白皮书

突破系统瓶颈 激发 Pulsar 潜能
英特尔® 傲腾™ 持久内存打造高性能消息平台

持久性保证是消息队列的重要特点，它极大降低了业务开发的复杂性。Apache Pulsar 的存储架构和英特尔® 傲腾™ 持久内存的优势十分契合，两者的结合带来的极致时延和 TPS，使 Pulsar 在关键业务场景中的优势得到了更进一步的提升。

— StreamNative 联合创始人
Apache Pulsar PMC 成员
翟佳

目录

引言

在数字经济时代，企业如何深入挖掘数据价值，搭建以数据为核心的业务体系，将业务流程以数据的形式加以沉淀，构建以应用为中心软硬结合的数据平台，成为未来企业数字化转型和业务创新的关键。

消息队列作为系统应用的基础构件，凭借异步通信、应用解耦、流量削峰的能力，广泛地应用于各种业务场景，例如银行系统的事务处理，大数据的日志分析，人工智能中的深度学习、个性化推荐，物联网领域的自动驾驶、工业互联网等等。应用场景的多样性也对消息队列提出更高的需求。

消息队列中的数据安全性不言而喻。目前主要的消息队列中，缺乏对数据安全性和系统有效性的完美支持，特别是在一些关键的业务场景中，传统的消息队列难以在确保数据持久化的同时仍然保证系统的高性能，企业往往需要在二者之间做出抉择。

Pulsar 云原生分布式消息队列，提供了一种数据持久性与系统性能矛盾的解决方案。在 Pulsar 中可以优先保证数据的持久性，进入消息队列的数据实时写入到磁盘而不是内存或者缓存中，即使出现系统宕机，数据也不会丢失；而在确保数据持久性的前提下，Pulsar 依靠自身独特的存储设计，能满足很多场景所需的性能需求。但是随着系统规模的不断扩展，应用场景持续增加，在一些关键业务场景中，如股票、期货、金融计费、数字货币交易、关键事件日志、状态数据同步等，对 Pulsar 的时延、吞吐和一致性提出了更高的要求，在 1ms 的时延要求下 Pulsar 无法在满足数据持久性的同时，提供应用所需要的高性能。

例如在同城双活的实际应用中，机房一般距离较近，通信线路质量较好，对无状态的后台服务双活切换较易实现，但诸如数据库、文件系统、Redis 等有状态后台系统在数据存储、同步复制时对数据持久性、时延、TPS 的要求很高，维护难度很大。Pulsar IO 和跨地域复制功能被使用在此场景中时需要有更低的延时、更高的吞吐来满足系统的要求。为破解这一难题，StreamNative 与英特尔合作，利用高性能、低时延的英特尔® 傲腾™ 持久内存作为存储介质，开发了全新的存储插件，在确保数据持久化的前提下，降低了系统时延，提升了消息吞吐率，改善了 Pulsar 的整体性能，能很好地满足关键业务场景中的需求。

Pulsar：云原生的统一消息流平台

Pulsar 是一个云原生分布式消息系统。Pulsar 通过特别的设计和抽象，统一地支持 Stream 和 Queue 两种消息消费模式，保持了 Stream 模式的高性能和 Queue 模式的灵活性。Pulsar 在保证大数据消息系统的性能和吞吐量的同时，提供了更多企业级的 Feature，包括方便的运维和扩展，灵活的消息模型，多语言 API，多租户，异地多备，和强持久性、一致性等等，解决了现有开源消息系统的一些不足。

