

Enabling Real-Time Processing of Massive Data Streams

Maintain excellent event stream latencies and sustained throughput by deploying Apache Kafka and Confluent Platform on Intel® processors, storage, and network technologies

- ✓ Apache Kafka is a distributed streaming platform used for data integration, real-time insights, and streaming analytics. From messaging, event sourcing, and monitoring to data processing and fault-tolerant storage, Apache Kafka empowers businesses with real-time data.
- ✓ Confluent Platform is a feature-enhanced distribution of Apache Kafka. It improves Apache Kafka with community and commercial features, and provides monitoring and a consolidated view for developers and site reliability engineers.
- ✓ Testing proves that 2nd Generation Intel® Xeon® Scalable processors, Intel® Optane™ SSDs, and Intel® Ethernet Network Adapter 800 Series with ADQ Technology are instrumental in enabling the low latency, high throughput, and data reliability necessary to execute demanding Apache Kafka streams processing.¹
- ✓ Intel's open source improvements in Java, Linux, and Apache Kafka deliver additional performance. Such capabilities are vital while the explosion of data sources and volumes are driving the need for greater ability to harness more data at real-time speeds.

[Learn about the Business Story →](#)

[Learn more about the Reference Architecture →](#)

Confluent Authors

Bert Hayes

Solutions Engineer

Eric Driscoll

Director of Strategic Programs,
US Public Sector

Josh Treichel

Senior Manager,
Partner Solutions Engineering

Intel Authors

Ankit M. Patel

Senior Component Design Engineer

Snehal Adsule

Cloud Solutions Engineer

Murali Madhanagopal

Enterprise Solutions Architect

Sandeep Togrikar

Senior Solutions Architect

Frank Ober

Data Center Solutions Architect

Business Story

Unleash Apache Kafka Performance with Confluent on Intel® Architecture

Solution Benefits

Open source Apache Kafka platform provides fundamental streams processing capabilities. Confluent's management tools simplify Apache Kafka deployment and improve platform performance, reliability, and scalability.

Developed jointly by Confluent and Intel, the Confluent Platform with Intel® technologies is a validated, high-performance, durable, and scalable solution:

- Up to 8 Gbps on commodity 10 GbE networking, and 76 Gbps on 100 GbE network with sub-50 ms latency
- Improved storage endurance 6.5x greater than standard NVMe storage when using Intel® Optane™ SSDs²
- Nearly linear scaling of the platform indicates that solution performance will not buckle as data loads and source system counts increase over time

Business Challenge: Enabling Real-Time Processing of Massive Data Streams

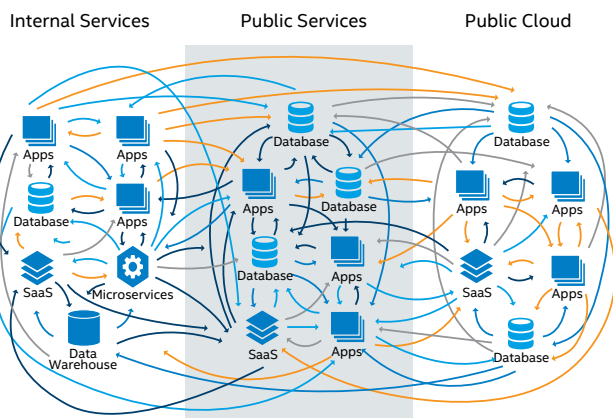
Data volumes continue to climb and the velocity of data handling continues to accelerate. Today, mission-critical applications need real-time analysis, but achieving that requires increasingly refined tools built on reliable, flexible, scalable platforms like Apache Kafka.

Data can move across any range of public or private services, and the web of interconnections between these services—from database archives to live software as a service (SaaS) applications to legions of IoT device feeds—can become a spaghetti architecture problem. This complexity can make solutions fragile and difficult to maintain. Apache Kafka can help decouple such data pipelines and simplify software architecture to make it manageable.

Eighty percent of Fortune 100 companies now use Apache Kafka for event streams processing,³ which is the handling and analysis of streams of event data to identify actionable patterns within those streams. Apache Kafka introduces a “broker” that manages traffic between data clients and producers (see Figure 1).

As with other open source platforms, managing Apache Kafka can be complex. Open source Apache Kafka is a foundation for message brokering and streams processing, but it can be streamlined by commercial implementations. To be scalable and reliably deliver real-time performance for modern workloads, Apache Kafka needs to run on a hardware foundation that can meet its demands for compute, network, and storage.

Conventional Data Exchange Approach



Decoupled Apache Kafka Data Exchange Approach

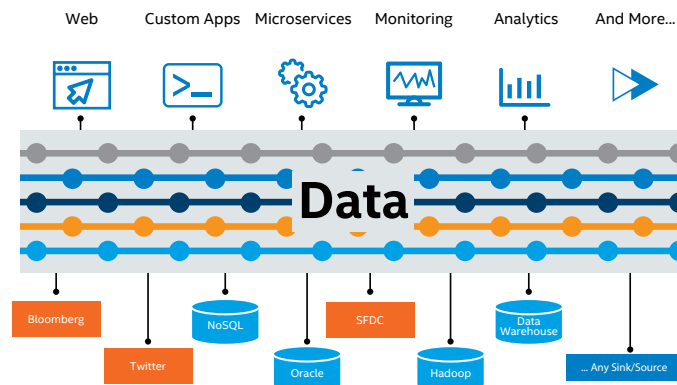


Figure 1. Conventional approaches to exchanging data between services and across infrastructures have grown to be quite cumbersome. This complexity introduces many potential problems, risks, and limitations. Apache Kafka decouples data producers from consumers and provides an optimized message broker exchange for stream traffic, especially when running on top of Intel® architecture-based systems.

