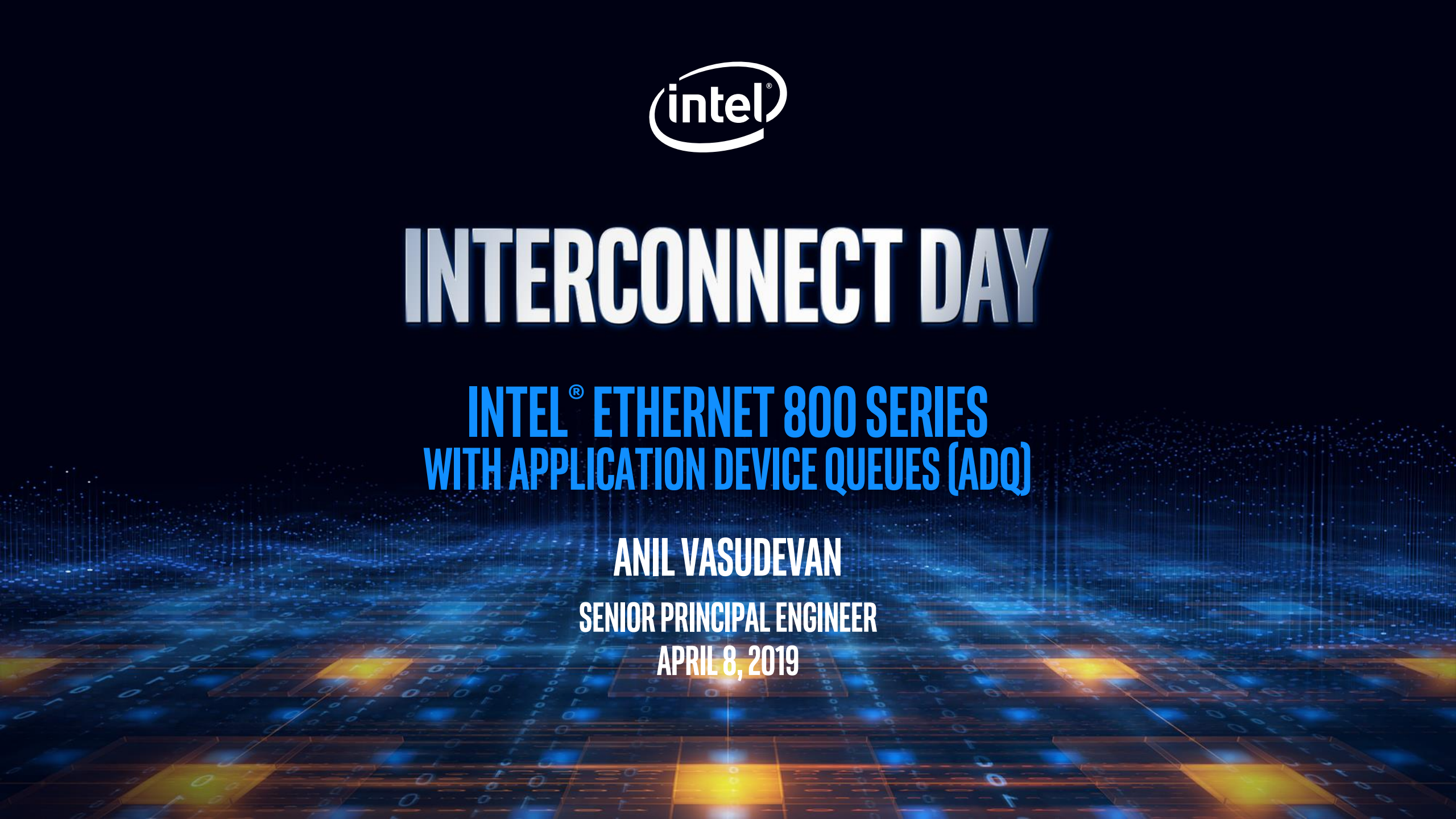


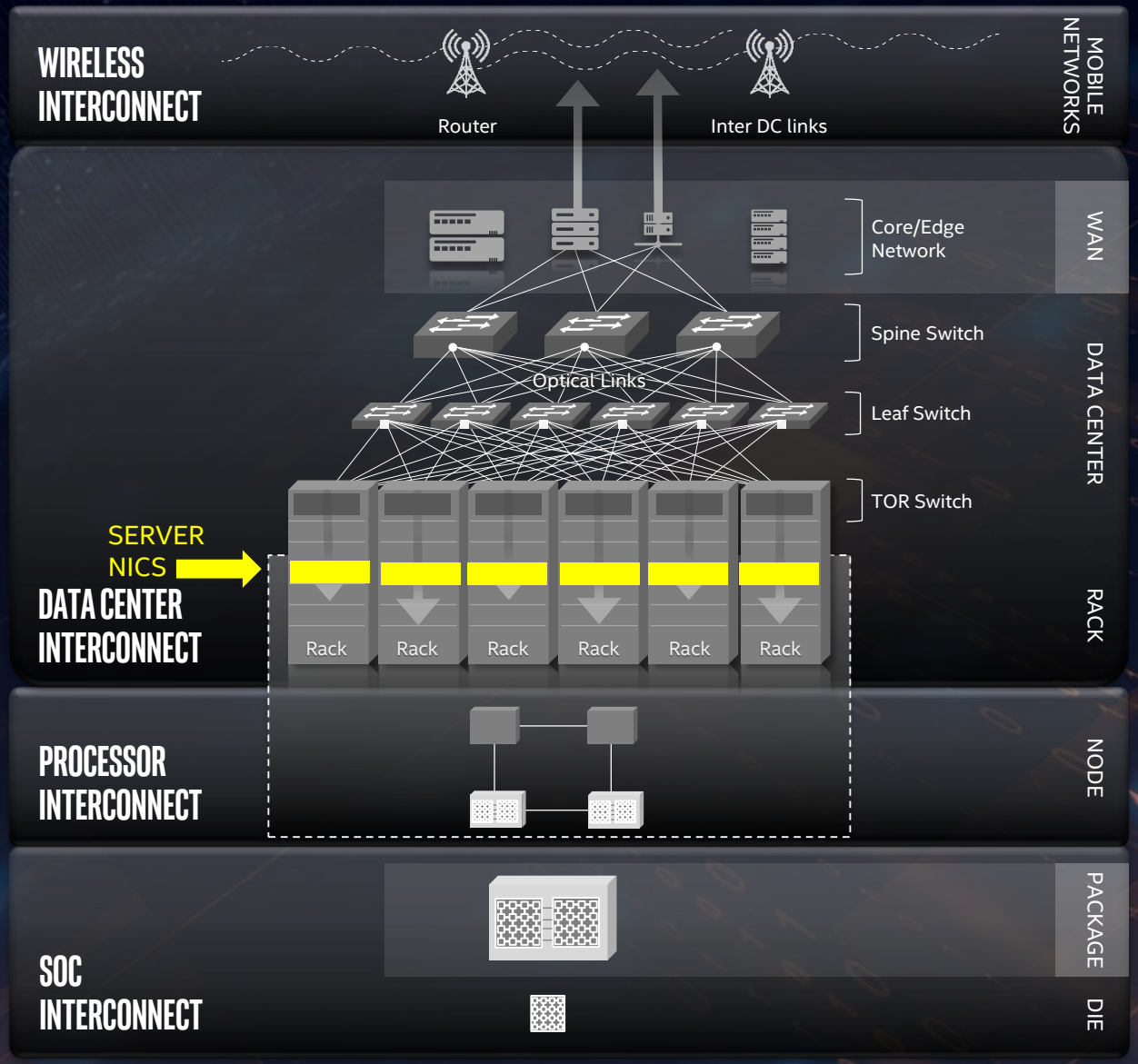
INTEL® ETHERNET 800 SERIES WITH APPLICATION DEVICE QUEUES (ADQ)