Pulsar 的迅猛发展，得益于独特的性能优势。

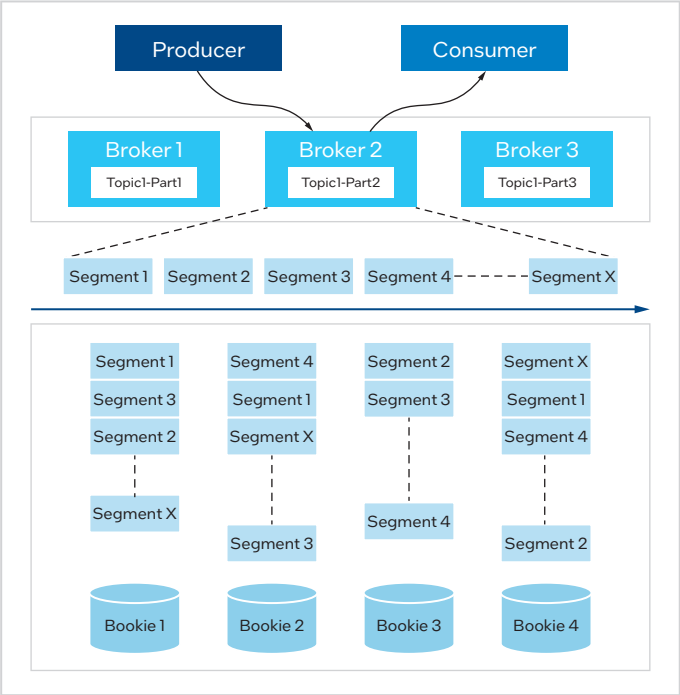

图1 Pulsar 架构图

Pulsar 是一个存储和计算分离的架构，如图 1 所示 Pulsar 分为两层，服务层 Broker 提供消息的管理和服务，同时承担负载均衡的作用，不存储消息，称为计算层；底层则由专为消息流存储而设计的 Apache BookKeeper 提供数据持久性和一致性的保证，称为存储层。Broker 负责整个 Pulsar 的业务逻辑，而 BookKeeper 负责数据的存储。存算分离的架构特点是 Pulsar 被称为云原生消息队列的主要原因。

云原生的架构，使 Pulsar 具备了无感知容错机制和秒级扩展扩容能力，目前 Pulsar 已经被广泛使用在各个领域和核心系统中，比如 Verizon Media、Narvar、Intuit、Iterable、腾讯、华为、滴滴、EMQ 等公司。

挑战：数据持久化影响 Pulsar 时延和吞吐率

传统消息队列直接依赖文件系统做数据的存储，在 Topic 数量过多、读老旧数据等场景下会明显影响性能。Pulsar 的存储层能够更加适应消息流场景的存储需求。Pulsar 默认提供更高的数据持久性保证，数据需要被持久化存储到磁盘中，而不是缓存或者 pagecache 中。在 Pulsar 存储节点里（图 2），当一个数据写入后，首先会被追加到内存日志中 (Journal)，接着数据被立即持久化保存到日志盘中 (Journal Disks)，然后通知客户端写入成功。这些数据同时也被写入到 Write Cache 中，Write cache 中的数据由后台运行线程，批量写入到作为 Ledger 盘的 HDD 或 SSD 中，一旦写入到 Ledger 盘，日志盘中的数据就可以被删除了。

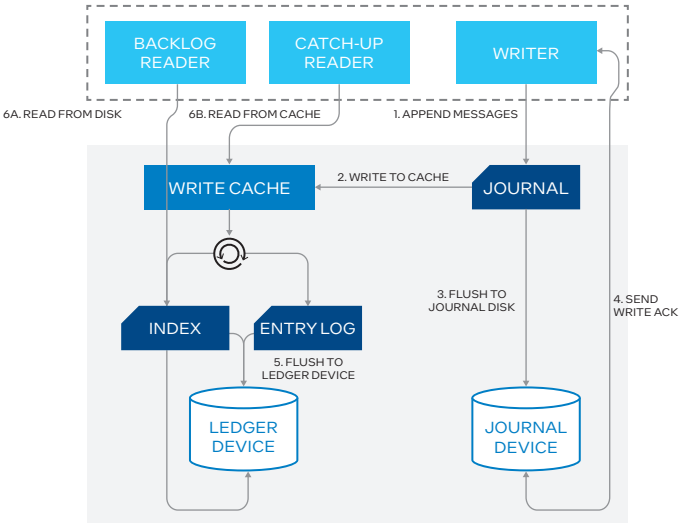

图2 BookKeeper 节点架构

从消息写入的路径中我们可以看出，日志位于数据的关键路径上，系统写性能取决于日志盘 (Journal) 写入的能力，特别是对数据持久性要求很高的场合，如果要提升系统的写性能，就要提高日志盘的性能，包括系统的吞吐率和访问时延。

