SOLUTION BRIEF

Intel Connectivity
Ethernet Products Group

Performance Testing Application Device Queues (ADQ) with Aerospike

Innovative Intel® Ethernet technology improves Aerospike database performance in benchmark testing.

Performance Terms

Throughput refers to how much data can be transferred from one location to another in a given amount of time, measured, for example, in transactions per second (TPS).

Latency in a network describes the average amount of delay in communication, measured, for example, in microseconds (µs).

Tail latency refers to the slowest response times (with the largest amount of delay) out of all the response times from a system. This is often measured as P(99) or 99th percentile latency, meaning the response time met by 99 percent of all requests.

Jitter is the variation in latency, or the opposite of response-time predictability.

Modern distributed data center applications rely on powerful processors, massive storage, and fast connectivity to achieve high performance and availability. Connectivity performance has traditionally been measured in terms of latency and throughput—that is, the average speed and quantity of data movement. As data centers scale, a third metric becomes increasingly important: predictability.

The Challenge of Predictability at Scale

When implementing a database application, end users typically expect a response in a given amount of time. Network architects design the application and network system to respond within that time. The predictability of an application's response time is typically measured in terms of jitter. Jitter refers to the *variability* of latency—the slight variation (earlier or later) in turnaround time when a response will be received—rather than the average latency in a server. The distribution of jitter increases with scale. Tail latency, the slow outliers in an otherwise fast system, becomes an increasingly significant factor for consideration as the data center scales with more servers.

For an analogy, consider flipping a coin. The odds of getting a heads result on the first flip are 1/2 or 50 percent. As you flip more coins, the probability of getting all heads results goes down exponentially: 1/2, 1/4, 1/8, 1/16, and so on. Likewise, with a server-based system, the probability of receiving every response within an acceptable response time goes down as the number of requests increases. While the likelihood of getting a response within the desired time in a server-based system is much higher (more than 99 percent), the magnitude of trials is also much higher, with tens, hundreds, thousands, or even tens of thousands of servers in use and tens, hundreds, or thousands of transaction requests per server. As a result, network designers must limit the number of servers assigned to a parallelized task, or limit the number of end users, to stay within the desired response time.

"Aerospike's industry leading performance increased more than 75% when leveraging ADQ!"

Srini Srinivasan, Chief Product Officer and Founder, and Paul Jensen, Director of Technology