ANIL VASUDEVAN

SENIOR PRINCIPAL ENGINEER

APRIL 8, 2019

INTERCONNECT SCOPE



LONGER DISTANCE
HIGHER LATENCY
LOWER BW

SHORTER DISTANCE
LOWER LATENCY
HIGHER BW

THE GOAL

MOVE DATA FASTER

INTEL® ETHERNET 800 SERIES DELIVERS

IMPROVE APPLICATION PERFORMANCE WITH APPLICATION DEVICE QUEUES (ADQ)

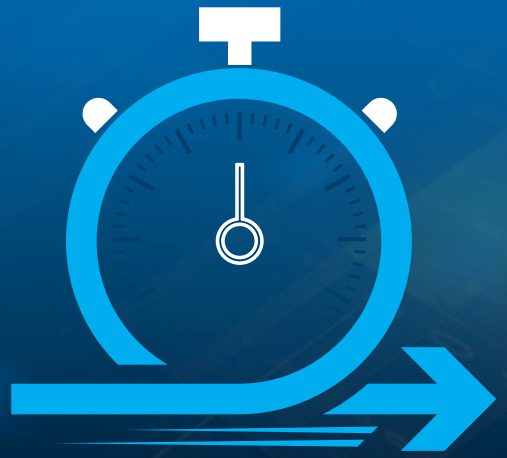
IMPROVE PACKET PROCESSING EFFICIENCY WITH DYNAMIC DEVICE PERSONALIZATION (DDP)

IMPROVE DATA MOVEMENT WITH CONCURRENT SUPPORT FOR IWARP AND ROCE V2 RDMA

SCALE-OUT APPLICATION PERFORMANCE PARAMETERS

PREDICTABILITY

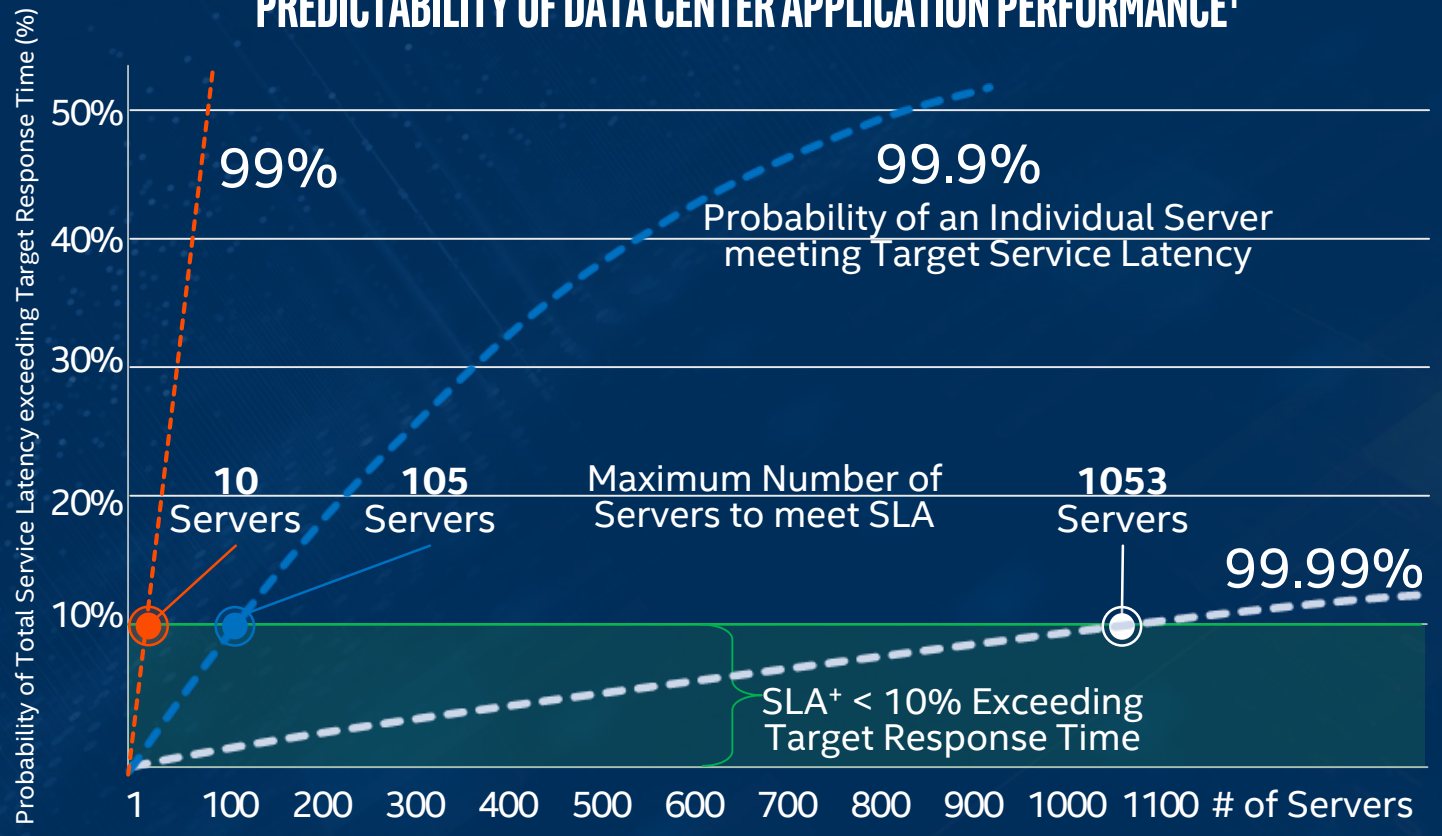
LATENCY



THROUGHPUT

WHY APPLICATION RESPONSE PREDICTABILITY MATTERS

PREDICTABILITY OF DATA CENTER APPLICATION PERFORMANCE[1]



Meeting the Scale-out Challenge

- Reducing variability in application response time (jitter) improves throughput and reduces latency

Benefits of Reducing Jitter

- More servers can be added to parallelize task
- Support more end-users with existing hardware

HIGHER PREDICTABILITY ENABLES MORE SERVERS WORKING IN PARALLEL WITHIN A DESIRED RESPONSE TIME

1. "The Tail at Scale" – Communications of the ACM. February 2013
Jeffrey Dean – Google Senior Fellow and Luiz Andrée Barroso – Google Fellow / VP of Engineering
https://cseweb.ucsd.edu/~gmporter/classes/fa17/cse124/post/schedule/p74-dean.pdf