为了提高写日志的性能，Pulsar 用户通常使用高速的 NVMe SSD 作为日志盘存储介质。在对数据持久化要求很高的场合，NVMe SSD 存储作为日志盘越来越力不从心。首先，NAND Flash 介质需要以块为单位写入磁盘，在写入前都需要有擦写的操作，导致其写带宽较低，通常只有 1-2GB/s；同时数据写入也会导致 SSD 的磨损，特别是对于大量数据写入的场景，SSD 使用寿命大幅缩短；而更换损坏的 SSD 不但降低了系统的可用性，也为运维带来额外的成本支出。另一方面，基于 NAND Flash 的 SSD 的访问时延通常在 100us 的量级，导致端到端的访问时延高达数毫秒甚至数十、数百毫秒，无法应对一些需要低时延的场景，吞吐率也大幅下降。

解决方案：
英特尔® 傲腾™ 持久内存加速 Pulsar 数据写入

英特尔® 傲腾™ 持久内存 (PMem)

英特尔® 傲腾™ 持久内存（以下简称持久内存），采用创新的英特尔® 傲腾™ 技术，是一种颠覆性的内存产品，它基于革命性的 3D XPoint 介质（图 3），具有高速、低时延、持久数据保护等优势。3D XPoint 能够在三维矩阵中堆叠内存网格，提高介质密度、增强性能，提供持久性。不同于传统的 NAND Flash 介质，傲腾介质提供基于字节的寻址能力，可以直接使用 load/store 指令访问数据，而传统的 NAND Flash 由于需要对整块进行擦除，因此写效率很低，而傲腾介质实现了就地更新，大幅提升了写性能。此外，NAND Flash 只有有限的擦写次数，因此其寿命也为人诟病，而傲腾介质具有超长的寿命，大大超越 NAND SSD。