INCREASES APPLICATION **PREDICTABILITY**

REDUCES APPLICATION **LATENCY**

IMPROVES APPLICATION **THROUGHPUT**

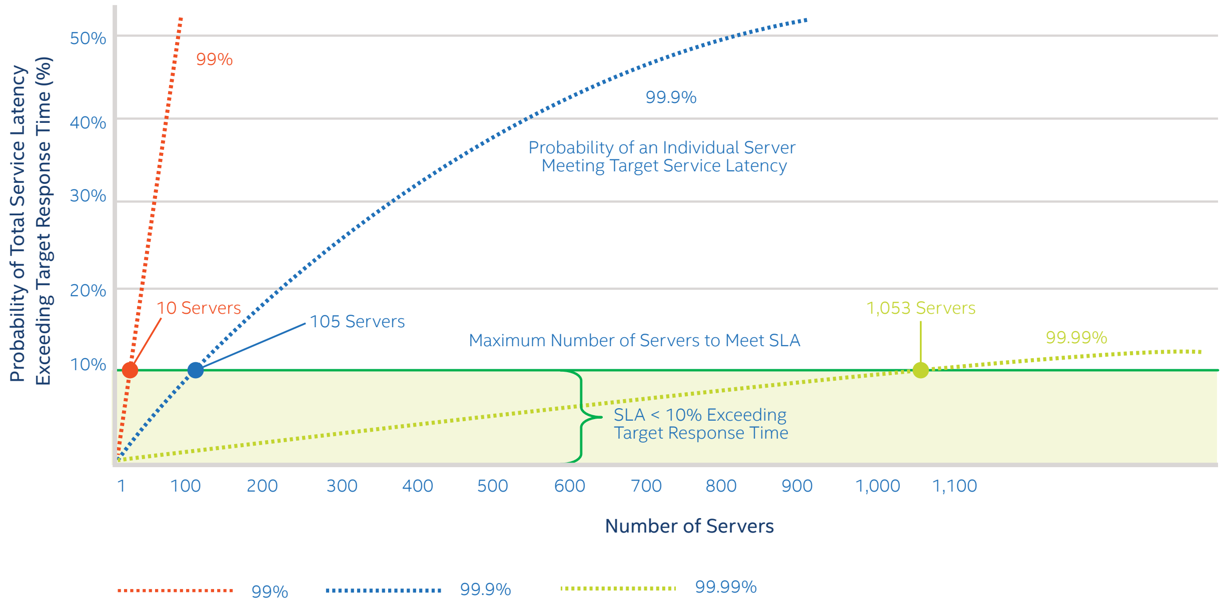


Figure 1. Higher predictability enables more servers working in parallel within the desired response time[1]

Jitter becomes a limiting factor for scalability when adding more servers causes system latency to exceed the user-experience requirements (see Figure 1). This is why, at scale, low latency and high throughput are not enough; predictability is also required.

Introducing Application Device Queues for Predictable, Scalable High Performance

Application Device Queues (ADQ) is an open technology for system-level network input/output (I/O) performance that improves application response predictability, and thereby data center scalability, in a cost-effective manner. ADQ dedicates queues and shapes traffic for the transfer of data over Ethernet for critical applications using standard operating system networking stacks and interfaces that can be supplemented with Intel hardware technologies for improved performance. The goal of ADQ is to ensure that high-priority applications receive predictable high performance through dramatically reduced jitter.

Increasing the predictability of application response times by lowering jitter enables more compute servers to be assigned to a task and can allow more users to access the system, providing a better end-user experience. Even applications that are not large scale can benefit from higher consistency, enabling them to meet service-level agreements (SLAs) more easily.

Data Express Lanes

A good way to understand ADQ is by analogy to freeway traffic. Imagine you want to get from your home to the airport in time to make your flight. If you take the freeway when traffic is light, the trip takes 30 minutes; but if the traffic is heavy, it might take as long as 90 minutes (see Figure 2). This means that, to be safe, you need to plan 90 minutes for the trip, which might waste up to an hour of your time. Likewise, a data-application developer who, not knowing how long it will take to receive requested data, must design around a worst-case scenario.

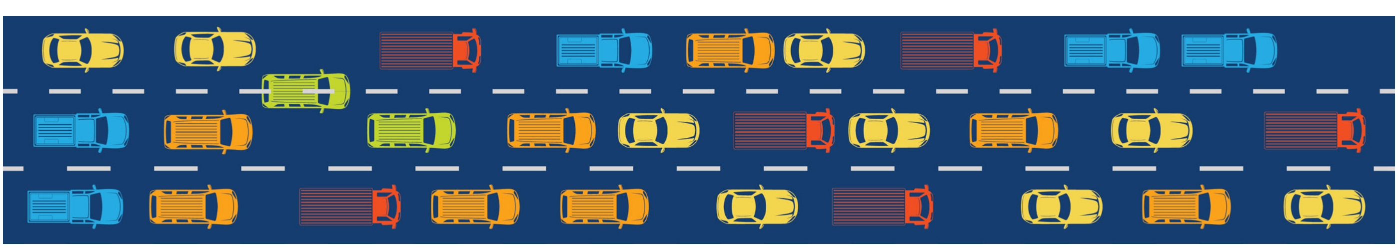


Figure 2. Is there a traffic jam in your data center?

