白皮书

联通沃云基于 Optane™ SSD 的性能调优研究

作者：中国联通 张振尧 吴兴义 孙方臣 刘中、英特尔 方亮 朱韦韦 高峰 胡明月

目录

你的云应用场景中有密集的读请求需求吗？亦或你的云系统有磁盘读写瓶颈吗？希望本篇白皮书介绍的调优方式也能有助于您解决类似问题。

项目简介

联通沃云海量数据检索平台，每日要从各省采集业务数据，单省日均采集记录达到 500 万条，最高峰值达到千万条记录；而后入库对数据做解析和分析后，提供对外的开放大数据服务，受信的第三方通过开放的接口可随时调用大数据平台中的数据。平台采用 ES (ElasticSearch)作为分布式搜索引擎，由于云主机 IO 读写性能限制，导致大规模查询和按条件查询响应较慢。ES 大部分场景都是读多写少，因此在对系统做分表和 ES 检索优化的同时，探索提升云主机 IO 读写性能，尤其是减少读请求的延迟是一个非常重要的优化方向。

沃云海量数据检索平台是基于成熟的 OpenStack 构建的大型云服务平台。OpenStack 问世已经将近十个年头，目前 OpenStack 已经广泛应用于各行各业，包括电信，金融，政务，交通运输行业以及企业内部 IT 和业务支持。近年来业界对于 OpenStack 云的整体性能调优也在持续进行中。比如将各种 GPU 和 FPGA 加速设备直接通过 SRIOV 方式暴露给虚拟机，让虚拟机可以直接使用硬件设备；在存储节点和计算节点之间通过更加快速的协议相连，比如 NVMEoF；在远程存储端增加各种缓存以提升存储性能等等。这些调优极大地提升了 OpenStack 云的整体性能，增加了云系统的可用性。但目前业界主流的后端存储和计算节点之间往往还是采用 ISCSI 协议或者 RBD 相连。这些连接方式兼容性和稳定性非常好，但是有个致命的弱点就是延迟太大，典型的延迟都是毫秒级。毫秒和微秒对于人类来说没多少区别，但是对于一个计算机系统来说，相差一千倍。

那怎么减少延迟提升性能呢？目前业界已经出现最小延迟达到 10 微秒的固态硬盘，比如 Intel®的 Optane™ SSD，它的标称最小延迟可以达到 10 微秒。对于一个虚拟机来说，

远程盘的延时难以轻易改善，但要是在虚拟机所在的本地计算节点挂载一块高速固态硬盘，作为远程盘的缓存，那这样性能就可以大大得到提升了。沿着这个思路中国联通联合英特尔尝试对联通沃云系统进行调优。为了不影响联通沃云海量数据检索平台的日常运行，我们的调优会基于一个轻量级的实验室环境进行。

优化前性能测试

首先，我们先对未进行调优的系统进行性能测试和采集，作为基准数据。

测试方法和环境

测试中，我们模拟线上沃云的云环境，在实验室服务器上搭建 OpenStack 平台，然后创建一个虚拟机。服务器、虚拟机以及远程的 Ceph 盘的配置如右表所示。在工具的选取上，我们采用存储领域广泛使用的 FIO 性能测试工具，然后在虚拟机里对远程挂载过来的磁盘进行随机读，随机写，随机读写（7/3）等典型负载指标进行测试。

特别需要指出的是基准测试是直接对远程盘进行，这里没有额外的优化，没有使能 Optane™ SSD 当缓存。

测试延迟时，FIO 块大小设为 4k，iodepth 设为 1；测试 IOPS 时，块大小设为 4k，iodepth 设为 16。

测试环境

物理机器	
- **CPU**	2 Socket; Intel(R) Xeon® Platinum 8260 CPU @ 2.40GHz
- **内存**	384G
- **网卡**	Intel Corporation Ethernet Connection X722 for 10GBASE-T (rev 03)
- **SSD**	Intel® Optane™ SSD P4800x 750G
- **OS**	Ubuntu 19.10
- **Kernel**	5.3.0-24-generic
- **OpenCAS**	19.09.00.00001212
虚拟机	
- **CPU**	16 VCPU
- **内存**	20G
- **OS**	Ubuntu 19.10
- **Kernel**	5.3.0-24-generic
- **FIO**	3.12
远程盘	
- **类型**	Ceph
- **大小**	100G
- **Ceph 配置**	1 MON 3 OSD
- **Ceph 版本**	13.2.8 mimic

