White Paper

Unleashing the Power of Apache Pulsar, A High-Performance Messaging Platform, with Intel® Optane™ Persistent Memory

Persistence assurance is an important feature of message queues and simplifies business development. The storage architecture of Apache Pulsar is a great fit for the strengths of Intel® Optane™ persistent memory.

Together, they deliver a low-latency, high-TPS solution, further enhancing Pulsar's advantages in mission-critical scenarios.

— Co-Founder of StreamNative
Apache Pulsar PMC Member
Jia Zhai

Contents

Summary

Intel and StreamNative partnered to develop a new storage solution for mission-critical data.

Together, they developed a plug-in that uses the high-performance, low-latency Intel® Optane™ persistent memory as the storage medium that enables Apache Pulsar's messaging and event-streaming platform to deliver improved system latency, message throughput, and overall performance while ensuring data persistence.

Introduction

In the digital economy, the transformation and business innovation of enterprises hinge on how successfully they leverage the value of data, build data-centric business systems, manage business process data, and establish an application-centric data platform.

Message queues are a basic component of system applications, allowing asynchronous communication, application decoupling, and traffic peak clipping in a broad range of business scenarios. These can include banking, transaction processing, big data log analytics, AI-driven deep learning and personalized recommendations, as well as IoT applications like autonomous driving and the Industrial Internet. This diversity of scenarios requires dynamic, high-performing message queues.

In each of the use cases, the data security of message queues is an important component of the system, and this is where many conventional message queues fall short today. Conventional message queues such as RabbitMQ and ActiveMQ often fail to provide both the data security and system effectiveness that data applications require. In mission-critical scenarios, many enterprises are forced to make a trade-off between system performance and data persistence.

In order for companies to succeed in building businesses that are able to unlock and leverage the power of data, they need to choose the right tech stack - both software and hardware. For today's mission-critical message queues, many organizations choose Apache Pulsar.

Challenge

Apache Pulsar and SSD Storage

Pulsar is a cloud-native distributed messaging platform. Its design and abstraction allow messages to be consumed in high-performance streams or flexible queues. Besides ensuring the performance and throughput of big data message systems, Pulsar avoids the shortcomings of existing open-source message systems by providing enterprise-grade features: convenient O&M and expansion, flexible message models, a multi-language API, multi-tenancy, standby geo-redundancy, and strong data persistence and consistency.

Apache Pulsar ensures data persistence by writing data entering the message queue to the disk in real time, instead of the memory or cache, preventing data loss even if the system is down. This data persistence and a decoupled storage design enable Pulsar to meet the needs of many mission-critical applications. However, Pulsar is just one piece of the puzzle. For high-scale, business-critical applications such as stock trading, futures trading, financial billing, crypto-currency trading, critical event logging, and state data synchronization, organizations need performance at 1 ms latency. If you are using Pulsar with a typical SSD, you may have issues delivering this latency above a certain scale.

For example, while the proximity and communication lines of intra-region, active-active data centers may facilitate active-active failover for stateless background services, stateful background systems such as databases, file systems, and Redis require persistent, low-latency, and high-TPS data storage and synchronous replication. Organizations using Apache Pulsar with an SSD were not always able to hit the required latency and throughput.

Apache Pulsar Architecture

As shown in Figure 1, Pulsar features a two-layer architecture that separates compute from storage. On the top compute layer, brokers provide message management and services, with an automatic load balancing mechanism. On the underlying storage layer, Apache BookKeeper provides persistent and consistent storage for message streams. Brokers are responsible for Pulsar's entire business logic, while BookKeeper handles data storage. Pulsar owes its designation as a cloud-native message queue to this separation of storage and compute.