+ SLA = Service-Level Agreement

HOW TO IMPROVE PREDICTABILITY
ANALOGY: TIME TO REACH AIRPORT FOR FLIGHT

APPLICATION DEVICE QUEUES (ADQ) IMPROVES PREDICTABILITY WITH DEDICATED LANES AND RATE LIMITING

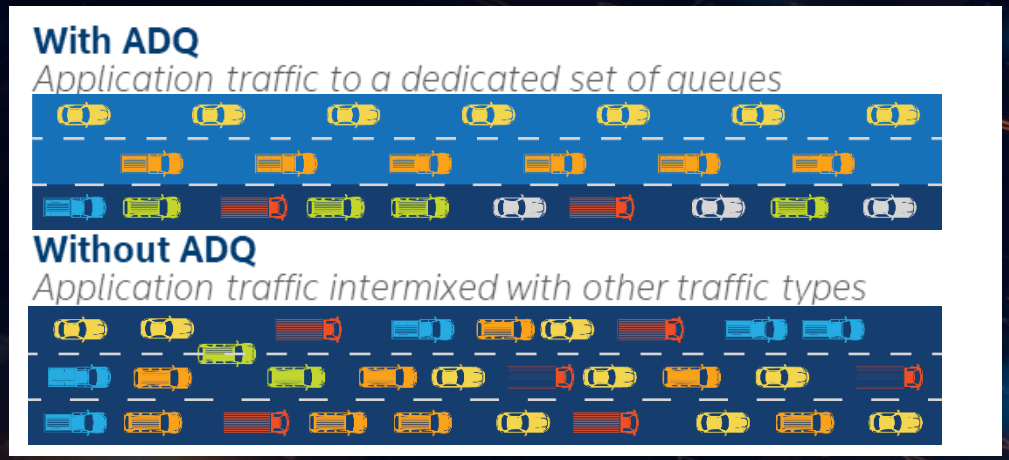
INTEL® ETHERNET 800 SERIES WITH APPLICATION DEVICE QUEUES (ADQ)[1]

What is ADQ?

- An application specific queuing and steering technology

How does ADQ work?

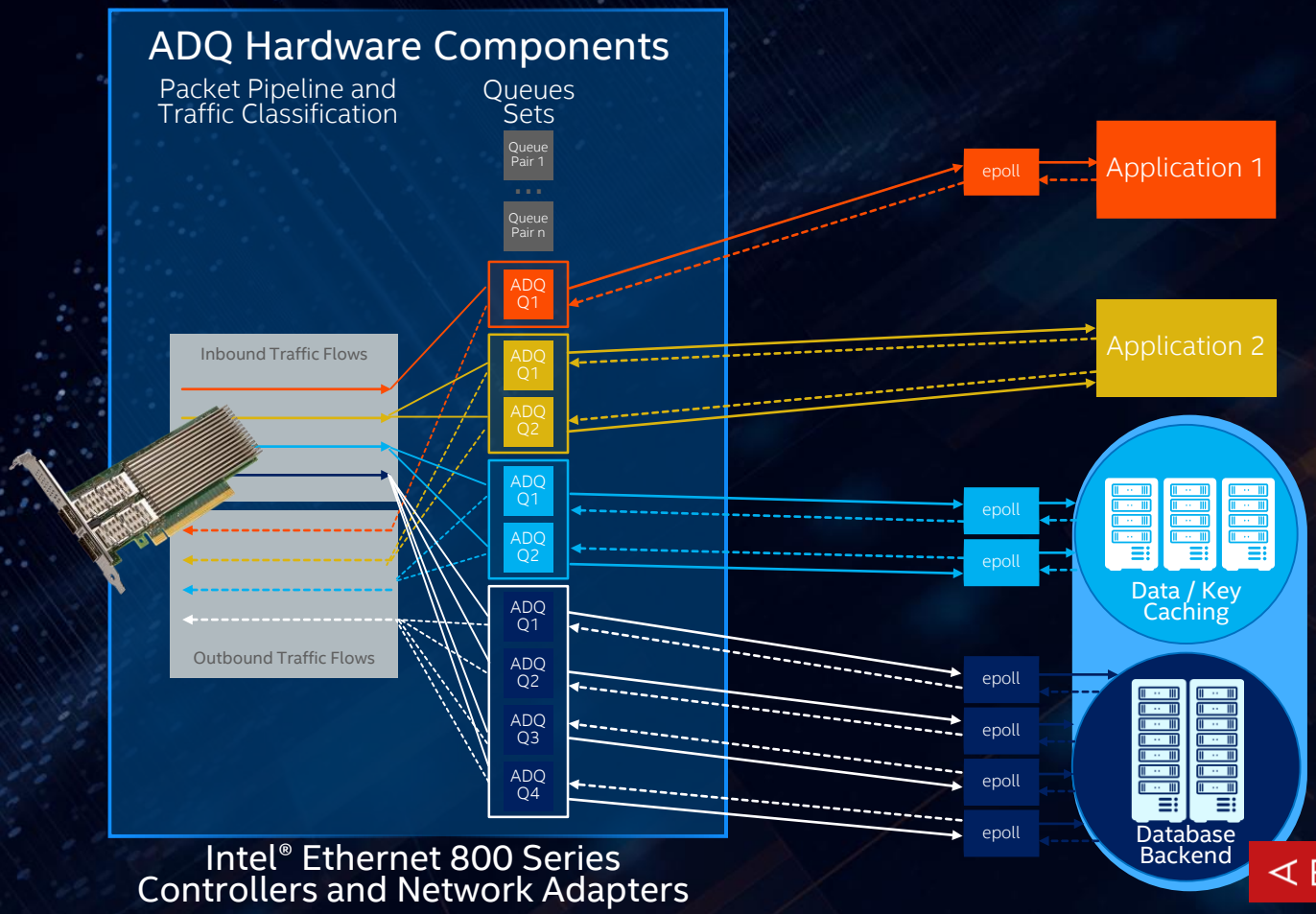
- Filters application traffic to a dedicated set of queues
- Application threads of execution are connected to specific queues within the ADQ queue set
- Rate limits application egress (Tx) network traffic



WHAT ARE THE BENEFITS OF ADQ?[2]

1. Features & schedule are subject to change. All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.
2. See Demo or Performance Testing Application Device Queues (ADQ) with Redis* Solution Brief