Common Use Cases for High-Performance Streams Processing

Message brokers like Apache Kafka perform functions such as decoupling data pipelines from data production, queueing, processing, storage, and consumption. These are resource-intensive tasks, but when put into practice, a high-throughput message broker might do well in these use cases:

- **Real-time fraud detection.** Rather than performing batch analysis of transactions at the day's end and searching for anomalies, Apache Kafka streams processing allows for real-time aggregation and analysis of ATM and other transactions, allowing applications to identify anomalous events within seconds.
- **Facial recognition.** Surveillance based on live, high-resolution video feeds must be ingested before it can run through a detection model. The ingest and detect stages are separate use cases.
- **Anomaly detection and operational monitoring.** A SIEM tool such as Splunk is hungry for data that it can capture and correlate. Pre-built and vendor-supported integrations like the Splunk Add-on for Kafka⁴ and Confluent Splunk Sink Connector make it easy to collect telemetry data.
- **Mainframe-to-database offloading.** Mainframe systems remain a vital part of many organizations, and bridging their data to analytics platforms in real time such that "a single Apache Kafka broker can handle hundreds of megabytes of reads and writes per second from thousands of clients"⁵ can add tremendously to those mainframes' total value.
- **Cybersecurity monitoring.** By combining information storage and management with event management and analysis, organizations can realize near-real-time analysis of physical and network security footage and alert-streams to help alleviate the need for constant human supervision.

Apache Kafka Provides Low-Latency, High-Throughput Bolstered by Confluent

Apache Kafka is based on a publication-subscription (pub-sub) model. The software platform is a hub that intakes data streams from "producer" sources, organizes them into a range of "topics," and outputs them to "consumer" applications. As Figure 1 shows, this simplifies and streamlines the convoluted messaging architectures.

Apache Kafka offers simplicity, scalability, reliability, and security features to fit many use cases and data forms across platforms using Apache Kafka Connect API. It can support batch or real-time streaming message transfer and enable a loosely coupled use-case design wherein each data processing stage can scale independently of other stages. This eliminates tightly coupled applications and custom codes to improve manageability. The built-in monitoring features can help in cluster administration. Confluent builds on top of Apache Kafka, providing more than the core message queuing functionality of Apache Kafka. Confluent Platform adds tools for cluster management, including additional security and supports an extensive list of [connectors](#). Confluent also emphasizes reliability and the organization of data from multiple sources and source types.

Confluent Platform

Confluent Platform provides key commercial and community features across several areas:

- **Confluent Control Center.** Enables intuitive and streamlined Apache Kafka management and monitoring.
- **Confluent Replicator.** The replicator makes it easy to maintain multiple clusters in multiple data centers by managing replication of data and topic configuration between locations.
- **Multi-region Clusters.** Large-scale deployments run across multiple data centers and multiple geographic regions. This feature mitigates high latency, low throughput, and increased costs when producing and consuming messages.
- **Confluent JMS Client.** Implements the JMS 1.1 standard API, using Apache Kafka brokers as the backend. This is useful when replacing JMS message brokers so existing applications can integrate with a modern streaming platform without requiring major application rewrites.
- **Streams Processing and ksqlDB.** Provides an interactive SQL interface for streams processing, without the need to write code in a programming language such as Java or Python.
- **Tiered Storage.** Starting with Confluent Platform 6.0, this feature makes storing large volumes of data in Kafka manageable by moving warm data to s3 object stores and scaling brokers only when more resources are needed.

Confluent Platform also includes role-based access control and broad support for on-premises deployments.

Intel® Technology Further Boosts Performance

Intel provides technologies across networking, storage, and central processing that can enhance data center performance and solution value within an Apache Kafka/Confluent streaming context (see Figure 2).

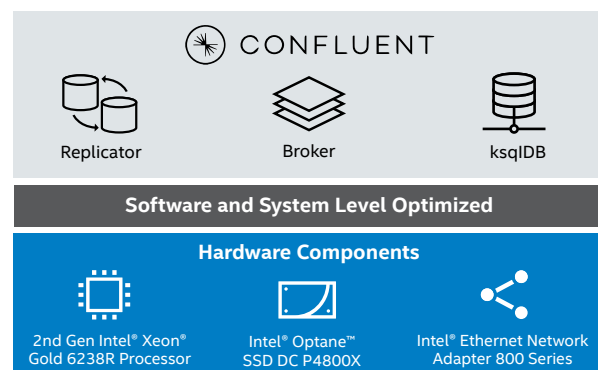


Figure 2. Confluent Platform provides features that go beyond open source Apache Kafka to provide additional performance, reliability, and scalability.

A Confluent Platform, running on Intel® hardware featuring a 10 Gbps network connection, can sustain up to 8 Gbps throughput with low latency. Intel testing shows a performance ingestion rate of up to 86 TB of data per day.⁶ And because this data was stored on Intel Optane SSDs, which deliver up to 6.5x⁷ greater reliability than conventional 3D NAND SSDs, data durability and high availability were significantly enhanced. The solution can be scaled up by

upgrading to 100 GbE Intel® Ethernet Network Adapter 800 Series to achieve up to 10x network throughput, while maintaining lower latencies (see [Table 3](#)).

Better Together: Confluent and Intel

While Confluent and Intel share complementary technologies, they also share common goals. Both companies pursue three strategies to deliver performance-optimized event streaming for a compute-everywhere world. This includes:

- **Build New Solutions.** Yesterday's products and approaches are only a good starting point. Confluent and Intel collaborate to develop and validate on-premises and cloud-based streams processing solutions. This cooperation makes it easier for customers to start using solutions and provides a clearer idea of what performance and benefits to expect.
- **Contribute to the Open Source Community.** The Confluent team created Apache Kafka for the open source community, and Intel's history of open source contributions is long and broad. For example, Intel's contributions to Java Development Kit 15 yielded a 12 percent latency improvement and 20 percent throughput improvement.⁸
- **Build Ecosystem Relationships.**¹ Confluent and Intel help OEMs, system integrators, cloud service providers, and other groups receive similar development and support experiences regardless of the solution deployment mode.

Solution Architecture: Fast and Scalable Enterprise Streaming

Confluent Platform provides the foundational tools and support of open source Apache Kafka while adding several management capabilities and broader functionality. The process of producers publishing records tagged to a given topic and consumers subscribing to topics involves extensive message flow, often across multiple clusters and different data centers. Therefore, Apache Kafka deployments require low latency, high throughput, and fast processing so those records keep streaming between points and through applications at near-real-time levels.

Figure 3 shows the value-add components that Confluent Platform provides to make Apache Kafka enterprise-ready. Confluent features enable organizations to deploy Apache Kafka with high performance and reliability. However, these features deliver the most benefit when running on current, data center-optimized Intel® compute, storage, and networking technologies. The reference architecture for Confluent Platform incorporates the following Intel technologies:

- **2nd Generation Intel Xeon Scalable processors.** They are built for enterprise applications. They incorporate architecture improvements for compute- and data-intensive workloads and are well suited for ingesting and analyzing massive data quantities.
- **Intel Optane SSDs.** These SSDs bring a middle ground to the enterprise memory and storage hierarchy. They provide the low-latency benefits of DRAM, the high capacity of NAND SSDs, and better endurance characteristics than 3D NAND SSDs can offer.⁹ Intel Optane SSDs often serve in caching capacities or as overflow for capacity-bound in-memory applications.