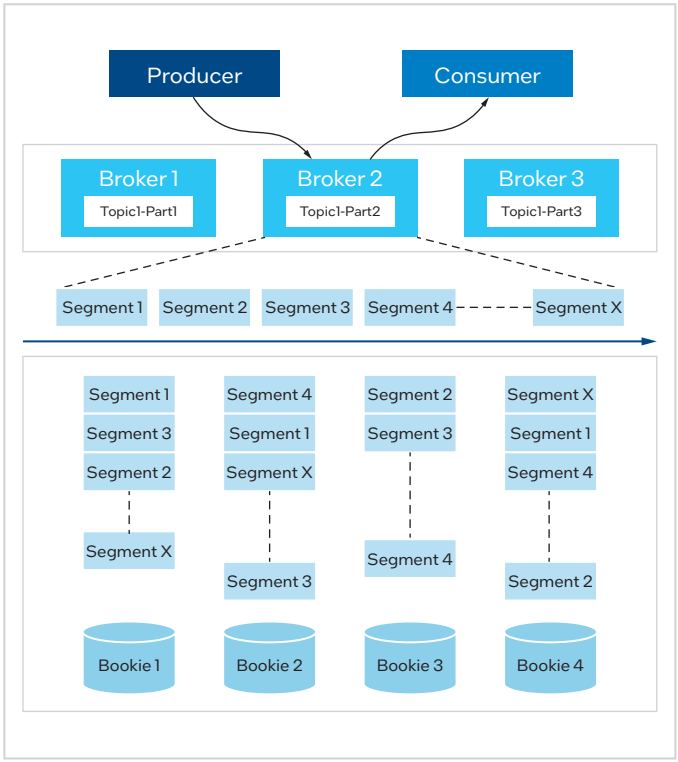

Figure 1: Pulsar architecture

Thanks to the cloud-native architecture, Pulsar excels in its transparent fault-tolerant mechanism and second-level scalability. Pulsar is widely used across industries and powers the core systems of companies such as Verizon Media, Splunk, Narvar, Intuit, Iterable, Tencent, Huawei, and Didi.

Persisting Data Without Impact to Latency and Throughput

Conventional message queues rely on file systems for data storage, which causes the storage performance to deteriorate when there are too many topics or when old data needs to be read. Pulsar's storage layer, by comparison, is better designed to support message-streaming scenarios.

To ensure data persistence, Pulsar stores data on the disk, instead of the cache or page cache. In Pulsar's storage nodes (Figure 2) data is appended to the journal and immediately persisted to the journal disk, and then the journal disk notifies the client of the write success. At the same time, the data is written to the write cache and then flushed to the ledger disk (an HDD or SSD) by the running background thread. Once written to the ledger disk, the data in the journal disk can be deleted.

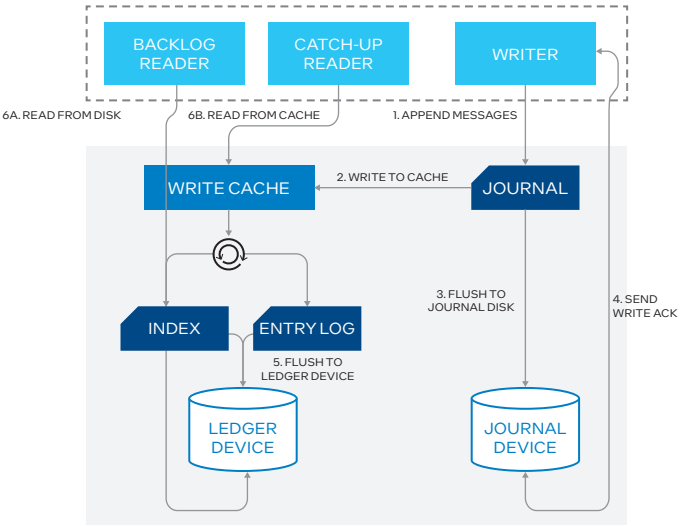

Figure 2: BookKeeper node architecture

According to the write path of messages, the journal is located on the critical data path. The write performance of the system depends on the write capability of the journal disk. In scenarios that require high data persistence, a high-performance, high-throughput, and low-latency journal disk is needed to improve the write performance of the system.

To improve journal write performance, Pulsar users often employ high-speed NVMe SSDs as journal disks for storage. However, such storage struggles to deliver high data persistence. In NAND flash storage, data is written in blocks to the disks, and existing data must be erased before new data can be written, resulting in a low write bandwidth, which is usually only 1-2 GB/s. Write cycles wear out SSDs, greatly shortening the service life of SSDs in scenarios where a large amount of data is written. Replacing a damaged SSD not only reduces the availability of the system but also incurs additional O&M costs. Moreover, the access latency of NAND flash-based SSDs is usually at the 100 µs level, resulting in an end-to-end access latency of up to several milliseconds or even tens or hundreds of milliseconds, affecting the latency and throughput of the system.