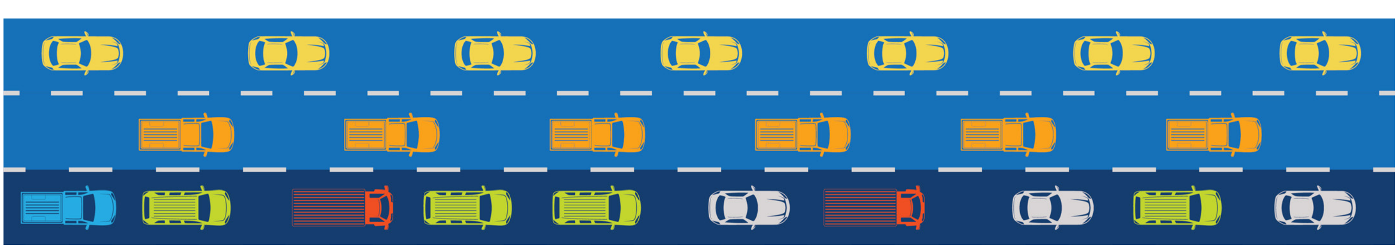


Figure 3. ADQ is like dedicated express lanes for your application data

Now imagine dedicated express lanes available on the freeway allocated only to people traveling from your location to the airport (see Figure 3). On-ramps are metered, off-ramps are limited, lanes are assigned, and speed is fixed so that the trip always takes 30 minutes—unaffected by any other traffic on its way to different destinations.

ADQ provides fast and predictable data-transfer performance that application developers can rely on so they, and network operators, can optimize their applications accordingly. By creating dedicated pipes between application threads and device queues, ADQ not only reduces contention for resources, but it also minimizes or eliminates synchronization operations such as locks and multi-thread sharing. Interrupts and context switching can also add traffic turmoil. ADQ uses busy polling to reduce the number of interrupts and context switches. By using a combination of these methods, ADQ improves application performance and reduces jitter.

ADQ offers application-specific, uncontended, smooth-flowing traffic, because there is no sharing of traffic from other applications on these queues. Traffic shaping reduces jitter by avoiding contention (no traffic jams), rate-limiting traffic (metered ramps), and reducing the number of interrupts and context switches per second (no lane changing).

ADQ Requirements

To take full advantage of ADQ, you'll need an application that has been enabled (if necessary) for ADQ, the latest Intel Ethernet 800 Series technology, and the updated Linux kernel, as shown in Table 1.

Table 1. ADQ deployment requirements

Component	Requirement
Application	Case 1: No change (for example, any single-threaded application or by defining traffic cases alone) Case 2: ADQ-enabled application (for example, Aerospike 4.7 or later for optimized performance)
Configuration	Standard operating system tools and features (for example, iproute2, traffic control [TC], ethtool,and cgroup)
Operating System	Linux 4.19 or later
Ethernet Driver	Intel Ethernet 800 Series driver
Ethernet Network Interface Controller (NIC)	Intel Ethernet 800 Series

Testing Aerospike with ADQ

Intel and its partners have conducted tests to quantify performance improvements achieved by using ADQ with a variety of real-world data center applications. In September 2019, Aerospike conducted testing on the performance of the Aerospike database application.

About Aerospike

Aerospike is a global leader in next-generation, hyperscale data solutions. Aerospike's unique Hybrid Memory Architecture unlocks the full potential of modern hardware and helps eliminate friction that can hold back companies from delivering full value from vast amounts of data at the edge, to the core, and in the cloud. The Aerospike shared-nothing architecture includes three layers: a client layer, a clustering and data-distribution layer, and a data-storage layer.

Aerospike is used by companies including PayPal, Wayfair, Nielsen, Adobe, and many others.[2]

Aerospike is a good representative database back-end application to test (see Figure 4) because its distributed nature—moving large amounts of data to and from the edge, the core, and the cloud—puts a premium on network performance.

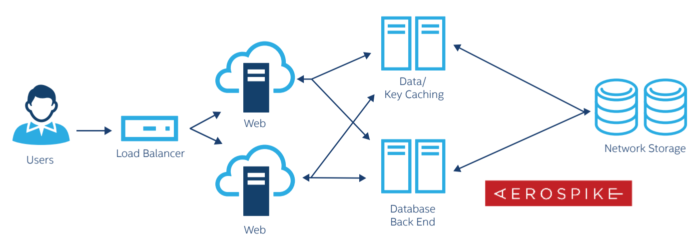


Figure 4. Aerospike is a representative application for testing a database back end

Aerospike version 4.7 is the first commercial database supporting the Intel Ethernet 800 Series with ADQ. Updating the Aerospike server codebase to support ADQ turned out to be relatively easy, because the server codebase was already structured to tie network processing to particular threads as part of its auto-pinning mechanism. A new "auto-pin ADQ" configuration option now ties server-transaction threads to the CPU core servicing the device queue.[3]

Test Setup Details

Aerospike measured ADQ performance in a test environment comprising one dual-socket, dual-NIC server and six clients, all connected to a 100 gigabit Ethernet (GbE) switch (see Figure 5).