- **Intel® 3D NAND SSDs.** These PCIe/NVMe-based SSDs deliver scalable, cost-effective performance and low latency. They also offer outstanding quality, reliability, manageability, and serviceability to minimize service disruptions.
- **Intel® Ethernet network products.** Intel Ethernet network controllers, adapters, and accessories enable agility in the data center to deliver services efficiently and cost-effectively. Intel® Ethernet 800 Series Controllers with ADQ Technology showcase these benefits, providing up to 100 Gbps bandwidth and novel functionality to improve packet processing performance. They organize and prioritize traffic queueing for faster response times, higher predictability, and greater scalability. Compatible with the Open Compute Platform, these connectors support high throughput, reliability, and compatibility.

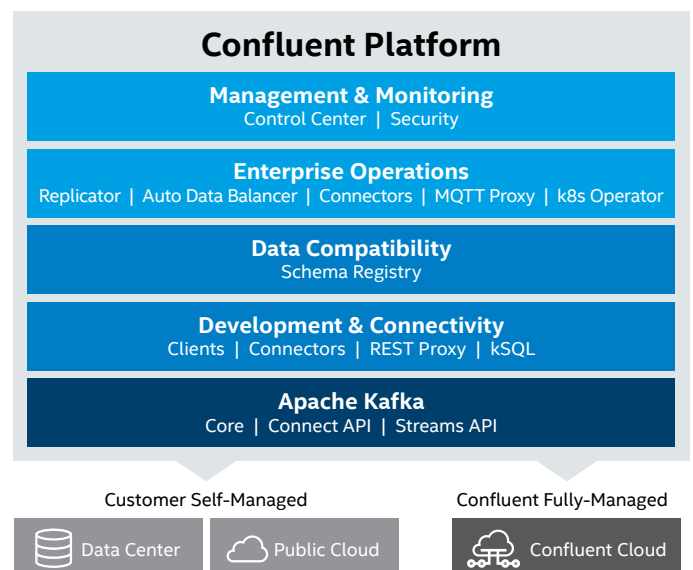


Figure 3. Apache Kafka can sit at the nexus of cutting-edge data sources and real-time applications putting that data to work. Much of Apache Kafka's core functionality is freely available, but Confluent builds many tools and enhancements onto this core.

Conclusion

The case for streams processing grows increasingly pressing as businesses require real-time responsiveness in meeting market demands. Apache Kafka is now the go-to streams processing solution for businesses deploying on-premises, across the cloud, or a mixture of the two. For those implementing Apache Kafka in enterprise settings, Confluent emerges as the default choice for maximum functionality and platform support. And to attain high performance, dependable results from the Confluent solution (regardless of deployment environment), a hardware foundation comprised of Intel technologies can deliver excellent results.

Ready to learn more? Turn the page for a detailed Reference Architecture discussion.

Reference Architecture

Unleash Apache Kafka Performance with Confluent on Intel® Architecture

Summary of Findings¹⁰

- Improves network throughput up to 20 percent when using a modern NIC, such as the Intel Ethernet Network Adapter 800 Series with ADQ Technology
- Up to 5x producer and 9.1x consumer throughput compared to a 2014 benchmark by moving to the latest 2nd Gen Intel® Xeon® Scalable processors, networking, storage, and the latest Confluent Apache Kafka
- Up to 6.5x improved endurance¹¹ with Intel® Optane™ SSDs versus 3D NAND SSDs on Apache Kafka broker storage
- Linear scaling of throughput and compute as network bandwidth increases from 10G to 100G
- Open source Java JDK 15 improvements in Apache Kafka of roughly 20 percent
- Optimal application settings for network I/O threads and reliability for best performance at 10/100G network capacity

Table of Contents

Confluent/Apache Kafka Reference Architecture Overview	5
Test Results and Analysis	6
Historical vs. Current Benchmark Results	6
Boost Applications with Modern Network (100 GbE)	7
Higher-Endurance SSDs Can Lower System Total Costs	8
JDK Performance Improvements to Apache Kafka	8
Intel® Technology Significantly Improves Apache Kafka Performance	9
CPU-driven Benefits	9
Storage-driven Benefits	9
Network-driven Benefits	9
Apache Kafka Cluster Sizing Guidelines: Scale Up, Then Scale Out	9
Conclusion	10

Confluent/Apache Kafka Reference Architecture Overview

In the solution brief, [Figure 3](#) illustrates how Confluent's platform builds on and extends the capabilities of Apache Kafka. [Figure 4](#) provides an architectural view of how a Confluent/Apache Kafka cluster can be built for multi-DC deployment. This section describes how the selection of compute, storage, and networking resources, along with key software configuration attributes, shapes application performance across multiple metrics, and helps with disaster recovery.

In this model, two regions (DC-1 and DC-2) provide high availability. DC-1 (Leader ISR) is the active, working collection of brokers. DC-3 keeps a quorum in case of DC failure. DC-2 (Observer Replicas) remains on standby, ready to pick up DC-1's operations if a service interruption occurs. In both cases, "ZooKeeper" systems serve as coordinating nodes. If a failure occurs, a ZooKeeper seamlessly makes the other partition become the leader.

The streams processor and control-center systems complete the second and third pieces of this Confluent solution, which is a resilient, highly scalable design appropriate for a wide array of Apache Kafka and stream-processing workloads.