性能测试及数据收集

采集的性能数据包括平均延迟，延迟的波动率表征数值 P99，P99.9，P99.99 等数据。同时对每秒读写请求数 IOPS 进行了采集。考虑到顺序读和顺序写会大量合并读写请求，并且这也不是联通沃云的实际应用场景，所以我们没有专门采集顺序读和顺序写的性能数据。

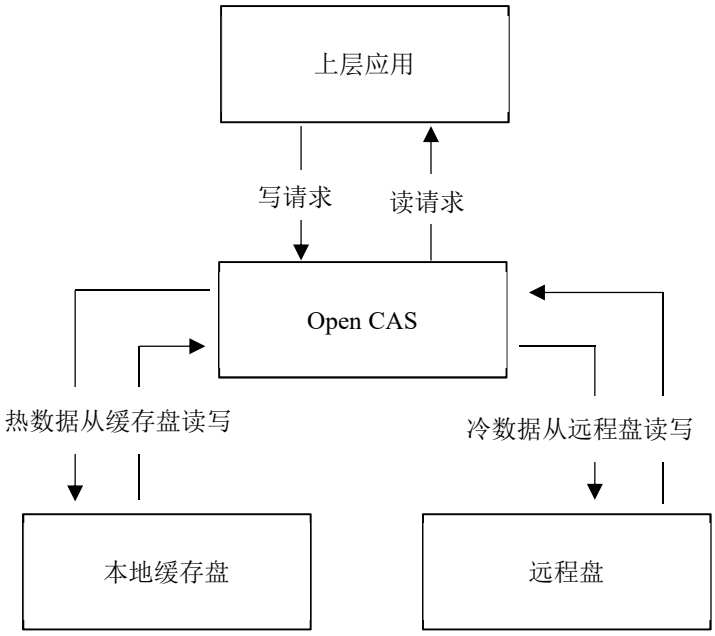
方式	平均延迟 （微秒）	P99	P99.9	P99.99	IOPS
随机读	2006.2	2343	2835	9110	19142.84
随机写	4991.5	6390	8160	14484	2496.16
随机读写(7/3)	2030.72/5438.97	2278/7046	2835/11731	20055/17171	5340.96/2292.75

优化方法及优化后性能测试

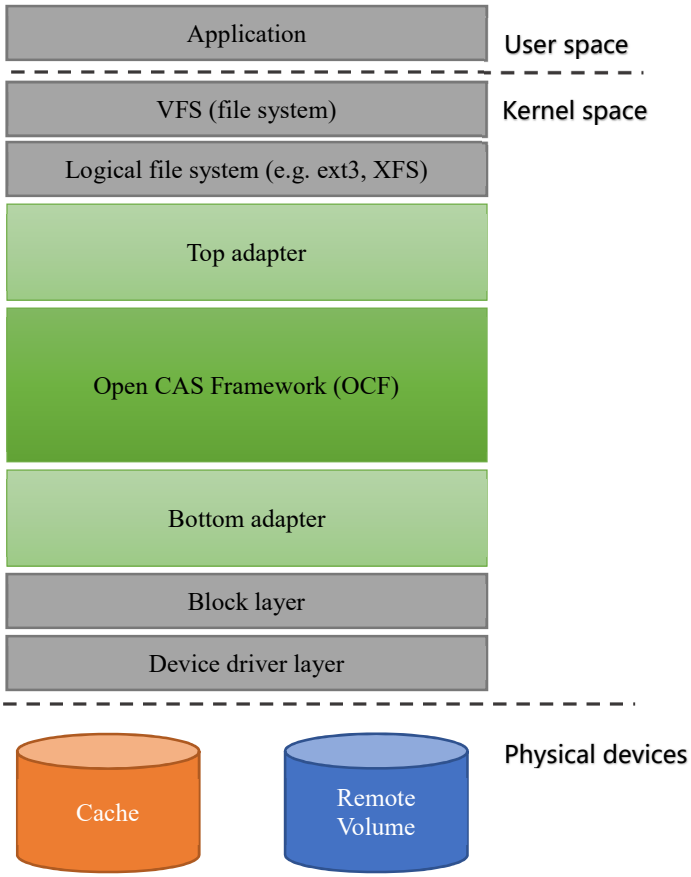
要使用 Optane™ SSD 给远程盘当缓存，需要一个成熟的本地缓存软件来实现。市面上流行的缓存软件包括 dm-cache，bcache 等，均已集成到 Linux 的内核中，非常方便使用。但在云系统上使用时，因为考虑到虚拟机热迁移等高可靠性需求，所以不能把缓存的元数据保存在远程盘上，且要保证缓存被同一个机器上的所有虚拟机共享，因此不能使用 dm-cache 和 bcache。这里我们推荐使用 Intel®共享的开源缓存软件 Open CAS。