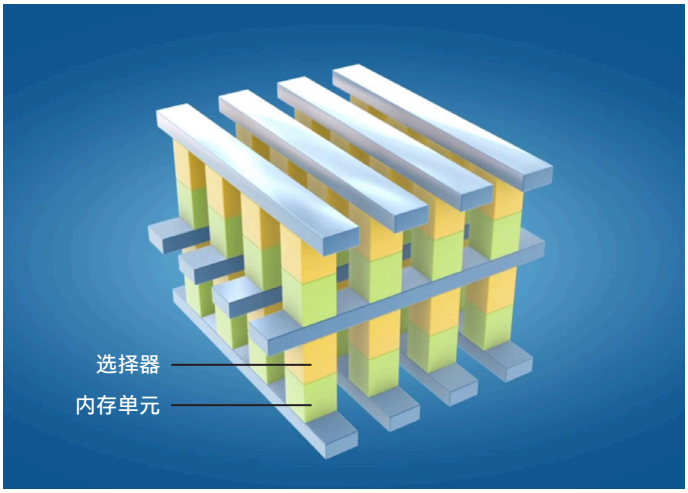

图3 通过使用可提供高密度、低时延和持久性的革命性 3D 结构，英特尔® 傲腾™ 内存和存储媒体兼具了内存和存储的属性

在图 4 的内存存储金字塔模型中，超高速的 DRAM 位于最顶端用于存储热数据。在实际应用中，DRAM 与作为存储的 SSD 层之间，存在严重的性能不匹配的情况，通常 DRAM 的访问时延是几十纳秒，而速度最快的 NVMe SSD 时延也高达数十微秒，要慢六个数量级，这种巨大差异对系统性能有很大的负面影响。英特尔® 傲腾™ 持久内存的时延一般为 100-340ns，比 SSD 要快两到三个数量级。而与 DRAM 相比，它的性能相当，而且能实现数据持久化，并有更大的容量。因此，在存储金字塔模型中，DRAM 和 SSD 之间引入的英特尔® 傲腾™ 持久内存和英特尔® 傲腾™ SSD，能有效地消除传统存储设备之间的性能鸿沟。

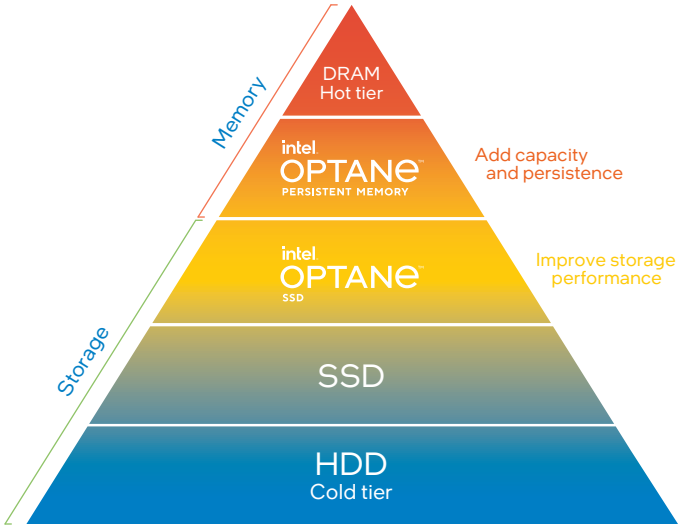

图4 在存储层级中，英特尔® 傲腾™ 持久内存位于 DRAM 下方

为了应对 Pulsar 前述的挑战，我们提出了使用英特尔® 傲腾™ 持久内存作为 Pulsar 的日志盘的方案，能大幅改善 NAND SSD 时延的问题，确保数据持久性的同时仍然保持高性能。

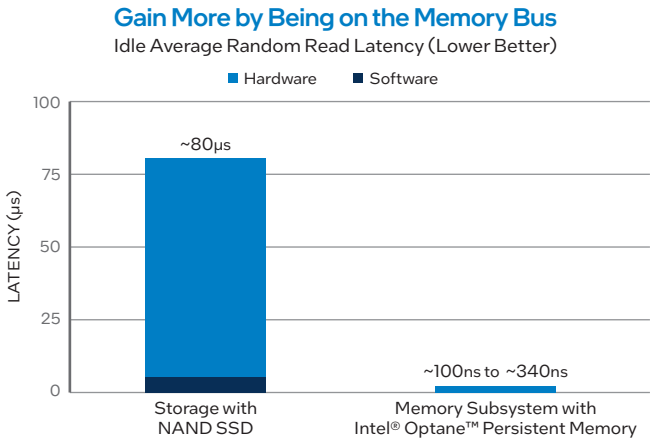

图5 NAND SSD 与英特尔® 傲腾™ 持久内存读时延对比

为了实现此目标，英特尔公司与 StreamNative 公司展开合作。StreamNative 公司由 Apache Pulsar 创始团队成员于 2019 年创立，对 Pulsar 有深刻的理解，致力于围绕 Apache Pulsar 打造下一代云原生批流融合数据平台，助力企业挖掘实时数据价值。StreamNative 与英特尔技术人员基于傲腾持久内存，开发了面向 PMem 的日志访问插件，能充分发挥傲腾持久内存高带宽、低时延的特点，有效提升 Pulsar 的吞吐率，而且还能降低写消息时延。