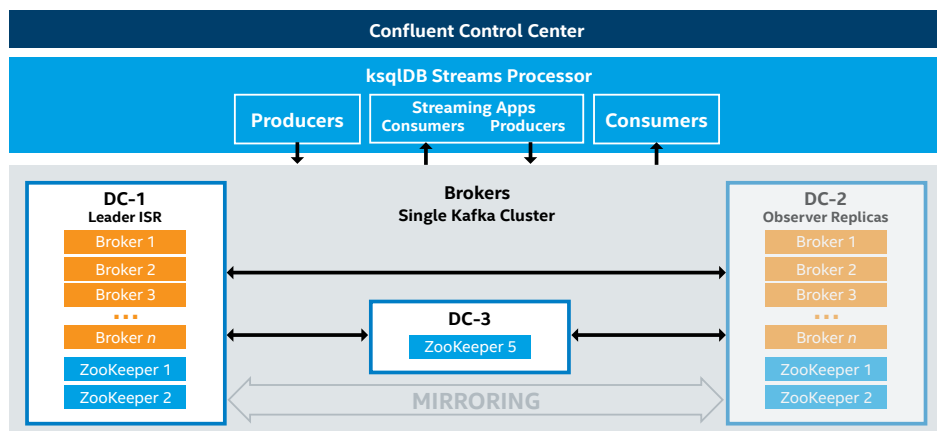


Figure 4. A fully configured Confluent Platform founded on Intel® technologies carries homogeneous systems across the architecture: right mix of compute, high-end storage, high levels of broker memory, and a strong network backbone.

Table 1. Basic bill of materials for systems configuration

	Confluent Control Center (1 node)	Confluent ksqldb + Kafka Connect + Confluent Schema Registry (2 nodes)	Brokers + Zookeeper (3 nodes)
Processor	2-Socket 2nd Gen Intel® Xeon® Gold 6238R processor (28 cores)	2-Socket 2nd Gen Intel Xeon Gold 6258R processor (28 cores)	2-Socket 2nd Gen Intel Xeon Gold 6238R processor (28 cores)
Memory	384 GB DDR4 (12x 32 GB)	384 GB DDR4 (12x 32 GB)	192 GB DDR4 (6x 32 GB)
Network	Intel® Ethernet Network Adapter E810-CQDA2 (10/100G)	Intel® Ethernet Network Adapter X722 (10G)	Intel® Ethernet Network Adapter X722 (10G) or Intel Ethernet Network Adapter E810-CQDA2 (100G recommended)
Storage	OS: 1x 480 GB Intel® SSD DC S4510 Storage: 2x 4 TB Intel® SSD DC P4510	OS: 1x 480 GB Intel® SSD DC S4510 Storage: 2x 4 TB Intel SSD DC P4510	OS: 1x 480 GB Intel® SSD DC S4510 Storage: 4x 750 GB Intel® Optane™ SSD DC P4800X ^a or 4x 4 TB Intel SSD DC P4510 (3D NAND)

^a With the Confluent Platform 6.0 tiered storage feature, you can use a high-performance enduring Intel Optane storage locally on the broker and use the s3 object storage to reduce costs. For more information, see <https://docs.confluent.io/current/kafka/tiered-storage.html>.

Table 1 details the key components of our Confluent/Apache Kafka solution. The selection of Intel compute, storage, and network components was central to the performance gains observed during testing.

Test Results and Analysis

Historical vs. Current Benchmark Results Show System Components are Critical

A benchmarking test was performed six years ago to assess Apache Kafka performance.¹² The test platform ran on the commodity hardware compute, storage, and networking components for that time. Because the importance and value of Apache Kafka has increased since then, Intel and Confluent reran those same benchmark tests using the 2nd Gen Intel Xeon Gold processors, Intel Optane SSDs, and 100G Intel networking components more appropriate to current workload processing. These components offer many benefits for Apache Kafka and data-centric enterprise application workloads:

- **Intel Xeon Gold 6238R processor.** With 28 physical cores, each of which can run two logical threads, a dual-processor platform provides 112 threads using Intel® Hyper-Threading Technology. Additionally, the Intel Xeon Gold processor family offers more interconnect and accelerator engines than the Intel Xeon Silver processor or Intel® Xeon® Bronze processor options, which suits this platform's performance needs. Having 112 threads addresses the performance, efficiency, and licensing needs of this application.
- **Intel® Optane™ DC SSD P4800X.** This suits applications involving low latency needs and high volumes of data traffic. These SSDs consistently offer lower latency results compared to their 3D NAND storage-based counterparts while offering high endurance.¹³
- **Intel® Ethernet Network Adapter E810-CQDA2.** The move from 10G to 100G Ethernet networking offers a massive increase in connectivity bandwidth and many advantages, including technologies aimed at driving down latency to benefit a range of cloud-based architectures.

Through extensive testing, engineers identified the configurations and thread tuning needed to deliver optimal results. Those are presented here as a reference guideline.

Platform Throughput

Researchers achieved a 5x increase (producer throughput) when comparing producer performance of the 2014 configuration against an optimized 2020 configuration (see Figure 5). Similarly, throughput jumped by 9.1x for consumer performance.¹⁴ Performance in both groups more than doubled by leveraging Intel hardware's faster compute, storage, and network, and software tuning for the latest Apache Kafka for the 2020 configuration. The key aspects of this tuning are examined below.

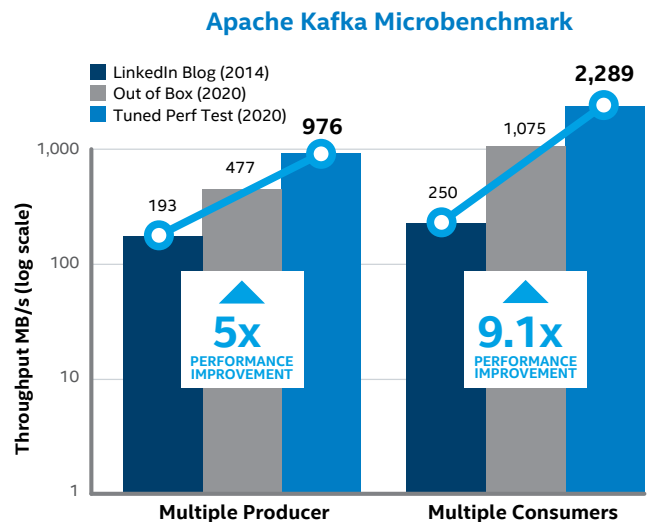


Figure 5. Apache Kafka performance testing showed significant throughput benefits from adopting modern Intel® architecture components across compute, storage, and networking.¹⁵

Observed throughput scaled linearly with the number of producers under test—up to a certain point. For example, two producers delivered 976 MB/s (488 MB/s per producer) across all thread counts. Adding I/O and network threads did not help accelerate the data processing of two producers. Fifteen producers yielded 7,324 MB/s across all thread counts—488 MB/s per producer again. This pattern held across producer counts of 2, 5, 10, 12, 14, and 15. Only at 20 producers did researchers observe variance in performance, with 110 threads delivering the expected 9,975 MB/s (488 MB/s per producer). Throughput scales

evenly with producers and is not influenced by thread counts when working across three broker nodes.