Solution

Intel® Optane™ Persistent Memory Overview

Intel® Optane™ persistent memory (PMem), powered by innovative Intel® Optane™ technology, is a revolutionary memory product built on 3D XPoint media (Figure 3). It features high speed, low latency, and persistent data protection. 3D XPoint is a memory technology that stacks memory grids in a 3D matrix to improve media density, performance, and data persistence. Unlike conventional NAND flash media, Optane media provides byte-addressable access to data using load/store commands. While the conventional media requires entire blocks to be erased before writing, the Optane media enables in-place updates, for substantially higher write performance. In addition, NAND flash memory has a limited number of write cycles during its lifespan compared to Optane memory, which has a much longer service life than NAND SSDs.

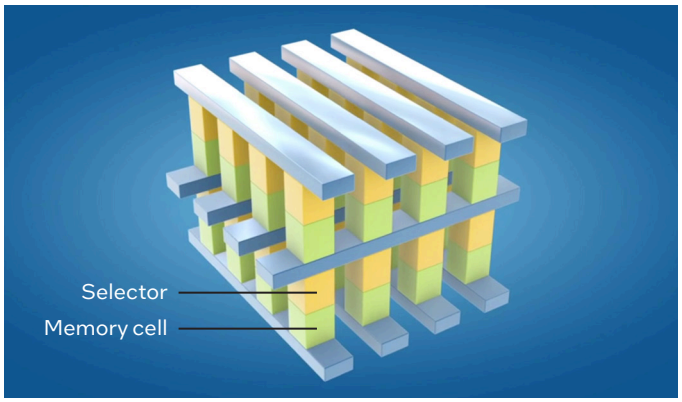

Figure 3: Intel® Optane™ memory and storage media combines memory and storage attributes by using a revolutionary 3D structure that enables high density, low latency, and persistence

As shown in the memory and storage pyramid (Figure 4), ultra-fast DRAM sits at the very top in the hot-data tier. In practice, a significant performance disparity exists between DRAM and SSD in the storage tier. The access latency of DRAM is generally tens of nanoseconds. In contrast, the latency of the fastest NVMe SSDs is up to tens of microseconds, 6 orders of magnitude slower than that of DRAM. This huge gap adversely affects system performance. The latency of Intel® Optane™ PMem is usually 100-340 ns, which is 2-3 orders of magnitude faster than that of SSDs. Compared with DRAM, Intel® Optane™ PMem delivers a similar performance while enabling data persistence and a larger capacity. Therefore, Intel® Optane™ PMem and Intel® Optane™ SSD have been introduced in the pyramid between DRAM and SSD to effectively bridge the performance gap between conventional storage devices.

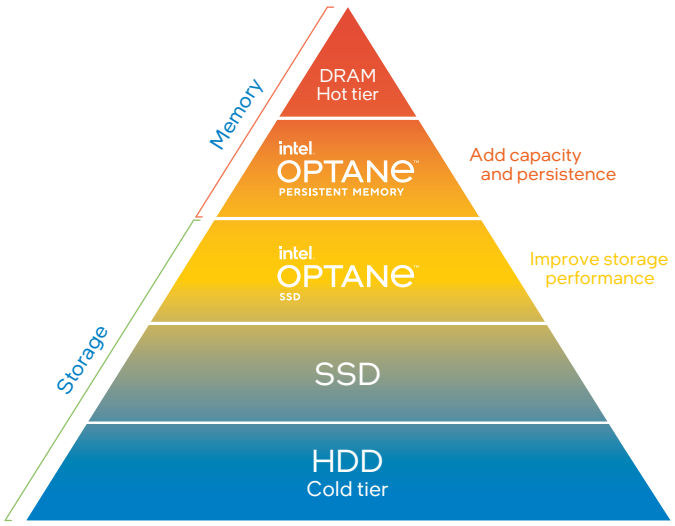

Figure 4: In the storage hierarchy, Intel® Optane™ PMem is just below DRAM

Intel® Optane™ PMem and Apache Pulsar

To overcome the impact of data persistence on latency and throughput, Apache Pulsar uses the Intel® Optane™ PMem as the journal disk. Figure 5 shows how Optane ensures data persistence while maintaining high performance.

