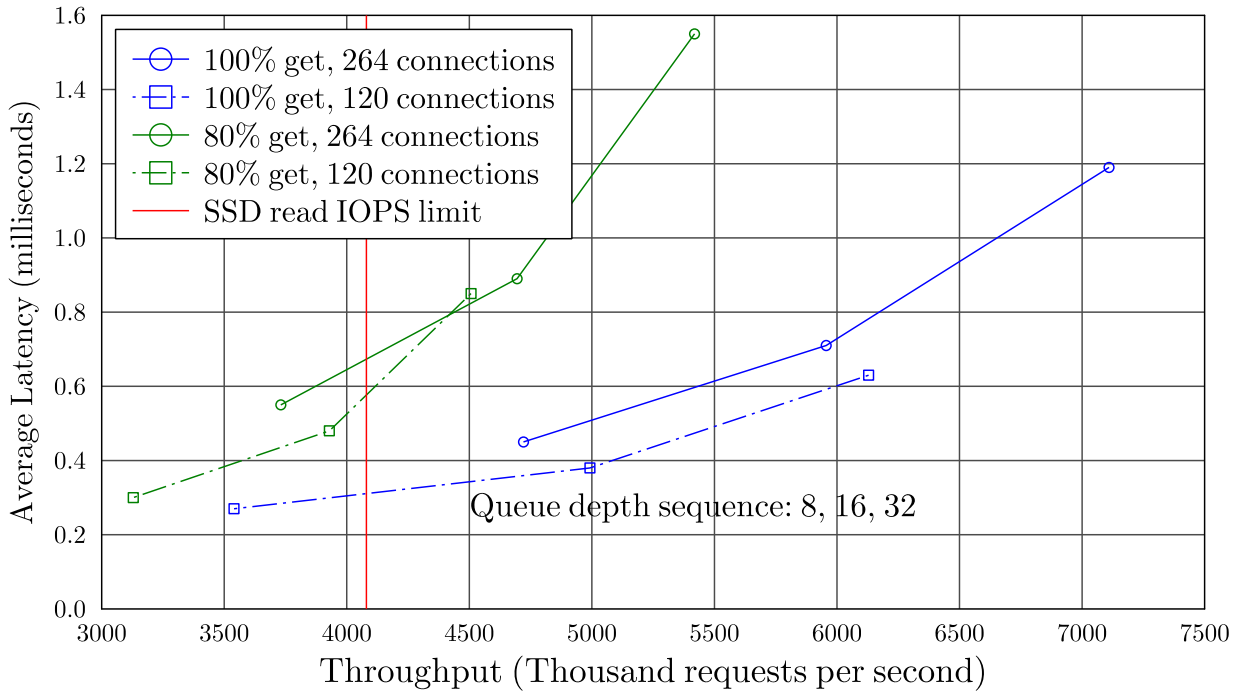

FIGURE 3: LATENCY/THROUGHPUT CURVES FOR SMALL-RECORD, FACEBOOK/ETC WORKLOADS AT VARIOUS TEST LOADS WITH READ-HEAVY 100% GET AND 80% GET / 20% SET RATIOS

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 3.7M gets per second in a uniform workload to 6.6M gets per second in the Zipfian case. Similarly, the 80/20 mixed workload improves from 3.2M operations per second to 4.8M 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 configuration of a particular MemStac installation can depend on system requirements, expected workload distributions, and cost optimizations.

These numbers achieve the state of the art for SSD-based key-value systems, and are well into the range of standard-MemCached class performance. 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].

MemCached performance is expected to improve over time, as is that of MemStac performance. OmniTier's contemporary experiments with highly-optimized versions of MemCached now show read-heavy performance two to three times of that described in the classic paper. Yet, MemStac's performance remains competitive in those workloads in its first implementation, even with small amounts of DRAM. MemStac's superior performance in write-heavy workloads against the same optimized configuration of MemCached was previously presented.

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	800GB Toshiba PX04PMC080 NVMe SSD. Published performance specifications are: 4KiB random reads, 660K IOPS; 4KiB random writes, 185K IOPS; sequential read, 3100 MiB/s; sequential write, 2350 MiB/s. Six SSDs 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	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. Similarly, all Memcached results are taken with two instances of MemCached software.

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 Toshiba 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 little or no change in performance. Alternatively, users may elect to buy substantially more cache size for the same price as DRAM solutions, providing substantial increases in system 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.

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

[7] 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 particular 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 particular 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, 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.