Boost Applications with Modern Network (100 GbE)

The performance story changes upon examining latency. Figure 6 details how and when latency increases as the number of producers, and thus the amount of system throughput increases.¹⁶

In configuring the test cluster, engineers evaluated a wide range of I/O and network thread count options. They generally found that a ratio of 15 (I/O) to 8 (network) threads was ideal for 10 GbE tests, and 80 (I/O) to 30 (network) threads for 100 GbE tests. Note that in Table 2 the totals stay within each system's 112-thread compute limit.

Table 2. Network thread count options

	Number of Threads						
I/O	15	30	45	60	70	80	90
Network	8	16	24	32	40	30	10
Total	23	46	69	92	110	110	110

Intel measures tail latency (that is, 99th percentile latency) as a major performance indicator instead of mean latency. We were able to "scale up" data ingestion by increasing the number of producers and higher network bandwidth. Each producer is producing 500 MBps network throughput; therefore, we were able to scale the solution to 15 producers with less impact on latency. However, with 20 producers producing 10 million events/second throughput, latency was dramatically increasing. Intel recommends "scaling out" the solution with an additional broker.

Within the latency test results, another interesting observation developed that underpins why adhering to the

configuration recommendations is important. The interplay between specific thread counts and observed latency at those settings can be striking. For example, with a 70:40 I/O-to-network thread ratio, 14 producers offer a 12 ms latency. An 80:30 mix yields 20 ms latency. A 60:32 ratio for 14 producers, however, showed latency many multiples higher. A minor adjustment in thread configuration can create remarkable differences in results. Intel and Confluent exerted considerable effort of trial and error in testing so businesses don't have to.

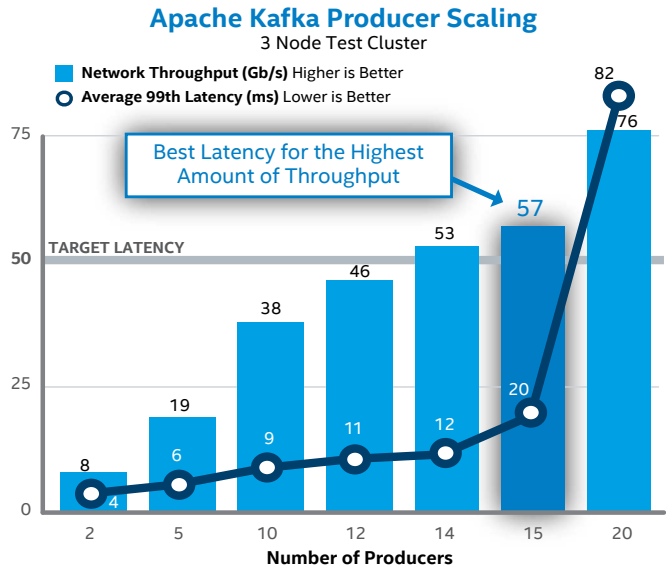
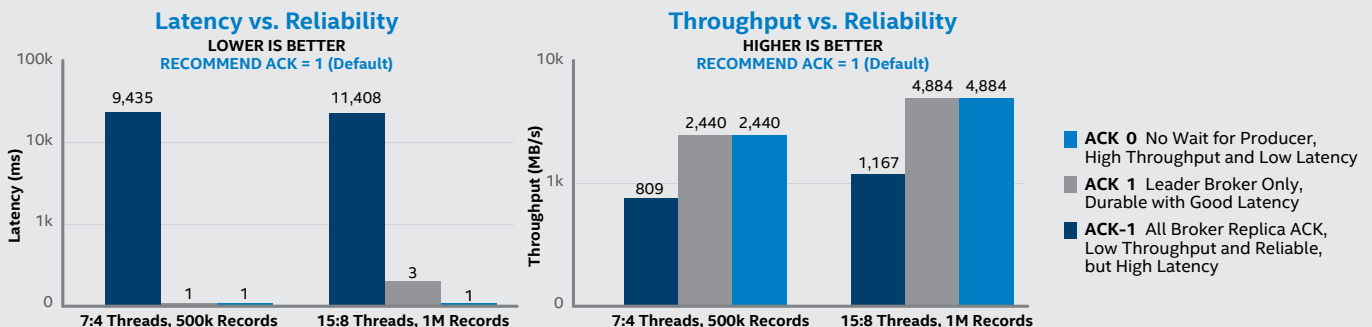


Figure 6. While Apache Kafka throughput performance scales linearly with increasing numbers of producers, latency remains relatively flat until it increases sharply once producer counts reach a certain level.¹⁷

Tuning the System: ACK vs. Latency and Throughput

ACK commands signify when one process or system receives a message from another. Acknowledgments (ACK) and negative acknowledgments allow a sender to know a receiver's state and respond accordingly. ACK codes are one setting that administrators can adjust in an Apache Kafka configuration. This setting can have a marked impact on platform latency and throughput. ACK codes assist with establishing reliability in network communications, but the specific ACK configuration can influence throughput and, to a much greater degree, latency. An ACK setting of -1 should be used carefully and only for mission-critical applications.



Higher-Endurance SSDs Can Lower System Total Costs

Intel Optane SSDs are widely recognized for exhibiting latency response times that are better than conventional 3D NAND storage. These SSDs are well-suited to applications needing high volumes of rapid transactions, especially those demanding real-time performance.

Intel® Optane™ storage offers significantly higher endurance than 3D NAND storage, which in turn can have a profound impact on the viability and ongoing costs of an Apache Kafka solution deployment.