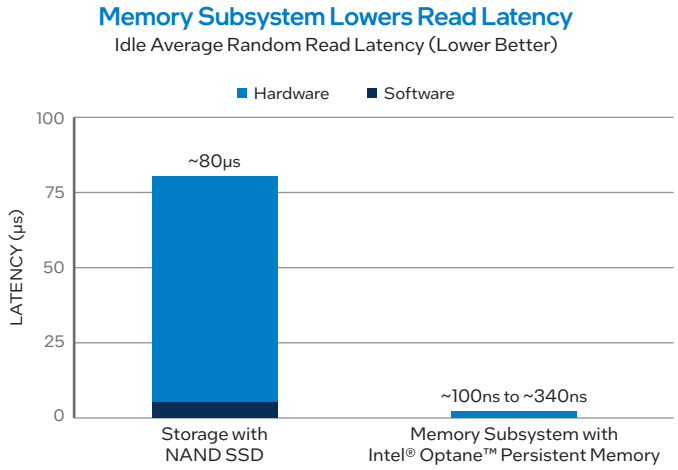

Figure 5: Comparison of read latency for NAND SSDs and Intel® Optane™ PMem

Intel Corporation partnered with StreamNative, Inc. StreamNative and Intel engineers developed a PMem-oriented log access plug-in built on Optane PMem. The plug-in takes full advantage of Optane PMem's high bandwidth and low latency to improve throughput and write latency in scenarios that require data persistence and consistency.

Enabling BookKeeper to Efficiently Access PMem

To figure out how to apply Intel® Optane™ PMem to Pulsar, the team first needed to understand how Optane supports Memory Mode and App Direct (AD) Mode.

In Memory Mode, the CPU memory controller treats PMem as volatile memory and DRAM as a cache for PMem. Memory Mode provides larger and cheaper memory, but it is volatile and cannot deal with scenarios that require data persistence.

AD Mode enables data persistence in a number of ways. For example, Optane PMem can serve as a conventional block-based storage device to achieve data persistence. However, the huge software overhead prevents taking full

advantage of the high performance of Optane PMem. In this scenario, FSDAX Mode or DEVDAX Mode should be used to ensure access efficiency.

In these 2 modes (Figure 6), software and applications written in the standard NVM PMem Programming Model can directly access Intel® Optane™ PMem. By mapping Optane PMem to the application memory space using mmap, applications no longer require system calls, interruptions, and context switches, so data is not replicated between the user space and the kernel space. Optane PMem excels in byte-addressable access to storage media using load/store commands at a cache line granularity much smaller than 4 KB, which reduces access latency significantly and prepares enterprises for high-speed business scenarios.

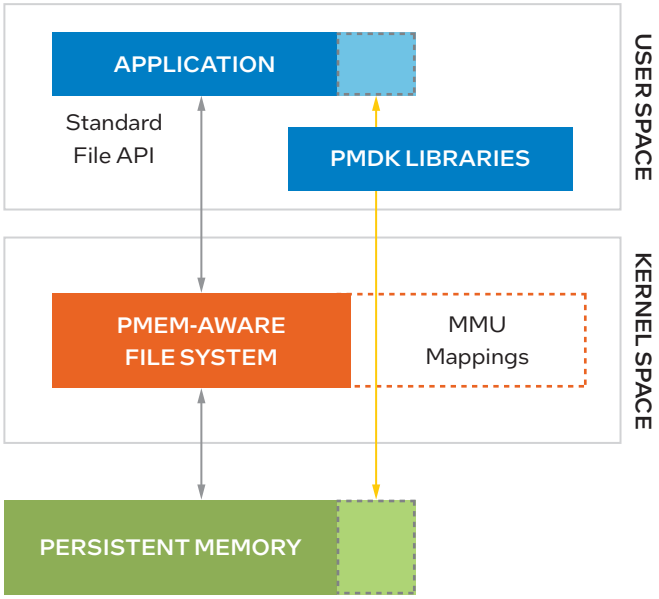

Figure 6: Accessing PMem via DAX Mode

To enable the BookKeeper journal module to write data to PMem efficiently, the team designed a PMem-based plug-in and implemented a PMem Channel Provider using PMem-AD in FSDAX Mode and the Persistent Memory Development Kit (PMDK).