INTEL® ETHERNET – APPLICATION DEVICE QUEUES (ADQ)[1]
LATEST NETWORK TECHNOLOGY INNOVATION FOR INTEL® ETHERNET 800 SERIES



ADQ Basics

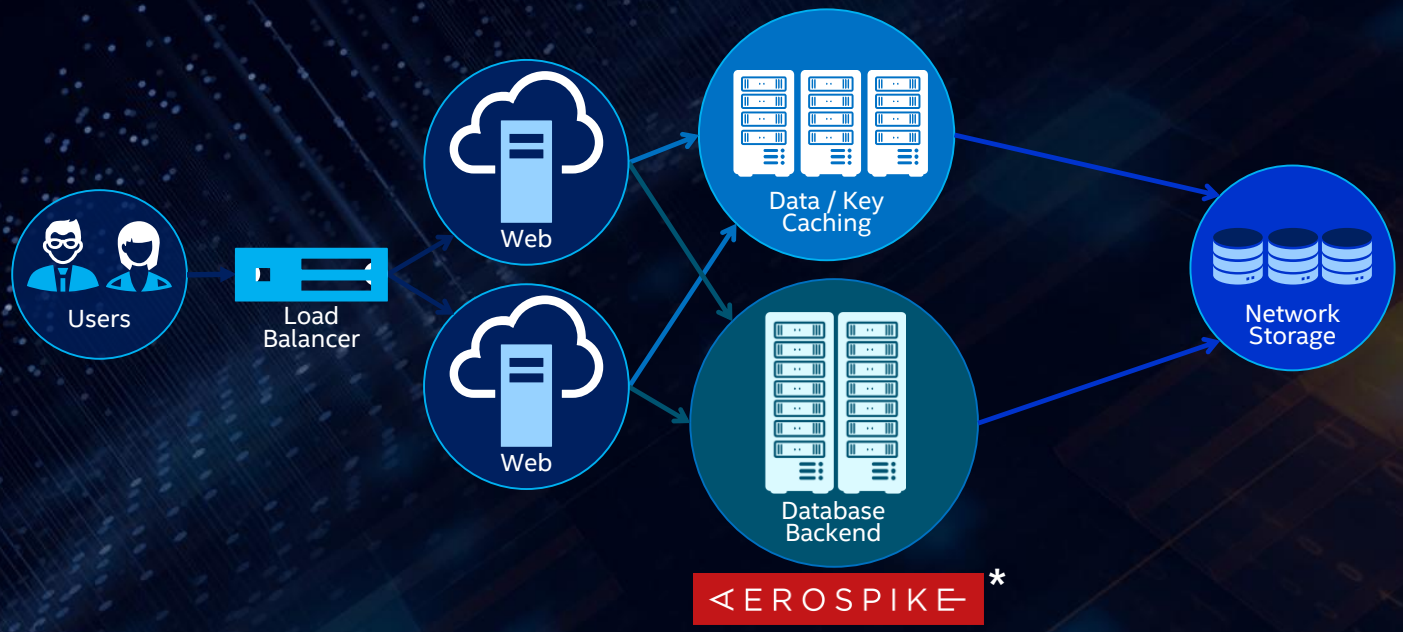
- Filters application traffic to a dedicated set of queues
- Application threads of execution are connected to specific queues within the ADQ queue set
- Rate limits application egress (Tx) network traffic

COMPONENT	REQUIREMENT
APPLICATION	Case1: No change (e.g. single instance) Case 2: Intel provided reference code
CONFIGURATION	Standard OS Tools (Traffic Configuration (TC), ethtool)
OS *	Linux 4.19 or later
ETHERNET DRIVER	Intel® Ethernet 800 Series Driver
ETHERNET NIC	Intel® Ethernet 800 Series

AEROSPIKE *

1. Features & schedule are subject to change. All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

APPLICATION DEVICE QUEUES (ADQ)[1]
PERFORMANCE IMPROVEMENTS ACROSS MULTIPLE TIERS



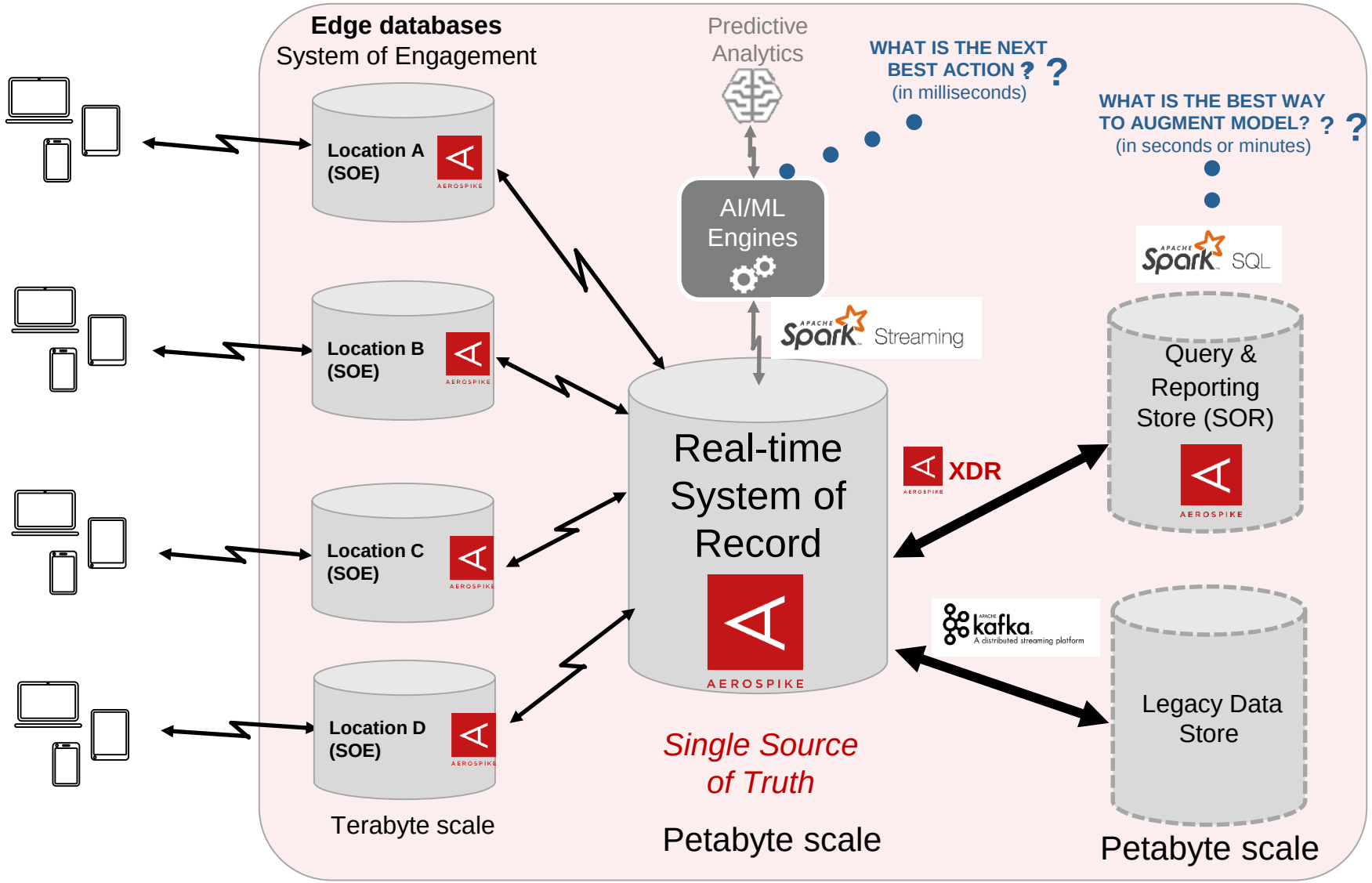
" Getting useful insights in real-time out of Big Data comes with a set of major challenges such as predictable low latency and maximum throughput at the network layer. Aerospike, as always, is at the forefront of addressing these challenges. We expect the Intel® Ethernet 800 Series with Application Device Queues (ADQ) coupled with Aerospike Enterprise will help get predictable performance, higher data throughput and lower latency. We are pleased to work closely with Intel to bring this exciting new technology to our customers."