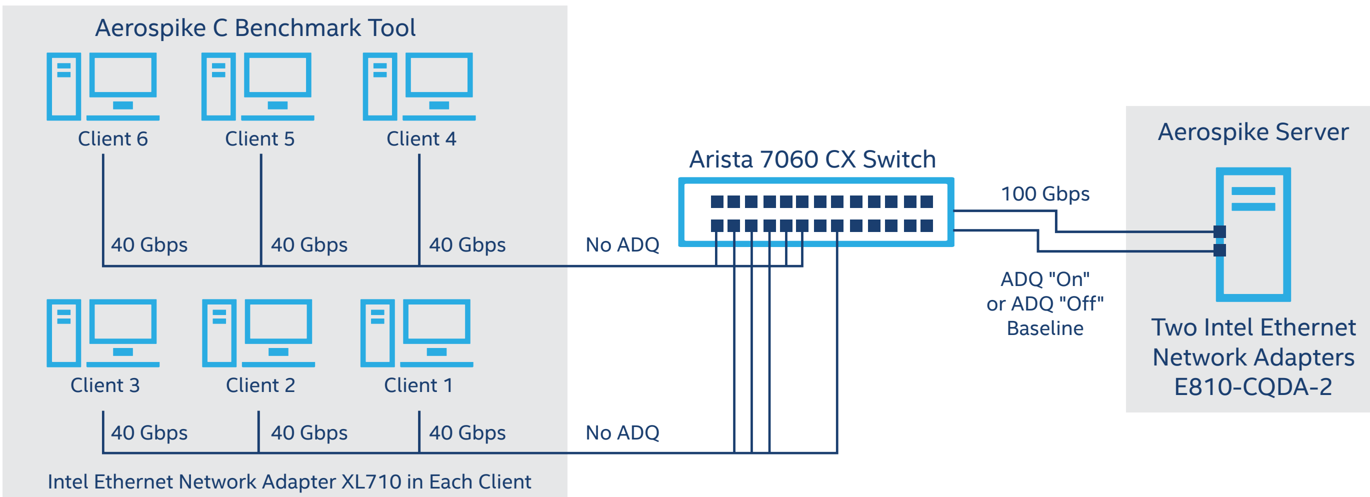


Figure 5. Aerospike test setup

Performance data was gathered by running 18 instances of the Aerospike C benchmark (three per client) against the server. Aerospike includes a non-uniform memory access (NUMA) feature to improve performance. The test configuration used the NUMA feature for testing both with and without ADQ. Relative performance was measured by comparing the new auto-pin ADQ option (ADQ "on") with the previously optimized baseline case of NUMA CPU pinning (ADQ "off" baseline). For detailed test-configuration information, see **the Appendix**.

Test Results

Tests were performed for predictability, latency, and throughput, with results as follows.

Predictablity Results: Increased More Than 45 Percent[4]

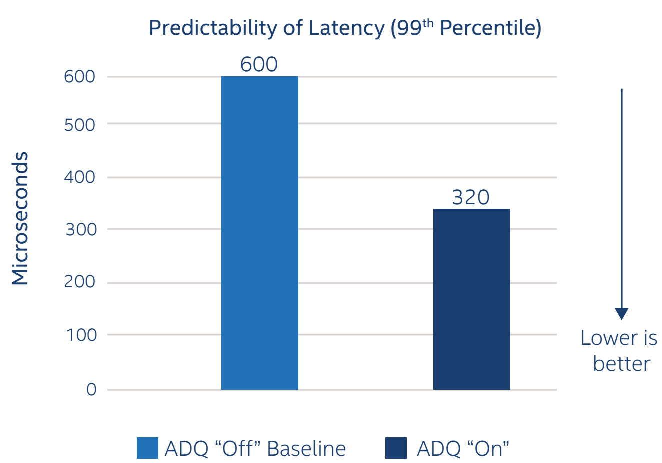


Figure 6. Latency in microseconds (µs) achieved by 99 percent of requests[4]

Aerospike's testing shows an increase of more than 45 percent in response-time predictability for ADQ "on," which uses auto-pin ADQ, compared to the ADQ "off" baseline, which uses NUMA CPU pinning. With ADQ, response times were predictably below 320 microseconds (µs) 99 percent of the time. Without ADQ, response times were only predictably below 600 µs 99 percent of the time (see Figure 6).