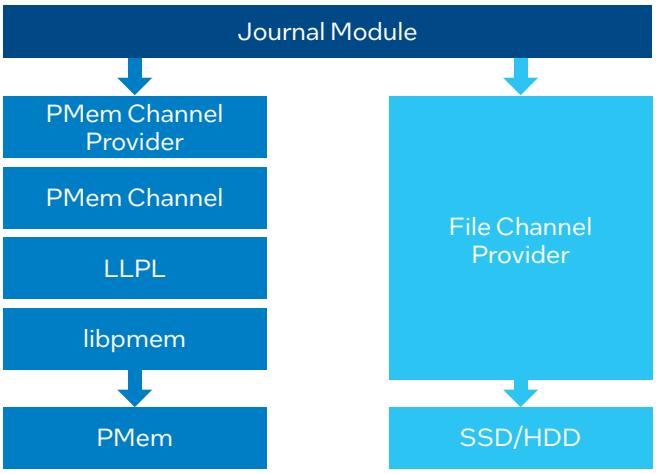

Figure 7: Internal structure of PMem Channel Provider

Figure 7 shows the internal structure of the PMem Channel Provider. At the bottom layer, libpmem, provided by PMDK, allows access to PMem. The upper layers consist of the Java API layer (LLPL), PMem Channel (an API class that implements the channel), and PMem Channel Provider. This non-intrusive approach provides support for PMem and enables a simple configuration for improved system performance.

According to the FSDAX Mode of SNIA's NVM Programming Model, the PMem plug-in directly writes data to the bottom medium through mmap system calls. Data reads and writes bypass the kernel, eliminating data replication between the kernel and the user as well as the overhead of context interactions between Kernel Mode and User Mode. As a result, Pulsar can take full advantage of low-latency, high-bandwidth PMem hardware.

Performance Comparison of Pulsar with PMem and Other Storage Media

With the newly developed PMem plug-in, Pulsar's performance was tested using Intel® NAND SSD as the journal disk and Intel® Optane™ PMem as the journal disk. The test configuration comprises 3 servers powered by the 3rd-generation Intel® Xeon® Scalable processors as bookies, separately combined with the 2nd-generation Intel® Optane™ PMem and Intel® P4510 NAND SSD as the journal disk. In addition, 3 other servers were running the brokers and OpenMessaging 0.0.1 as the test tool. The main configuration of BookKeeper, which provides storage solutions for Pulsar, is shown in Table 1. (For details, see the end of this document.)

	SSD as Journal Disk	1xPMem as Journal Disk	4xPMem as Journal Disk
CPU	Intel® Xeon® Platinum 8358 @2.6GHz *2		
Memory	64G DDR4 DRAM @3200MT *16		
Ledger Disk	NVMe SSD Intel P4510 2TB *6		
Journal Disk	Intel P4510 2TB *1	256GB Optane PMem 200 *1	256GB Optane PMem 200 *4

Table 1: BookKeeper (3-node) system configuration

First, the team tested Pulsar's lowest P99 latency using the 2nd-generation Intel® Optane™ PMem and Intel® P4510 NAND SSD.

As shown in Figure 8, using an SSD as the journal disk for storage resulted in a low P99 latency of 1.475 ms. With PMem as the journal disk, the lowest P99 latency was 0.655 ms (1 PMem module) and 0.666 ms (4 PMem modules), which were only 44% and 45% of the SSD, respectively. Based on this test, using PMem as the journal disk reduces the lowest P99 latency of the system to less than half of the SSD, achieving an ultra-low latency of less than 1 ms.