AEROSPIKE Srini Srinivasan, Founder and Chief Product Officer

SIGNIFICANTLY IMPROVES PREDICTABILITY, LATENCY AND THROUGHPUT [2]

1. Features & schedule are subject to change. All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.
2. See Demo or Performance Testing Application Device Queues (ADQ) with Redis* Solution Brief

Aerospike real-time database ecosystem



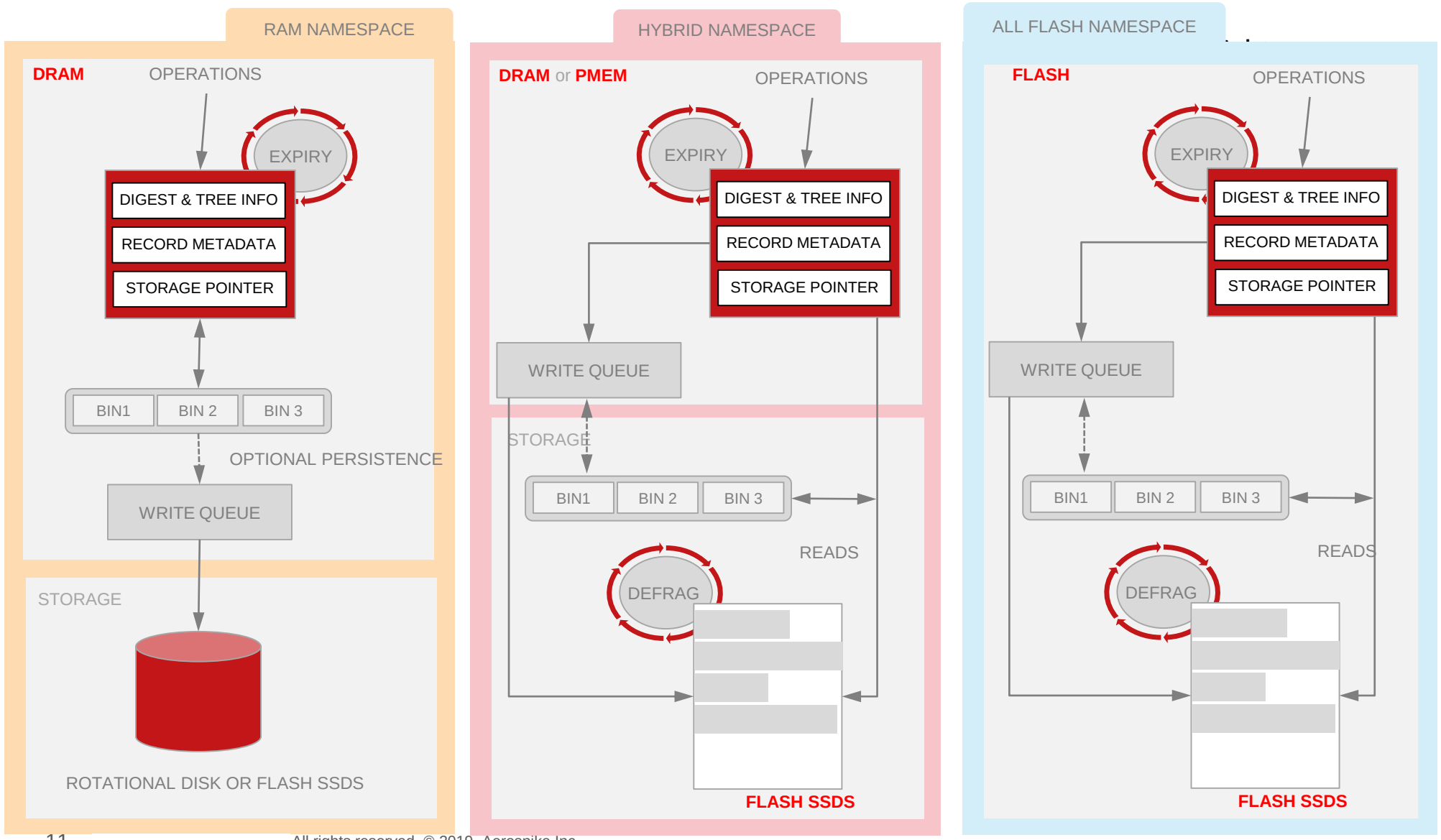
Consumers require customized real-time user experience

DB must deliver predictable performance at scale

- ✓ **Throughput > 1M TPS**
- ✓ **Latency < 1ms**
- ✓ **Reliability > five 9s**
- ✓ **Scale up to petabytes**
- ✓ **Strong consistency**
- ✓ **TCO < 2-5X of other DBs**

AEROSPIKE

Aerospike delivers speed at scale using Hybrid-Memory Architecture



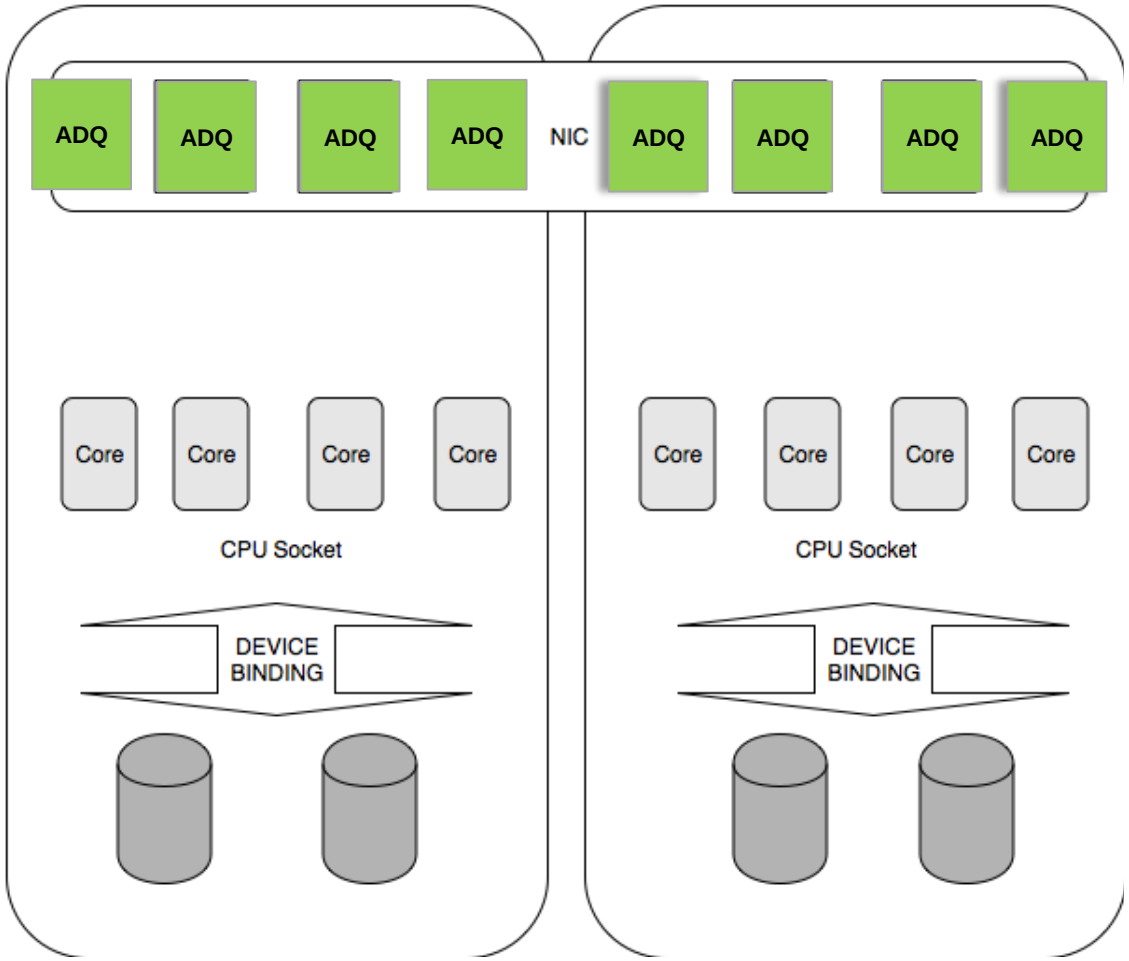
Highlights

1. Direct device access
2. Large Block Writes
3. Indexes in DRAM
4. Highly Parallelized
5. Log-structured FS “copy-on-write”
6. Fast restart with shared memory