实现 BookKeeper 对 PMem 的高效访问

在具体的应用场景中，如何将英特尔® 傲腾™ 持久内存应用于 Pulsar 中呢？

英特尔® 傲腾™ 持久内存有两种主要工作模式 — 内存模式和 APP Direct 模式（以下简称 “AD 模式”）。

在内存模式下，CPU 内存控制器会把持久内存视作易失性的系统内存，而将 DRAM 内存用作持久内存的高速缓存。内存模式通常是用于提供更大更便宜的内存，但是它是易失性的，无法应对需要数据持久化的场景。

要实现数据持久化，需要使用 APP Direct 模式（简称 “AD 模式”）。AD 模式也有多种用法，其中某些用法可以将傲腾持久内存用做传统的基于块的存储设备，实现数据的持久化，但是因为其庞大的软件开销，无法充分发挥傲腾持久内存的高性能。要实现更高效的访问，需要使用 FSDAX 模式或者 DEVDAX 模式。

当使用这两种模式时（见图 6），按照行业标准 NVM 持久化内存编程模式编写的软件和应用，能直接访问英特尔® 傲腾™ 持久内存。通过 mmap 将傲腾持久内存映射到应用内存空间，应用不再需要系统调用、中断和上下文切换，消除了用户空间与内核空间的数据拷贝，它提供字节寻址能力，利用 load/Store 指令，实现以比 4KB 小得多的 cacheline 的粒度来访问存储介质，能大幅降低访问时延，满足高速业务场景的需求。

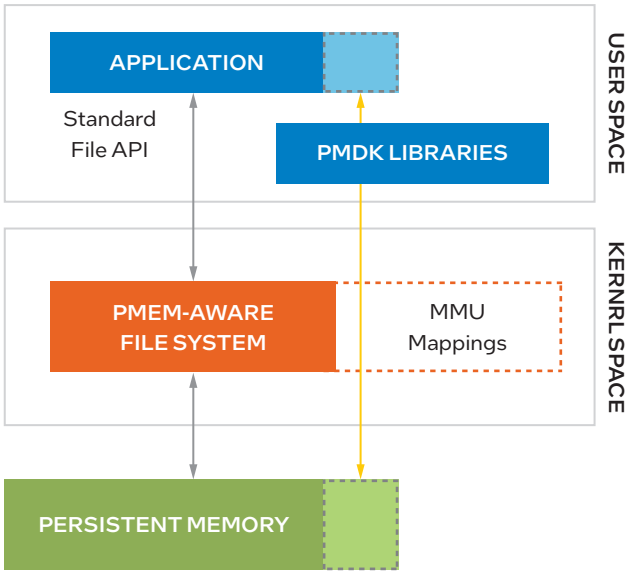

图6 通过 DAX 模式访问持久内存

要使用这种模式，需要对应用进行相应的修改。

在保证系统兼容性的前提下，我们使用持久内存 AD 的 FSDAX 模式，利用 PMDK（持久性存储器开发套件），设计了基于 PMem 的插件，实现了 PMem Channel Provider，使得日志模块可以将数据高效地写入 PMem。

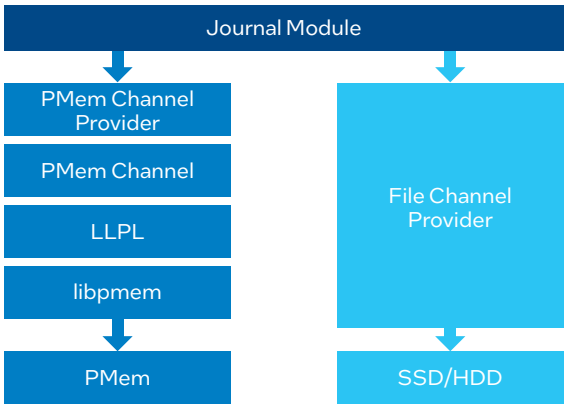

图7 PMem Channel Provider 内部结构

图 7 为 PMem Channel Provider 的内部结构，在底层我们使用了 PMDK 提供的 libpmem 来访问 PMem，在其上层分别是 Java 的接口层 LLPL，PMem Channel 类（实现了 channel 的接口），和 PMem Channel Provider。通过这种非侵入的方式，实现对 PMem 的支持，用户只需要进行简单的配置就可以应用 PMem 来提高系统的性能。