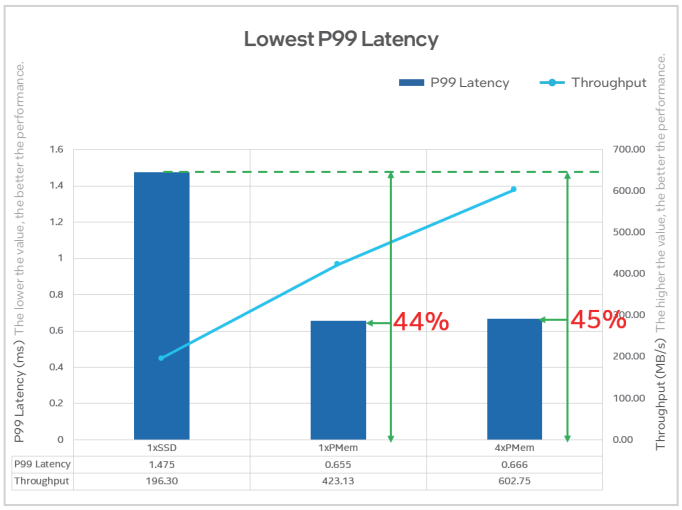

Figure 8

Second, the team tested Pulsar's throughput at low latency using the two media. As shown in Figure 9, using the SSD achieved a P99 latency of 1.475 ms. Using PMem resulted in a latency of 1.439 ms (1 PMem module) and a latency of 1.448 ms (4 PMem modules), which were close to the latency of the SSD. However, when applying the SSD as the journal disk, the system throughput was 196 MB/s. And even more significantly, the system throughput reached 4202 MB/s with 1 PMem module (21 times that of the SSD) and the system throughput was up to 12716 MB/s with 4 PMem modules (64 times that of the SSD).

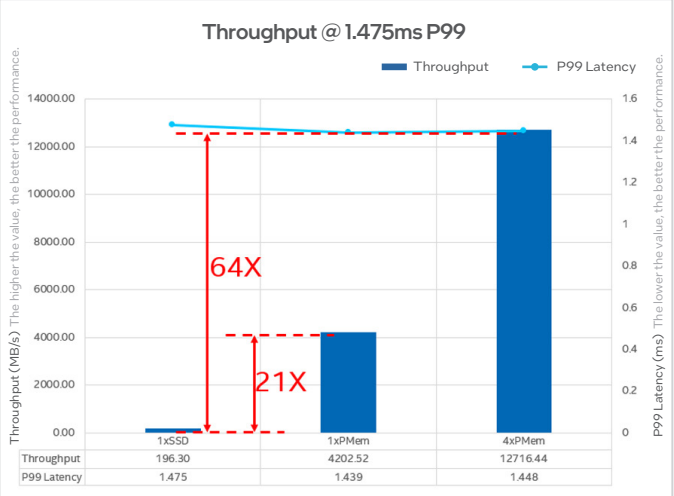

Figure 9

Lastly, the team evaluated Pulsar's maximum throughput using the PMem and the NAND SSD.

As shown in Figure 10, using the 1 PMem module for 1 KB and 4 KB messages, the maximum throughput was 1.26 times and 1.28 times that of the SSD, respectively. Under the 5 ms P99 SLA, the throughput was 1.5 times and 1.66 times that of the SSD, respectively.

Figure 10

As shown in Figure 11, using 4 PMem modules for 1 KB and 4 KB messages achieved a maximum throughput of 3.34 times and 3.56 times that of the SSD, respectively. Under the 5 ms P99 SLA, the throughput was 3.27 times and 4.56 times that of the SSD, respectively.