Open CAS 介绍

Open CAS 是一个类似 Linux 里 bcache 的缓存软件，它可以使用一块或多块性能优异的块设备作为缓存盘，对同一个操作系统里的其他块设备进行缓存，从而提高系统整体的磁盘读写性能。整体的缓存架构如下图：



Open CAS 分为三层，中间的是核心框架层（OCF）。之上是适配层，能模拟出一个虚拟的块设备供上层应用使用，这里的上层应用通常指的就是文件系统，比如 ext3。之下也有个适配层，用于实际调用后端的块设备。模块在操作系统内的层次结构如右图所示：



和主流的缓存软件一样，Open CAS 也支持多种缓存模式，具体有以下几种：Write-Through (WT), Write-Around (WA), Write-Invalidate (WI), Write-Back (WB), Write-Only (WO), Pass-Through (PT)。

默认的缓存模式是 Write-Through，也就是说所有的写请求会同时写到缓存盘和远程盘（被缓存的盘）。显然这种方式对于写请求来说是没有优势的，它主要的优势是在读请求

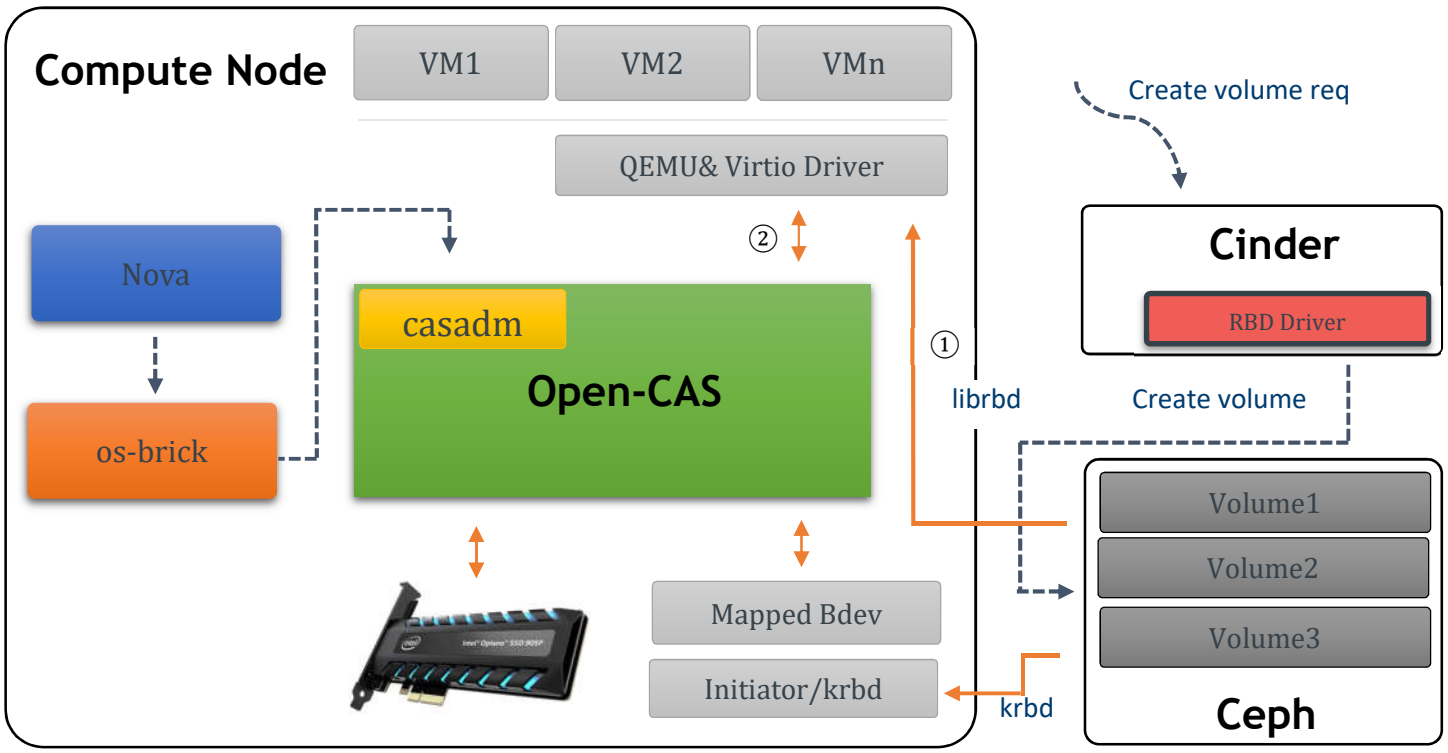
上，因为数据被缓存过了，下次读取时可以直接从缓存盘里读取，速度比较快。另外这种模式比较安全，数据一致性比较好，因为所有的数据都百分之百保存到了远程盘。

另外比较常用的缓存模式是 Write-Back (WB)。这种模式下数据会先写到缓存盘，然后就返回了。随后系统会基于一定的策略把缓存盘里的数据刷写到远程盘。这种模式会大幅提升写请求的性能，但缺点是可能存在潜在的数据一致性风险。比如某个时刻缓存盘坏了，而数据还没完全刷写到远程盘，此时远程盘里的数据就是不完整的。

关于 Open CAS 更多的详细信息可以参考官网：https://open-cas.github.io/index.html

优化方法