AEROSPIKE

Aerospike is designed for wire-line speed

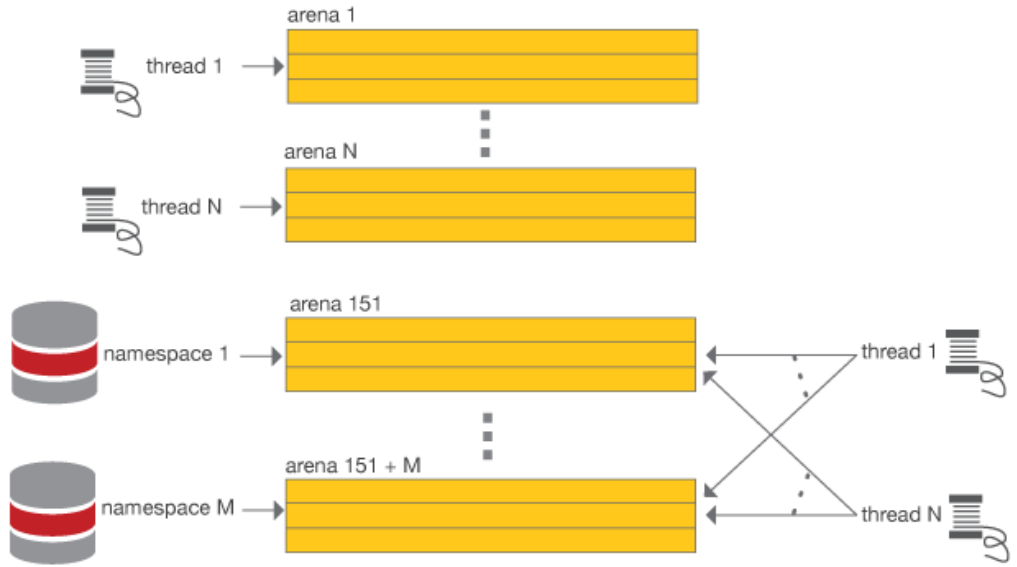
Multi-core Architecture



Optimized C based DB kernel

1. Multi-threaded data structures (NUMA pinned)
2. Nested locking model for synchronization
3. Lockless data structures
4. Partitioned single threaded data structures
5. Index entries are aligned to cache line (64 bytes)
6. Custom memory management (arenas)