Latency Results: Reduced More Than 15 Percent[5]

The Aerospike benchmark provides a histogram of latency, showing the percentage of results that exceeded a series of thresholds from 40–600 µs. Figure 7 shows latency results from a client with ADQ "off" as the baseline compared to with ADQ "on." Note that significantly higher latencies occurred with the ADQ "off" baseline: for example, 11 percent (ADQ "off" baseline) versus zero (ADQ "on") exceeded 360 µs.

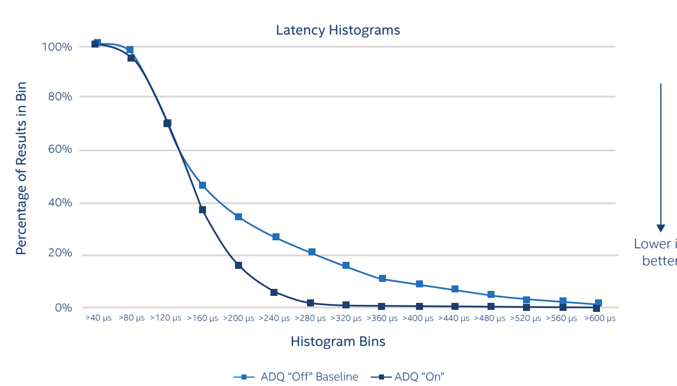


Figure 7. Histogram buckets (ADQ "off" baseline and ADQ "on")[5]

The P(99) abstraction is the single best representation of "tail latency," and it is therefore used for the predictability calculation. One way to comprehend the average latency reduction is to take the weighted average of the number of samples in each bucket and compare the difference. Using this method, the latency decreases from 234.9 µs in the ADQ "off" baseline to 190.8 µs for ADQ "on," a decrease of 18.8 percent. Note that this testing was done with Intel Ethernet 800 Series adapters with ADQ enabled only in the server. Adding Intel Ethernet 800 Series adapters with ADQ into each of the clients is expected to reduce the latency even further.

Throughput Results: Increased More Than 75 Percent[6]

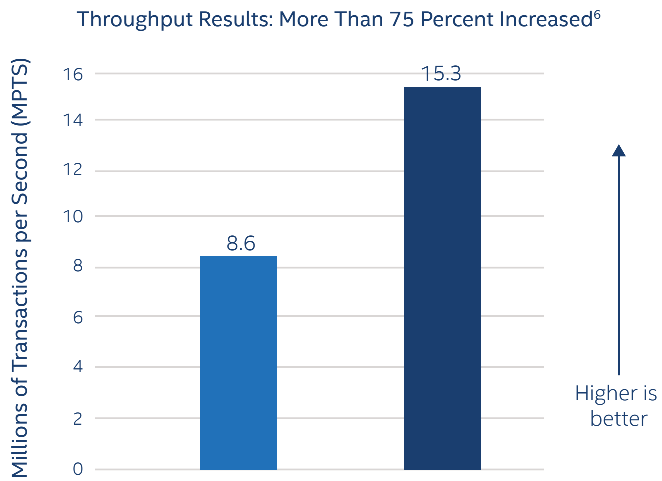


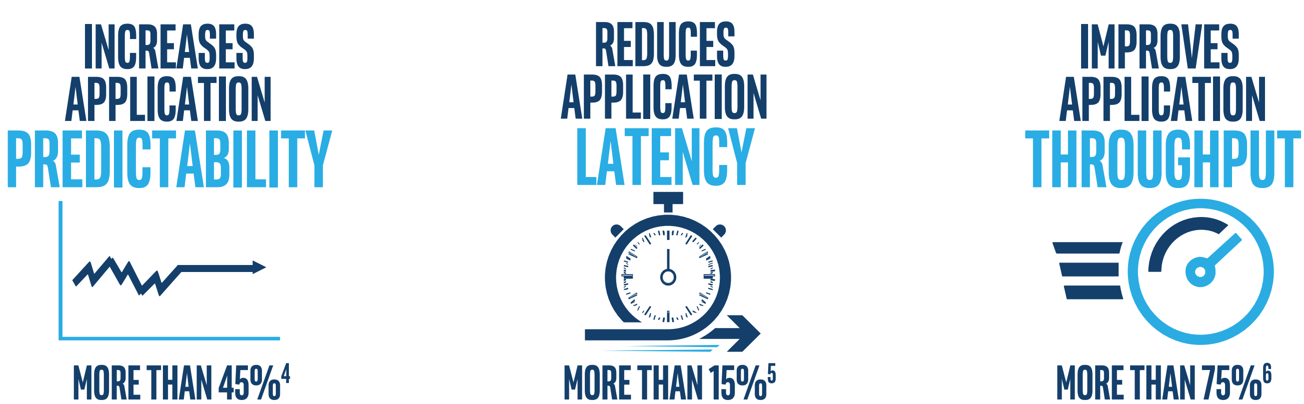
Figure 8. Throughput achieved with and without ADQ[6]