通过在计算节点上安装 Open CAS 软件，并且挂载一个 Intel® Optane™ SSD 作为缓存盘，来对远程挂载过来的 Ceph 盘进行缓存。Open CAS 对 Ceph 盘进行缓存后会生成一个新的虚拟盘，然后把这个虚拟盘指定给虚拟机。这样虚拟机所使用的磁盘其实是个虚拟盘，而非原来的 Ceph 盘了，从而实现了缓存功能。缓存命中与否，缓存和 Ceph 盘之间的同步均由 Open CAS 软件内部处理，外围系统只需配置缓存模式即可，不用关心具体实现细节。优化后的 OpenStack 系统架构如下图，其中①是优化前的路径，②是优化后的路径。



测试方法和环境

测试方法和前面基准测试的方法基本相同，不同的是把目标磁盘从远程的 Ceph 盘换成了 Open CAS 缓存过后虚拟出来的磁盘。

测试环境也和前述的实验基本相同，区别是计算节点本地挂载了一块 750GB 的 Intel® Optane™ P4800X SSD 作为缓存盘。该 SSD 标称的读写延迟为 10 微秒。实图如下：

Fio 详细参数如下：

```
[global]
blocksize=4k
iodepth=1 or (16 for IOPS)
ioengine=libaio
size=90G
time_based=1
ramp_time=0
runtime=60
group_reporting
thread
direct=1
numjobs=1
;-- 测试随机读 --
[test-randread]
filename=/dev/vdb
rw=randread

;-- 测试随机写 --
[test-randwrite]
filename=/dev/vdb
rw=randwrite

;-- 测试随机读写，读占70%，写占30% --
[test-randrw]
filename=/dev/vdb
rw=randrw
rwmixread=70
```