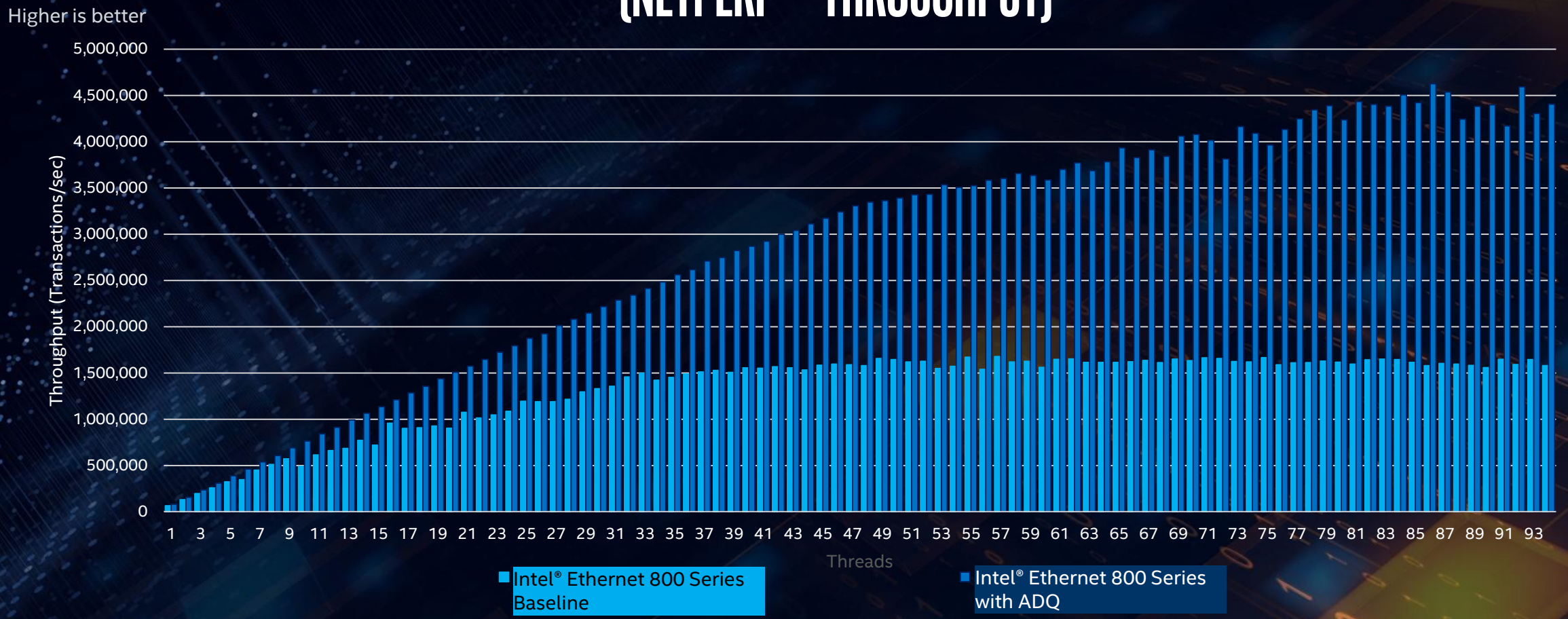
Memory Arena Assignment



AEROSPIKE

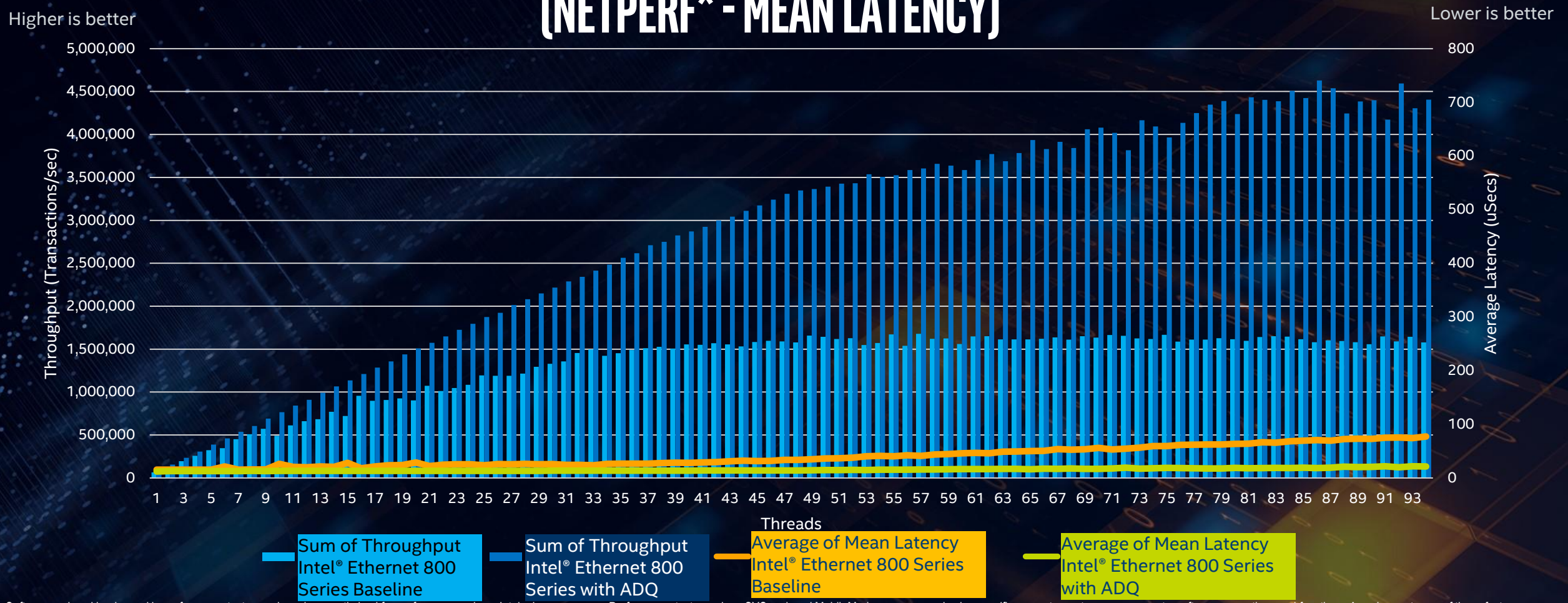
AEROSPIKE DEMO

INTEL® ETHERNET 800 SERIES BASELINE VS. WITH APPLICTION DEVICE QUEUES (ADQ)
(NETPERF* – THROUGHPUT)



Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance.
Source: Performance results are based on Intel internal testing as of March 2019, and may not reflect all publicly available security updates. See configuration disclosure for details. No product or component can be absolutely secure. Tests performed using Netperf 2.7.0 Open Source on 2nd Generation Intel® Xeon® Scalable processors with Intel® Ethernet 800 Series 100GbE on RedHat 7.5 L:inux (4.19.15.stable kernel) (see backup)

INTEL® ETHERNET 800 SERIES BASELINE VS. W/ADQ
(NETPERF* - MEAN LATENCY)

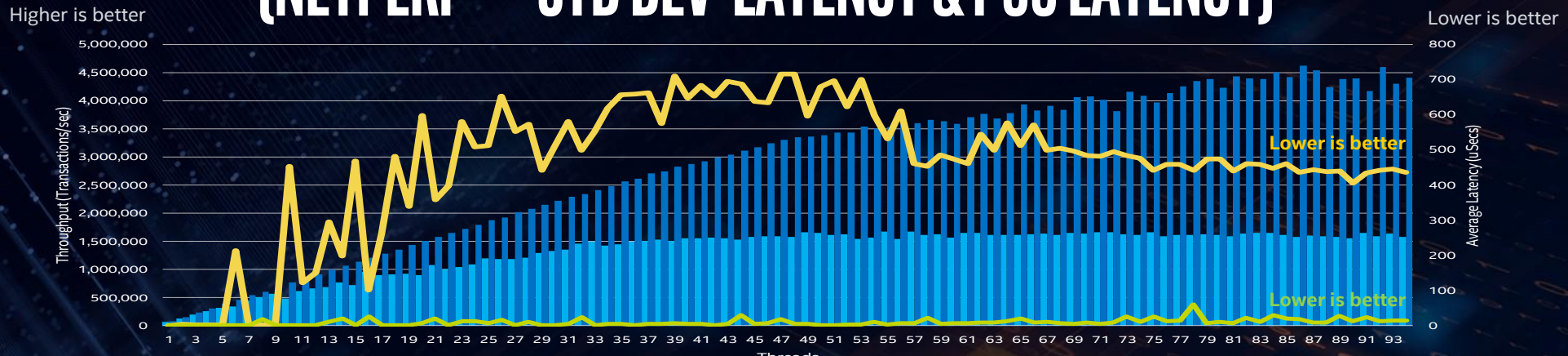


Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance.
Source: Performance results are based on Intel internal testing as of March 2019, and may not reflect all publicly available security updates. See configuration disclosure for details. No product or component can be absolutely secure. Tests performed using Netperf 2.7.0 Open Source on 2nd Generation Intel® Xeon® Scalable processors with Intel® Ethernet 800 Series 100GbE on RedHat 7.5 L:inux (4.19.15.stable kernel) (see backup)

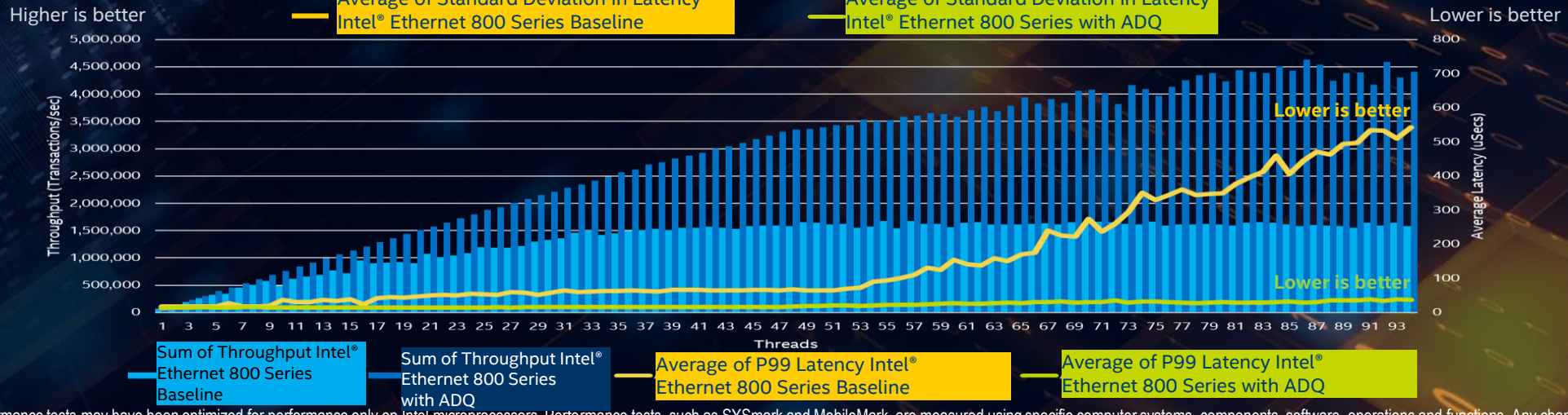
INTEL® ETHERNET 800 SERIES BASELINE VS. W/ADQ

(NETPERF* – STD DEV LATENCY & P99 LATENCY)