As shown in Figure 8, Aerospike's testing shows a throughput of 15.3 million transactions per second (MTPS) with ADQ "on," compared to only 8.6 MTPS with the ADQ "off" baseline. This represents more than a 75 percent increase in throughput achieved by turning on ADQ.

Final Analysis

Predictability, latency, and throughput performance metrics for Aerospike all showed significant improvement when ADQ was turned "on." Predictability improved more than 45 percent, latency decreased by more than 15 percent (comparing the weighted average latencies), and throughput increased more than 75 percent.

The dramatic improvement in network predictability, based on reduction in tail latency, is of critical importance as data centers and applications scale. Increasing predictability enables more servers to be added to a compute problem, more end users to be served, and greater consistency in meeting SLAs.



Learn More

For more information about the Aerospike 4.7 database, which supports the Intel Ethernet 800 Series with ADQ, visit **aerospike.com/blog/aerospike-4-7-blog-intel-adq-support/**.

For more information about the Intel Ethernet 800 Series with ADQ, visit **intel.com/ethernet**.

Appendix: Test-Configuration Details

Table 3. Test system configuration

	Server System under Test (SUT)	Client
Test By	Aerospike in Intel data center lab	Aerospike in Intel data center lab
Test Date	9/11/2019	9/11/2019
Platform	Intel Server Board S2600WFT (Intel® Xeon® Scalable processor)	Dell PowerEdge R630
# of Nodes	1	6
# of Sockets	2	2
CPU	Intel Xeon Platinum 8280 processor	Intel Xeon processor E5-2699 v4
Clock	2.7 GHz	2.2 GHz
Cores/Socket, Threads/Socket	28 cores/socket, 56 threads/socket	22 cores/socket, 44 threads/socket
ucode	0x4000013	0xb00002a
Intel Hyper-Threading Technology (Intel HT Technology)	Enabled	Enabled
Intel Turbo Boost Technology	Enabled	Enabled
BIOS Version	SE5C620.86B.0D.01.0286.011120190816	SE5C610.86B.01.01.0027.071020182329
System DDR Memory Configuration: Slots/Cap/Run Speed	12 slots, 64 GB DDR4, 2,666 megatransfers per second (MT/s)	16 slots, 8 GB DDR4, 1,866 MT/s
System Intel® Optane™ Persistent Memory (PMem) Configuration: Slots/Cap/Run Speed	Not applicable (N/A)	N/A
Total Memory/Node (DDR + Intel Optane PMem)	3,840 GB	128 GB
Storage (Boot)	Intel SSDSC2KB96, 960 GB	WDC WD1003FZEX-0, 1 TB
Storage (Application Drives)	N/A	N/A
Network Interface Controller (NIC)	Intel Ethernet Network Adapter E810-CQDA2	Intel Ethernet Network Adapter XL710
# of NICs/Node	2	1
Platform Chipset	Intel C620 Series Chipset	Intel C610 Series Chipset or Intel X99 Series Chipset
Other Hardware (Accelerator)	N/A	N/A
Operating System	CentOS Linux release 7.6.1810 (Core)	CentOS Linux release 7.5.1804 (Core)
Kernel	4.19.45	4.18.9
Indirect Branch-Restricted Speculation (IBRS) (0=disabled, 1=enabled)	1	1
Enhanced IBRS (eIBRS) (0=disabled, 1=enabled)	1	0
Retpoline (0=disabled, 1=enabled)	Unknown	Unknown
Indirect Branch Predictor Barrier (IBPB) (0=disabled, 1=enabled)	1	1
Page Table Isolation (PTI) (0=disabled, 1=enabled)	0	1
Mitigation Variants (1,2,3,3a,4,L1TF)	1,2,3,3a,4,L1TF	1,2,3,3a,4,L1TF
Workload and Version	Aerospike Enterprise Edition 4.7	Aerospike C Client 4.4.1 benchmark in async mode
Workload Instances	2/SUT	3/client
Total Workload Instances	2	18
Compiler	gcc (GCC) 4.8.5 20150623	gcc (GCC) 4.8.5 20150623
NIC Driver	Linux (ice): 0.11.7 NVM: 0x800018f7	Linux (i40e): 2.3.2-k EEPROM: 5.05 0x80002882 0.0.0