[test-randread] [test-randwrite] [test-randrw] 分别对应随机读、随机写、随机读写。做相应测试时要使能相应的代码块，而注释掉其他两个代码块。比如测试随机读时，使能[test-randread]代码块，而注释掉[test-randwrite] [test-randrw]。

性能测试及数据收集

我们分别采集了 Open CAS 的 Write-Through 和 Write-Back 缓存模式下的平均延迟、P99、P99.9、P99.99 和 IOPS 数据。

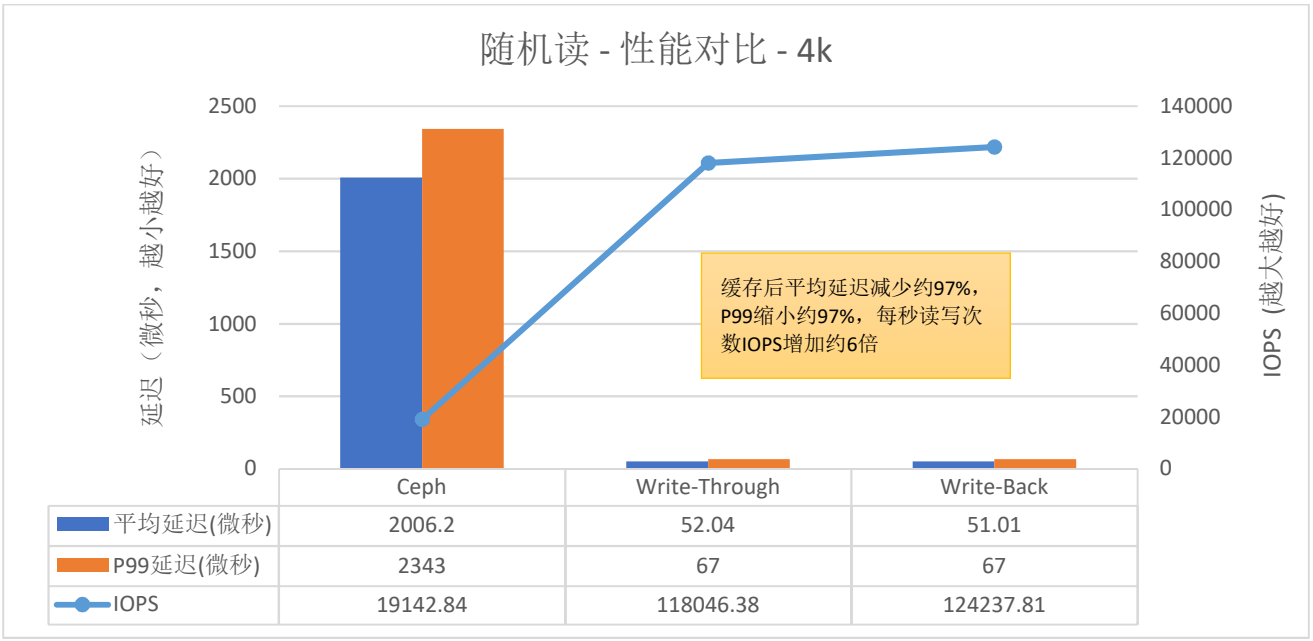
在随机读写测试结果的表格中，斜杠隔开的数值分别是读的性能数据和写的性能数据。所有延迟相关的数据，包括平均延迟、P99、P99.9、P99.99 等，单位都是微秒。IOPS 选取的请求队列深度为 16。

方式	平均延迟 （微秒）	P99	P99.9	P99.99	IOPS
缓存模式: Write-Through					
随机读	52.04	67	114	1598	118046.38
随机写	4543.23	6063	8717	12125	3185.91
随机读写(7/3)	190.39/4744.42	247/6652	265/8717	347/12256	7371.49/3163.94
缓存模式: Write-Back					
随机读	51.01	67	103	135	124237.81
随机写	118.22	194	273	314	53463.16
随机读写(7/3)	76.89/87.68	123/159	169/227	241/289	41032.94/17593.03