根据 SNIA NVM 编程模型中的 FSDAX 模式，PMem 插件通过内存映射 (mmap) 系统调用直接将数据写入到底层介质，数据的读写不再经过内核，避免了内核与用户间的数据拷贝，同时消除了内核态与用户态上下文交互的开销。因而能充分发挥 PMem 硬件的低时延、高带宽的特性，提高 Pulsar 响应速度与吞吐率。

英特尔® 傲腾™ 持久内存提高 Pulsar 吞吐

基于新开发的 PMem 插件，我们对其性能进行了测试和比较。测试团队选用英特尔® NAND SSD 和英特尔® 傲腾™ 持久内存分别作为 Pulsar 日志盘存储介质，对 Pulsar 和 BookKeeper 的性能进行测试和比较。在实际的配置中，我们使用了 3 台基于第三代英特尔® 至强® 可扩展处理器的服务器，作为 BookKeeper 存储服务器，分别搭配第二代英特尔® 傲腾™ 持久内存和英特尔® P4510 NAND SSD 作为日志盘。另外还有三台服务器运行 Broker 和测试工具。测试过程中使用了 OpenMessaging 0.0.1 作为测试工具。作为 Pulsar 存储服务器的 BookKeeper 主要配置信息如表 1（详细配置见文末）。

	SSD as Journal Disk	1xPMem as Journal Disk	4xPMem as Journal Disk
CPU	Intel® Xeon® Platinum 8358 @2.6GHz *2		
内存	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

表1 BookKeeper（三台）系统配置

首先，我们测试了他们所能达到的最低时延，以及在低时延下的吞吐率表现。

我们测试了分别使用两种介质时的 P99 的最低时延数据。在图 8 中使用 SSD 作为日志盘存储介质，P99 时延最低为 1.475ms，使用 PMem，P99 时延最低仅为 0.655ms（单条 PMem） 和 0.666ms（4 条 PMem），仅为 SSD 的 44% 和 45%。可以看到，使用 PMem 作为日志存储介质，能将系统最低时延 P99 降低到小于 SSD 一半，实现小于 1 毫秒的超低时延。

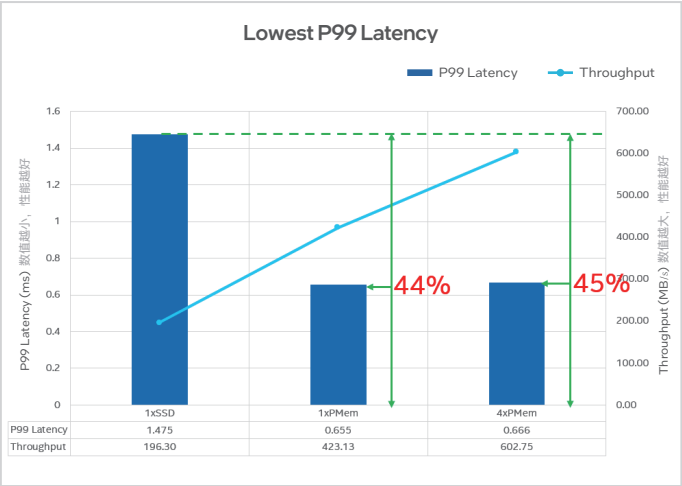

图8

我们也比较了二者在低时延情况下的吞吐率情况如图 9，使用 SSD，P99 时延最低为 1.475ms，使用 PMem，我们选取与 SSD 时延最为接近的 1.439ms（单条 PMem） 和 1.448ms（4 条 PMem）。在此情况下，使用 SSD 作为日志盘，系统吞吐率为 196MB/s，而使用单条 PMem 时，系统吞吐率达到 4202MB/s，是 SSD 的 21 倍，如果使用四条 PMem，系统吞吐率更是高达 12716MB/s，是 SSD 的 64 倍。