Although endurance can be measured in different ways, Intel assessed it in terms of lifetime drive writes per day, if every cell on the SSD were written to 60 times per day, measured in petabytes written. Under throughput levels (see Figure 7), Intel Optane storage offers roughly 6.5x greater endurance (lifetime durability) than enterprise-class 3D NAND storage. For example, with a daily throughput of 43.2 TB per day, 3D NAND storage showed an endurance of 1.05 years compared to 6.86 years for Intel Optane SSDs. This means far less storage downtime and an accompanying drop in data center operational maintenance costs. Depending on the workload, 3D NAND SSDs may have to be replaced every year, but Intel Optane SSDs will not.¹⁸

Intel® Optane™ SSDs vs. 3D NAND SSD Endurance

HIGHER IS BETTER

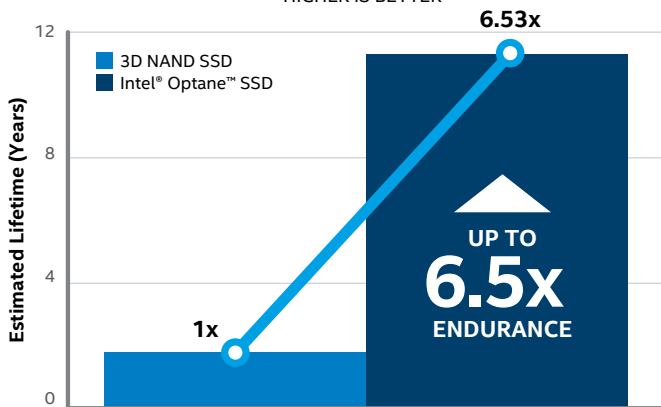


Figure 7. Intel's Confluent/Apache Kafka solution tests revealed that Intel® Optane™ storage would deliver several years of additional operation beyond what similar 3D NAND storage would provide under the same strenuous write loads.¹⁹

JDK Performance Improvements to Apache Kafka

For many years, Intel has been an integral part of the open source community. The company has made many open source contributions, both in the Linux and Java spaces. That includes helping JDK version 15 reach general availability in September 2020.

To show the importance of keeping software elements properly patched and configured, Intel tested a system based on two Intel® Xeon® Platinum 8180 processors and 100 GbE Intel® Ethernet 800 Series Controllers. Incoming and outgoing traffic was generated by three systems using default settings from the Apache Kafka workspace. The Apache Kafka server was configured for two brokers with 12 network and 16 I/O threads each.²⁰ The only difference in the two datasets was whether the platform had been patched to the new [JDK release](#).

All results for the non-patched configuration were normalized to a value of 1. The averaged results for the patched configuration realized a 9.6 percent gain in transmit throughput and a 21 percent gain in receive throughput. Additionally, average latency dropped by over 12 percent and P99 latency by just over 20 percent. This represents an impressive performance gain that arrives at no cost, simply by keeping software patches current.

JDK Performance Improvements to Apache Kafka

	Average Receive	Average Transmit	Average Latency	0.99 Latency
Non-Patched Build	1	1	1	1
Patched Build	1.21	1.09	0.88	0.79

Recommendations: Intel® Technology Significantly Improves Apache Kafka Performance

CPU-driven Benefits

As described earlier, the high physical and logical core counts offered by the 2nd Gen Intel Xeon Gold processor family, and in particular the 2.2 GHz Intel Xeon Gold 6238R processor used here, paired well with the I/O and network thread counts assessed in the Confluent/Apache Kafka cluster workload. The Intel Xeon Gold processor family, designed for two- to four-socket platforms, is optimized for high memory speed, capacity, and interconnects. This suits the performance and load demands common in real-time analytics and the decoupling work common to Apache Kafka applications. The processing power of the Intel Xeon Gold 6238R processor was directly responsible for this testing platform being able to maintain data latency levels at or below 20 ms in the face of 15 Apache Kafka producers, and it sustained network throughput in excess of 50 Gbps.

Storage-driven Benefits

Intel Optane storage is highly advantageous because of its 6.5x endurance benefit over 3D NAND storage. For write-intensive data ingestion workloads like Apache Kafka, endurance can help improve overall operational cost in data centers.²¹

With the Confluent Platform 6.0 tiered storage feature, high-performance Intel Optane storage can be used locally on the broker, and the cost-effective s3 object storage can be used for high volume data.

Network-driven Benefits

Using a highway metaphor best conveys why 40G, 50G, or 100G networking can be advantageous in an Apache Kafka environment. One Apache Kafka broker may exceed the bandwidth of a 10 GbE connection, just as heavy traffic can clog the capacity of a single-lane road. A 100 GbE infrastructure requires greater capital expenditure than a 10 GbE one, but it can provide far more capacity for future-proofing or, alternatively, the ability to tackle much greater workloads today. The Intel Ethernet Network Adapter E810-CQDA2 provided more bandwidth than needed for the three-broker cluster tested here. However, that allowed researchers to mimic the performance levels typical of 25 and 40 GbE connections and determine how far configurations and settings could be pushed for optimal benefit. With features specifically tailored to minimizing latency under demanding, data center-centric tasks, the Intel Ethernet 800 Series Controller showed that it delivers the performance and flexibility needed for scalable Apache Kafka applications.

Network bandwidth has an obvious and immediate impact on solution performance and scalability. However, this does not mean that every link in a Confluent/Apache Kafka architecture needs to be 100 GbE. Figure 8 depicts the optimal arrangement of this document's reference architecture within a single rack. Links between leaf and spine switches, as well as broker nodes and the spine, should be 100 G, but links between support nodes and the leaf switch can back down to 10 G, as these are far less likely to be under sustained load. This is one way in which the Confluent/Apache Kafka architecture can be right-sized and made more cost-efficient.

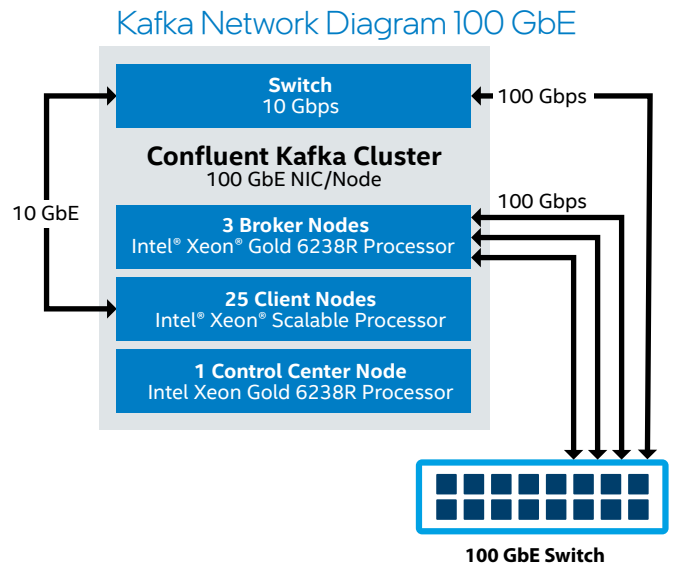


Figure 8. While the Confluent Apache Kafka architecture is flexible enough to be distributed across cloud-based resources, the lowest possible latencies will be found by deploying the reference architecture in a single rack.