Standard Deviation



P99



Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance.
Source: Performance results are based on Intel internal testing as of March 2019, and may not reflect all publicly available security updates. See configuration disclosure for details. No product or component can be absolutely secure. Tests performed using Netperf 2.7.0 Open Source on 2nd Generation Intel® Xeon® Scalable processors with Intel® Ethernet 800 Series 100GbE on RedHat 7.5 L:inux (4.19.15.stable kernel) (see backup)

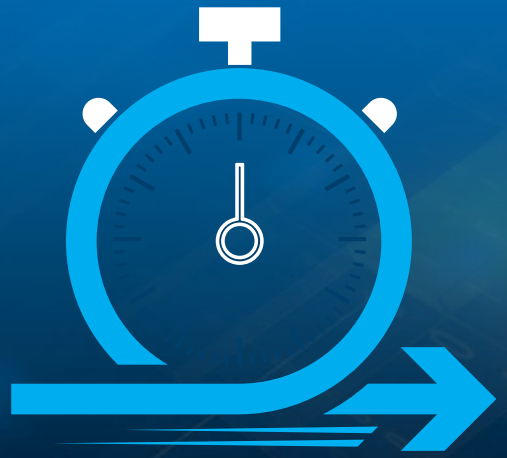
NETPERF DEMO

SUMMARY

INTEL® ETHERNET 800 SERIES WITH APPLICATION DEVICE QUEUES (ADQ) DELIVERS

PREDICTABILITY

LATENCY



THROUGHPUT

FAST, PREDICTABLE NETWORKING FOR FAST COMPUTE AND MEMORY

NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Performance results are based on testing as of February 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product or computer system can be absolutely secure. For more complete information visit http://www.intel.com/benchmarks.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo and others are trademarks of Intel Corporation in the U.S. and/or other countries. Apache, Spark, SQL , Kafka, the Apache Logos and others are trademarks of Apache Software Foundation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.


intel®

INTEL® ETHERNET ARCHITECTURE EVOLUTION

FOUNDATIONAL NICS

Intel® Ethernet
500 Series
Niantic

Intel® Ethernet
700 Series
Fortville

Intel® Ethernet A
800 Series
Columbiaville[1]

PRODUCTION Q3 '19

Capability Richness
More
Less

Fixed Pipeline

SR-IOV and VMDq

Partially Programmable Pipeline
- Table definition modifications with a Dynamic Device Personalization (DDP) profile package

Intel® Ethernet Adaptive Virtual Function (Intel AVF)

Queue and Steering Hardware Assists
- Application Device Queues (ADQ)

Fully Programmable Pipeline
- Table definition with DDP profile packages

Storage
- RDMA (iWARP* & RoCE*v2)

10GBE 40GBE 100GBE

INTEL(R) 800 SERIES BASELINE VS. INTEL® ETHERNET 800 SERIES WITH APPLICATION DEVICE QUEUES (ADQ) NETPERF* TEST CONFIGURATION

(PRE-PRODUCTION COLUMBIAVILLE AND PRE-PRODUCTION CASCADE LAKE)

SUT Configuration	
Platform	Intel® Server Board S2600WFTF
OS	Red Hat Enterprise Linux Server 7.5 (Maipo) (Kernel 4.19.15.stable)
Processor	Cascade Lake (Pre-production)
QPI/UPI Speed	10.4 GT/s
Memory Amt/Type/Speed	536 GB, 2400 MHz
Chipset	
BIOS Version	SE5C620.86B.0D.01.0134.091920181249
NIC/DUT	Intel® Ethernet X810 (“Tacoma Rapids”) A1
NIC/DUT Driver filename/version	ice.0.8.15, ice.0.8.15
TIC	521059
NIC EEPROM/NVM	0.00 0x80000941 255.65535.255
PCIe Generation	Gen 3
PCIe Lane Width	x16

Traffic Generator Configuration	
Platform	Intel® Server Board S2600WFTF
OS	Red Hat Enterprise Linux Server 7.5 (Maipo) (Kernel 4.19.15.stable)
NIC (Port/Speed)	Intel® Ethernet X810 (“Tacoma Rapids”) A1
SUT Virtualization Configuration	
Virt. Technology	N/A
# Host VMs	N/A
RAM & VCPU per VM	N/A
Guest OS	N/A
VCPU Affinity Configuration	N/A
Administrative	
Reviewed By	Daniel Maida
Reviewed Date	2/13/2019
Request ID	N/A

BIOS Version	Processor Configuration -> Hyper-Threading -> Enabled Processor Configuration -> Intel ® Virtualization Technology -> Disabled Power and Performance -> CPU Power and Perf Policy -> Performance Power and Performance -> Workload Configuration -> I/O Sensitive Power and Performance -> CPU P State Control -> Enhanced Intel (R) SpeedStep (R) Tech -> Enabled Power and Performance -> CPU P State Control -> Intel (R) Turbo Boost Technology -> Enabled Power and Performance -> CPU P State Control -> Energy Efficient Turbo -> Disabled Power and Performance -> CPU C State Control -> Package C-State -> C0/C1 State Power and Performance -> CPU C State Control -> C1E -> Disabled Integrated IO Configuration -> Intel (R) VT for Directed I/O -> Disabled System Acoustic and Performance Configuration -> Set Fan Profile -> Performance
OS/Kernel Changes	stopped and disabled: irqbalance, cpupower, firewalld, rngd, NetworkManager iptables -F SELINUX disabled scaling_governor set to performance for every core
Notes (Configuration to enabled ADQ)	Back-to-back setup with two X810-Tacoma Rapids A1 . Ice driver with compiled with ADQ flags. make -j $(nproc) CFLAGS_EXTRA='-DADQ_PERF -DADQ_PERF_COUNTERS' install BKC ADQ configuration for Netperf (Revision 1.0, February 11, 2019)

NETPERF* SYSTEM UNDER TEST AND CLIENT CONFIGURATIONS

	Config1 (Client)	Config2 (Server)
Test by	Intel	Intel
Test date	02/27/2019	02/27/2019
Platform	Intel® Server Board S2600WFTF	Intel® Server Board S2600WFTF
# Nodes	2	2
# Sockets	2	2
CPU	CLX K0 3.1 GHz 18C 24.75MB Cache, 205W, QQZ0	CLX K0 3.1 GHz 18C 24.75MB Cache, 205W, QQZ0
Cores/socket, Threads/socket	48	48
ucode	0x400000a	0x400000a
HT	On	On
Turbo	On	On
BIOS version	SE5C620.86B.0D.01.0134.091920181249	SE5C620.86B.0D.01.0134.091920181249
System DDR Mem Config: slots / cap / run-speed	2 slots / 568GB / 2400	2 slots / 536 GB / 2400
System DCPMM Config: slots / cap / run-speed	-	-
Total Memory/Node (DDR+DCPMM)	568 GB	536 GB
OS	RedHat 7.5	RedHat 7.5
Kernel	4.19.15.stable	4.19.15.stable
IBRS (0=disable, 1=enable)	1	1
eIBRS (0=disable, 1=enable)	0	0
Retpoline (0=disable, 1=enable)	0	0
IBPB (0=disable, 1=enable)	1	1
PTI (0=disable, 1=enable)	0	0
Mitigation variants (1,2,3,3a,4, L1TF)	1,2,3,3a,4, L1TF	1,2,3,3a,4, L1TF
Workload & version	netperf 2.7.0	netperf 2.7.0
Compiler	GCC	GCC