Sample Aerospike C Benchmark Command	–	nohup ./asd-bench -h 192.168.40.25 -k 1000000 -o S:32 -w RU,100 -a -c 142 -W 12 --latency 16,4 > lst1.out 2> lst1.err &
Network Switch	Arista 7060-CX 32-port 100 Gbps	Arista 7060-CX 32-port 100 Gbps switch, with client connections at 40 Gbps

Table 4. SUT network-adapter configuration settings

	ADQ "Off" Baseline with NUMA CPU Pinning	**ADQ "On"** with Auto-pin ADQ (NUMA ADQ Pinning)
System Settings		
Interrupt Moderation	Fixed	Fixed
IRQ Balance	No	No
Interrupt Affinitization	Yes	Yes
ADQ Settings		
Epoll Busy Poll	No	Yes
Socket Option for NAPI ID	No	Yes
TC-Mqprio Hardware Offload and Shaper	No	Yes
TC- Cloud Filter Enabling with TC-flower	No	Yes
Symmetric Queueing	No	Yes

[1] Jeffry Dean and Luiz Andrée Barroso. "The Tail at Scale." Communications of the ACM. February 2013. **https://cseweb.ucsd.edu/~gmporter/classes/fa17/cse124/post/schedule/p74-dean.pdf**.

[2] Aerospike. "Why our customers love and trust us." September 2019. **aerospike.com/customers/**.

[3] Aerospike. "Announcing Aerospike 4.7 – the First Commercial Database to Support the Intel® Ethernet 800 Series with ADQ." September 2019. **aerospike.com/blog/aerospike-4-7-blog-intel-adq-support/**.

[4] More than 45 percent predictability improvement with Aerospike using 2nd Generation Intel Xeon Scalable processors and the Intel Ethernet 800 Series with ADQ compared to without ADQ. Source: Aerospike testing as of September 11, 2019; Aerospike 4.7 on 2nd Generation Intel Xeon Scalable processors and the Intel Ethernet 800 Series, 100 GbE (see Appendix). Calculation: (new – old) / old x 100 percent for the lowest latencies achieved by 99 percent of requests for the baseline compared to with ADQ.

[5] More than 15 percent lower weighted average latency with Aerospike using 2nd Generation Intel Xeon Scalable processors and the Intel Ethernet 800 Series with ADQ compared to without ADQ. Source: Aerospike testing as of September 11, 2019; Aerospike 4.7 on 2nd Generation Intel Xeon Scalable processors and the Intel Ethernet 800 Series, 100 GbE (see Appendix). Calculation: (new – old) / old x 100 percent for the weighted average of latency samples for each bin = (190.800 µsec ADQ "on" - 234.949 µsec ADQ "off" baseline) / 234.949 µsec ADQ "off" baseline = 18.79% latency reduction.

[6] More than 75 percent throughput improvement with Aerospike using 2nd Generation Intel Xeon Scalable processors and the Intel Ethernet 800 Series with ADQ compared to without ADQ. Source: Aerospike testing as of September 11, 2019; Aerospike 4.7 on 2nd Generation Intel Xeon Scalable processors and the Intel Ethernet 800 Series, 100 GbE (see Appendix). Calculation: (new – old) / old x 100 percent for average transaction request rate across all runs for the baseline compared to with ADQ.

Printed in USA 0420/TK/PRW/PDF Please Recycle 341548-001US