Apache Kafka Cluster Sizing Guidelines: Scale Up, Then Scale Out

Although Intel selected a three-broker solution deployment scenario for modern enterprises, there is no one-size-fits-all Apache Kafka configuration. Deployments should be right-sized to fit the workloads and storage needs. The high data volume and mission-critical workloads need to scale up even when backed by powerful compute nodes, compared to other business functions that may need scale-out and higher data retention policies. Similarly, the networking infrastructure should match the throughput and storage needs of the solution.

Table 3. Relationship between data load and recommended network interface speed to support throughput

Number of Brokers	Network Utilization	Configured I/O	Threads Network	Average P99 Latency	Events/Second (1 kB message size)	Events/Day	Storage/Day (3x replication)	Recommended NIC ^a (to support throughput)
3	8 Gbps	15	8	4	1 million	86 billion	241 TB	10 GbE
3	19 Gbps	30	16	6	2.5 million	216 billion	603 TB	25 GbE
3	38 Gbps	45	24	9	5 million	432 billion	1,207 TB	40 GbE
3	46 Gbps	60	32	11	6 million	518 billion	1,448 TB	100 GbE
3	53 Gbps	70	40	12	7 million	605 billion	1,690 TB	100 GbE
3	57 Gbps	80	30	20	7.5 million	648 billion	1,810 TB	100 GbE
3	76 Gbps	80	30	82	10 million	864 billion	2,414 TB	100 GbE

^a NIC recommendation column does not account for consumer traffic.

Not every solution needs 100 GbE, and Table 3 shows the relationship between data load and recommended network interface speed. The highlighted row reflects Intel's recommended configuration for this platform, which offers the best overall performance for this number of brokers and data volume.

The data in this document shows that it is very unlikely that Apache Kafka users will consume the bandwidth of a 100 GbE infrastructure in a three-broker solution. Conversely, a 10 GbE infrastructure may prove insufficient for the same load levels. The recommended daily storage represents the sum of all bytes received by brokers. Since brokers work in parallel for both producing and consuming, the cluster requires three times the network bandwidth (including at the switch level) solely for workload data to accommodate 3x replication. For example, when incoming throughput is 53 Gbps, triple replication yields a need for 159 Gbps, which fits well within the 300 Gbps of bandwidth provided by three 100G broker connections. To understand the daily total recommended amount, use the 159 Gbps (which is 3x the original received 53 Gbps) and do the multiplication:

$$\frac{159 \text{ Gbps} \times (60 \text{ sec} \times 60 \text{ mins} \times 24 \text{ hours})}{8 \text{ (bits within a byte)}} = \sim 1,690 \text{ TB/day}$$

Intel recommends keeping Apache Kafka latency as low as possible within the bounds of affordability and feasibility. Given this configuration, a tail latency of 20 ms at 57 Gbps is advisable because the 80:30 thread settings deliver a slightly higher throughput (see Table 3). To reach this latency performance, researchers scaled up the components within each node through a process of optimizing component selection and software configuration.

However, in the pursuit of increasing platform performance, adjusting thread allocations and similar modifications fails to deliver additional throughput while preserving low latency. Greater performance now requires scaling out with additional brokers. To fully maximize the potential of 100 GbE in this scenario, one might want four or five brokers. Keeping three brokers allowed researchers to demonstrate a performance ceiling in this configuration.

Conclusion

This document provides a recommended reference architecture that optimally supports Confluent/Apache Kafka's features (including 3x replication) with high levels of performance and reliability. Researchers sought to achieve highly scalable data ingestion and throughput capacity while maintaining low latency.

These goals were achieved in a three-broker cluster architecture based on 2nd Gen Intel Xeon Gold processors, Intel Optane storage, Intel Ethernet Network Adapter 800 Series, and a finely tuned set of software configuration variables, most notably the ratio of I/O and network threads. This foundation yielded linear performance scaling with the number of producers and correlating increase in threads. With a ceiling observed in how far an Apache Kafka cluster could be "scaled up" within our configuration constraints, it then follows that a "scale out" of broker nodes would allow for higher levels of performance while maintaining desired low latency levels.

Apache Kafka is a central component to implement modern architectures and building event-driven applications, as a decoupler between high-volume datasets and downstream infrastructure. Using this reference architecture, enterprises can easily integrate Confluent/Apache Kafka into their application workflows and create resilient, highly effective solutions for data ingestion, movement, and storage.

Learn More

You may find the following references helpful:

- [2nd Generation Intel® Xeon® Scalable Processors](#)
- [Intel® Optane™ Persistent Memory](#)
- [Intel® Optane™ Solid State Drives](#)
- [Intel® SSD Data Center Family](#)
- [Intel® Ethernet Products](#)
- [Intel® Analytics](#)
- [Confluent Apache Kafka Platform](#)
- Paper: [High-Performance Data Analytics with Splunk on Intel® Hardware](#)
- Paper: [Building Building a Modern, Scalable Cyber Intelligence Platform with Apache Kafka](#)

Contributors

Dennis Kwong, Security Systems Engineer
Information Security, Intel

Jac Noel, Security Solution Architect
Information Security, Intel

Jason Stark, Technical Program Manager
Enterprise Strategy and Solutions, Intel

Merritte Stidston, Technical Solution Specialist
Data Platforms Group, Intel

Juan Fernandez, Technical Solution Specialist, Storage
Group, Intel

About Confluent



Born from the same team that created Apache Kafka, Confluent is a US-based software company that provides an Apache Kafka-based event streaming platform for accessing and analyzing distributed data in real time. Founded in 2014, the company is devoted to transitioning enterprises from legacy batch-processing methods into streaming while simultaneously helping those organizations adapt from on-premises limitations into multicloud flexibility. Deeply rooted in the open source community, Confluent is a privately held company headquartered in Mountain View, California.