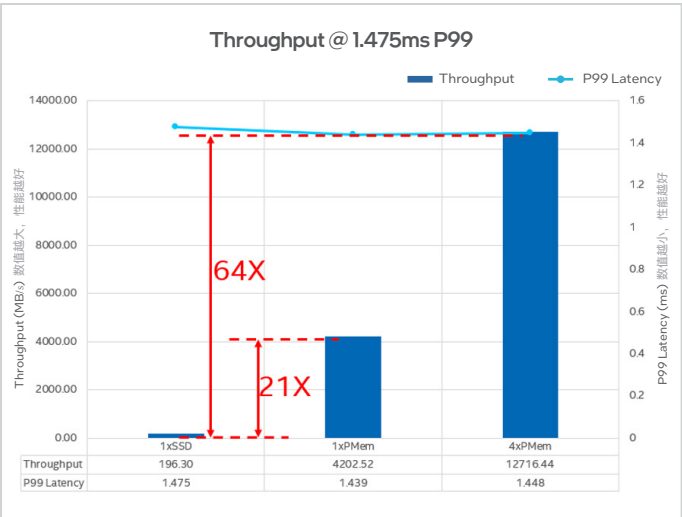

图9

我们也测试了使用 PMem 和 NAND SSD 所能达到的最大吞吐率。

图 10 中，单条 PMem，对于 1KB 和 4KB 的消息，其最大吞吐率分别为 SSD 的 1.26 倍和 1.28 倍，在 5ms P99 SLA 前提下，其吞吐率达到 SSD 的 1.5 倍和 1.66 倍。

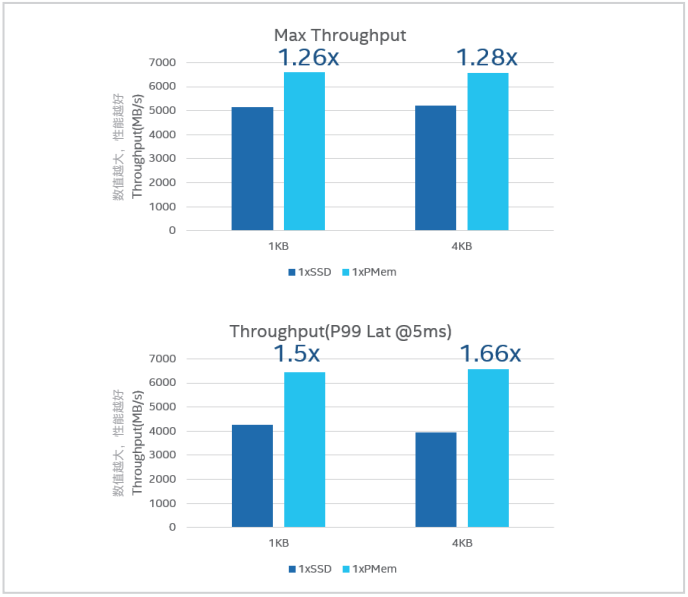

图10

如果使用 4 条 PMem，如图 11，对于 1KB 和 4KB 的消息，其最大吞吐率分别为 SSD 的 3.34 倍和 3.56 倍，在 5ms P99 SLA 前提下，其吞吐率达到 SSD 的 3.27 倍和 4.56 倍。