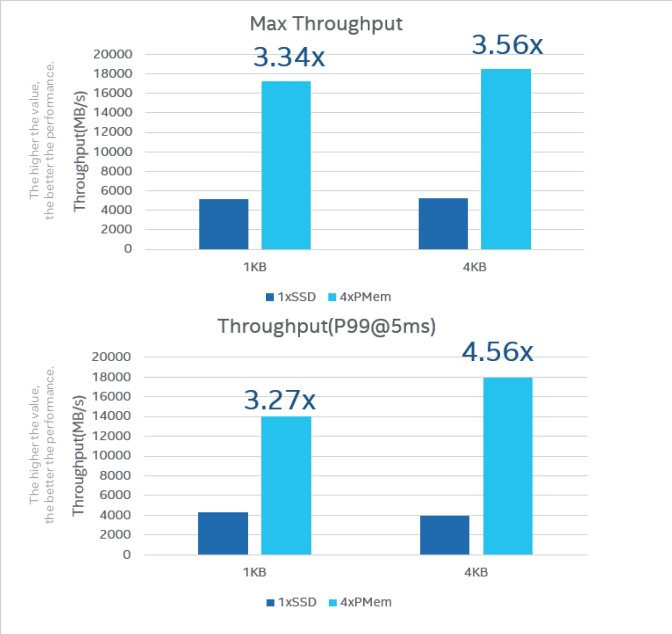

Figure 11

Conclusion

The test results show that using Intel® Optane™ PMem as Pulsar's journal disk for storage reduces system latency to less than half of the SSD, delivering a sub-millisecond latency, and improves the maximum throughput significantly.

To sum up the findings, Intel® Optane™ PMem is well suited to deal with scenarios that require high performance, such as finance and trading applications.

Looking Ahead

You can take full advantage of Intel® Optane™ PMem's low latency, high bandwidth, and data persistence in AD Mode. By using Intel® Optane™ PMem as Pulsar's journal disk for storage, your organization can easily cope with unexpected, high-throughput scenarios. Compared with SSDs, Intel® Optane™ PMem does not have the problem of a reduced service life. It improves system reliability and availability while contributing to a lower total cost of ownership (TCO).

Given the continuous development of Internet technology, the scenarios of message queues will become increasingly complex, demands on performance will get higher, and new trends will emerge in the field of message queues. With the support of Intel's technical background, StreamNative will work with Intel to enhance Pulsar's system performance and build a new-generation cloud-native messaging and streaming platform that advances the digital transformation of more enterprises and industries.

StreamNative

About StreamNative[1]

Founded by the original developers of Apache Pulsar and Apache BookKeeper, StreamNative builds a cloud-native event streaming platform that enables enterprises to easily access data as real-time event streams. Many StreamNative team members have been working on Apache BookKeeper-based event streaming systems, such as Apache Pulsar, for years and have seen the success of these systems at leading internet companies. In fact, some team members managed the largest BookKeeper production deployment in the world that processes trillions of events every day.

Our mission is to help businesses generate value from their enterprise data. Today StreamNative is focusing on growing the Apache Pulsar and BookKeeper communities and bringing our deep experience across diverse Pulsar use cases to companies around the world.

intel

[1] https://streamnative.io/

Test platform configuration:

3 BookKeeper nodes: Processors: Intel® Xeon® Platinum 8358 @2.6 GHz *2; Memory: 64 GB DDR4 DRAM @3200 MT *16; PMem: 256 GB Intel® Optane™ persistent memory 200 *8; NVMe SSD: Intel® SSD P4510 2 TB *6; (a P4510 2 TB SSD, 1 PMem module, and 4 PMem modules were separately used as the journal disk). Only CPU0 and the DRAM and PMem connected to it were used through NUMACTL. OS: CentOS 8.4.2105; kernel: 5.14.8; JDK: OpenJDK 1.8.0_322.

3 brokers: Processor: Intel® Xeon® Platinum 6240 @2.6 GHz *2; Memory: 16 GB DDR4 DRAM @2666 MT *12; OS: CentOS8.5.2211; kernel: 4.18.0-348-el8.x86_64; JDK: OpenJDK 1.8.0-292. CPU1 and the DRAM connected to it were used through NUMACTL.

Producer: Shared the physical servers with the brokers and used CPU0 and the DRAM connected to it through NUMACTL.

Pulsar: 2.9.2

BookKeeper: 4.14.4

No product or component can be absolutely secure. Costs and results may vary.

Intel does not control or audit third-party data. You should review third-party data and consult other sources to verify its accuracy. Performance varies by use, configuration, and other factors. For more information, see www.Intel.com/PerformanceIndex.

For workload/configuration information, see the appendix. Intel technologies may require enabled hardware, software, or service activation.

© Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel trademarks are trademarks of Intel Corporation or its subsidiaries in the United States and/or other countries.

*Other names and brands may be claimed as the property of others.