Solution Provided By:



¹ **Test platform configuration #1:** Test by Intel as of 8/21/2020. 3-node, 2x Intel® Xeon® Gold 6258R Processors, 28 cores, HT On, Turbo ON; Total Memory 384 GB (12 slots/32 GB/2933 MHz); BIOS: SE5C620.86B.02.01.0010.010620200716 (ucode: 0x400002c); CentOS 8, 4.18.0-147.8.1.el8_1.x86_64, gcc (GCC) 8.3.1 20190507 (Red Hat 8.3.1-4); Network: 10 GB Network - Intel® NIC X722 (3 brokers); OS Storage: 1x 480 GB Intel® SSD DC S4510; Data Storage: 4x 4 TB Intel® SSD DC P4510; 4x 375 GB Intel® Optane™ SSD DC P4800x; 1-node Confluent Control Center 5.5.0; 20-node Apache Kafka Clients: Intel® NIC X722 10 GB; Confluent Broker Tunings: Open JDK 11, Broker (Java) Memory: 128 GB, Topic config: [1 Topic, 480 Partitions, 3x replications, min_insyn replicas=2, ack=1(default),0,-1], Open files:16384, Max mmap:225000.

Test platform configuration #2: Test by Intel as of 8/21/2020. 3-node, 2x Intel® Xeon® Gold 6258R Processors, 28 cores, HT On, Turbo ON; Total Memory 384 GB (12 slots/32 GB/2933 MHz); BIOS: SE5C620.86B.02.01.0010.010620200716 (ucode: 0x400002c); CentOS 8, 4.18.0-147.8.1.el8_1.x86_64, gcc (GCC) 8.3.1 20190507 (Red Hat 8.3.1-4); Network: 100 GB Network - Intel® NIC E810-CQDA2 (3 brokers), OS Storage: 1x 480 GB Intel SSD DC S4510; Data Storage: 4x 4 TB Intel SSD DC P4510, 4x 375 GB Intel Optane SSD DC P4800x; 1-node Confluent Control Center 5.5.0; 20-node Apache Kafka Clients: Intel® NIC X722 10 GB; Confluent Broker Tunings: Open JDK 11, Broker (Java) Memory: 128 GB, Topic config: [1 Topic, 480 Partitions, 3x replications, min_insyn replicas=2, ack=1(default),0,-1], Open files:16384, Max mmap:225000.

² For performance comparisons, see intel.com/content/www/us/en/architecture-and-technology/optane-technology/faster-access-to-more-data-article-brief.html.

For endurance information, see intel.com/content/www/us/en/architecture-and-technology/optane-technology/delivering-new-levels-of-endurance-article-brief.html.

³ businesswire.com/news/home/20190123005240/en/Confluent-Propels-Data-Architecture-Event-Streaming-Era

⁴ splunkbase.splunk.com/app/2935

⁵ confluent.io/wp-content/uploads/confluent-attunity-mainframe-data-modern-analytics.pdf

⁶ See endnote 1.

⁷ See endnote 2.

⁸ Data provided to the JDK community to drive ADQ changes for JDK integration: bugs.openjdk.java.net/browse/JDK-8244858

⁹ See endnote 2.

¹⁰ See endnote 1.

¹¹ See endnote 2.

¹² engineering.linkedin.com/Apache-Kafka/benchmarking-apache-Apache-Kafka-2-million-writes-second-three-cheap-machines

¹³ intel.com/content/www/us/en/architecture-and-technology/intel-optane-technology/reimagine-memory-storage-in-the-data-center.html

¹⁴ See endnote 1.

¹⁵ **Test platform configuration #1:** Test by Intel as of 8/21/2020. 3-node, 2x Intel® Xeon® Gold 6258R Processors, 28 cores, HT On, Turbo ON; Total Memory 384 GB (12 slots/32 GB/2933 MHz); BIOS: SE5C620.86B.02.01.0010.010620200716 (ucode: 0x400002c); CentOS 8, 4.18.0-147.8.1.el8_1.x86_64, gcc (GCC) 8.3.1 20190507 (Red Hat 8.3.1-4); Network:10 GB Network - Intel® NIC X722 (3 brokers); OS Storage: 1x 480 GB Intel® SSD DC S4510; Data Storage: 4x 4 TB Intel® SSD DC P4510; 4x 375 GB Intel® Optane™ SSD DC P4800x; 1-node Confluent Control Center 5.5.0; 20-node Apache Kafka Clients: Intel® NIC X722 10 GB; Confluent Broker Tunings: Open JDK 11, Broker (Java) Memory: 128 GB, Topic config: [1 Topic, 480 Partitions, 3x replications, min_insyn replicas=2, ack=1(default),0,-1], Open files:16384, Max mmap:225000.

¹⁶ See endnote 1.

¹⁷ **Test platform configuration #2:** Test by Intel as of 8/21/2020. 3-node, 2x Intel® Xeon® Gold 6258R Processors, 28 cores, HT On, Turbo ON; Total Memory 384 GB (12 slots/32 GB/2933 MHz); BIOS: SE5C620.86B.02.01.0010.010620200716 (ucode: 0x400002c); CentOS 8, 4.18.0-147.8.1.el8_1.x86_64, gcc (GCC) 8.3.1 20190507 (Red Hat 8.3.1-4); Network: 100 GB Network - Intel® NIC E810-CQDA2 (3 brokers), OS Storage: 1x 480 GB Intel SSD DC S4510; Data Storage: 4x 4 TB Intel SSD DC P4510, 4x 375 GB Intel Optane SSD DC P4800x; 1-node Confluent Control Center 5.5.0; 20-node Apache Kafka Clients: Intel® NIC X722 10 GB; Confluent Broker Tunings: Open JDK 11, Broker (Java) Memory: 128 GB, Topic config: [1 Topic, 480 Partitions, 3x replications, min_insyn replicas=2, ack=1(default),0,-1], Open files:16384, Max mmap:225000.

¹⁸ See endnote 1.

¹⁹ See endnote 1.

²⁰ See endnote 1.

²¹ Intel. "Intel® Optane™ Technology Delivers New Levels of Endurance." intel.com/content/dam/www/public/us/en/documents/reference-guides/delivering-new-levels-of-endurance.pdf

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 information go to intel.com/benchmarks.

Performance results are based on testing as of August 21, 2020, and may not reflect all publicly available security updates. See configuration disclosure for details. No product or component can be absolutely secure.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer to learn more at intel.com.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: [Learn About Intel® Processor Numbers](https://intel.com).

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

All rights reserved. Intel, the Intel logo, Xeon, and Optane are trademarks of Intel Corporation in the U.S. and/or other countries.

Other names and brands may be claimed as the property of others. © Intel Corporation 1120/JSTA/KC/PDF 341991-001US