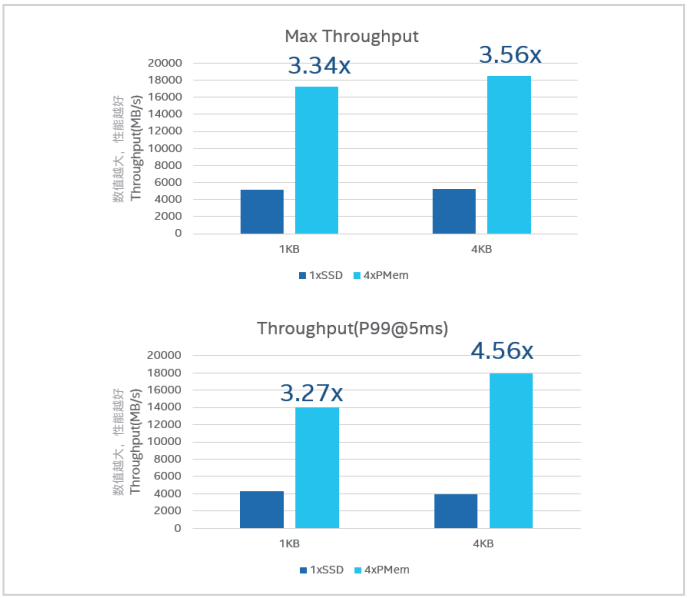

图11

从上述测试结果可以看出，使用英特尔® 傲腾™ 持久内存作为 Pulsar 日志盘的存储介质，不但能将系统最低时延降低到不足 SSD 的一半，实现亚毫秒级的时延，而且其最大吞吐率也有较大提升。能很好地应对金融，交易等这类对于性能要求严苛的应用场景。

展望

英特尔® 傲腾™ 持久内存在 APP Direct 模式下，能够充分发挥低时延、高带宽、持久化的优势，作为 Pulsar 日志盘的存储介质，可以从容应对突发性、高吞吐量的应用场景；相较于 SSD 没有使用寿命太短的烦恼，提高系统可靠性、可用性的同时，有效的控制总体拥有成本 (TCO)。

在未来，随着互联网技术的不断发展，消息队列的应用场景会日益复杂，性能需求不断提升，消息队列也必将呈现新的发展趋势。StreamNative 将与英特尔携手合作，借助英特尔深厚的技术底蕴，不断提升 Pulsar 的系统性能，打造新一代云原生批流融合的数据平台，为更多的企业与行业数字化转型提供快速、高效的驱动力。

StreamNative

关于 StreamNative[1]

StreamNative 是一家开源基础软件公司，由 Apache 软件基金会顶级项目 Apache Pulsar 创始团队组建而成，围绕 Pulsar 打造下一代云原生批流融合数据平台。StreamNative 作为 Apache Pulsar 商业化公司，包含 Pulsar 社区的主要维护者和贡献者，专注于开源生态和社区构建，致力于前沿技术领域的创新，创始团队成员曾就职于 Yahoo、Twitter、Splunk、EMC 等知名大公司。

intel

[1] https://streamnative.cn/

测试平台配置：

BookKeeper 三台： 处理器：英特尔® 至强® Platinum 8358 @2.6GHz *2；内存：64G DDR4 DRAM @3200MT *16；PMem：256GB 英特尔® 傲腾™ 持久内存 200 *8；NVMeSSD：英特尔® 固态盘 P4510 2TB *6；（分别使用一块 P4510 2TB 和 1 条和 4 条 PMem 作为 journal disk）。通过 NUMACTL 仅使用 CPU0 及与之连接的 DRAM 和 PMem。OS：CentOS 8.4.2105；kernel：5.14.8；JDK：OpenJDK 1.8.0_322。

Broker: 三台： 处理器：英特尔® 至强® Platinum 6240 @2.6GHz *2；内存：16G DDR4 DRAM @2666MT *12；OS：CentOS8.5.2211；kernel：4.18.0-348-el8.x86_64；JDK：OpenJDK 1.8.0-292。通过 NUMACTL 使用 CPU1 及与之连接的 DRAM。

Producer: 与 broker 共用物理服务器，通过 NUMACTL 使用 CPU0 及与之连接的 DRAM。

Pulsar: 2.9.2

BookKeeper: 4.14.4