性能数据对比及分析

这里对以上采集的数据分别按照随机读，随机写，7:3 的随机读写进行分组对比，并形成柱状图使得测试结果更直观易懂。图中 Ceph 指的是直接对 Ceph 远程盘进行测试的结果，无缓存。Write-Through 指的是加了 Open CAS 缓存，并且缓存模式为 Write-Through。Write-Back 指的是加了 Open CAS 缓存，并且缓存模式为 Write-Back。

- **随机读性能对比分析(4k)**



	Ceph	Write-Through	Write-Back
平均延迟(微秒)	2006.2	52.04	51.01
P99延迟(微秒)	2343	67	67
IOPS	19142.84	118046.38	124237.81

测试随机读取时，循环运行 FIO 测试工具 30 次。随着测试的进行，缓存的命中率也从开始几秒时的 1.3%左右逐步提升到最后的 95%-97%。这过程中性能表现也越来越好。这里数据采用 30 次运行里表现最好的数据。以下截图是 Open CAS 软件的缓存命中统计。

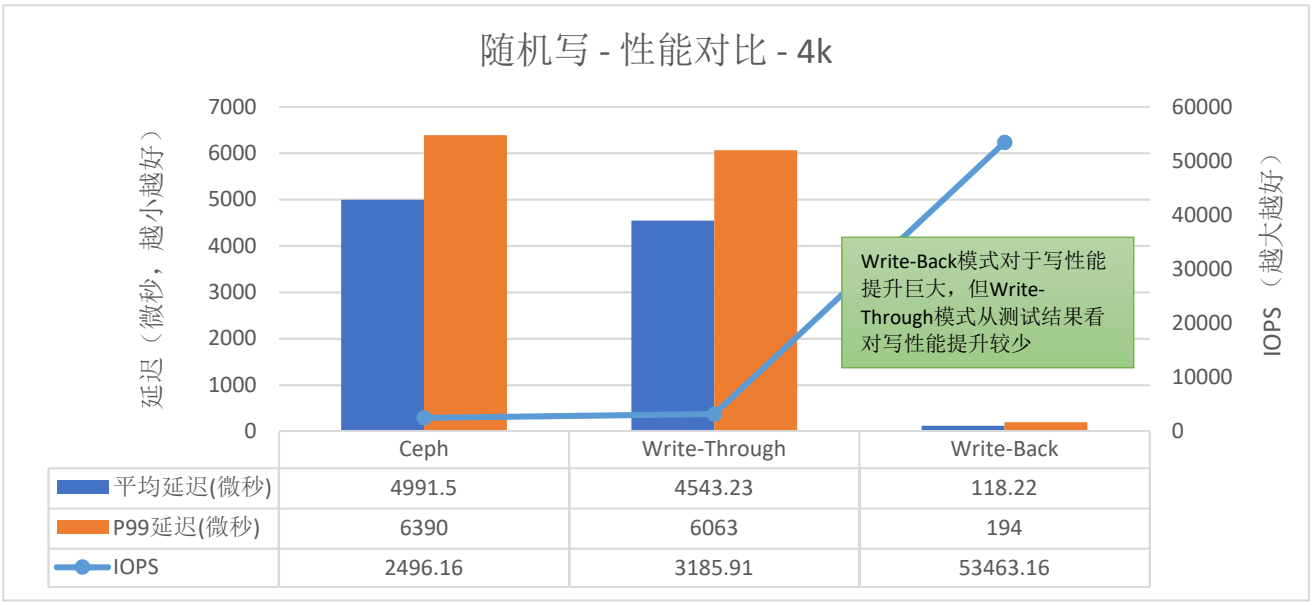
Request statistics	Count	%	Units
Read hits	203	1.3	Requests
Read partial misses	0	0.0	Requests
Read full misses	15972	98.7	Requests
Read total	16175	100.0	Requests

Request statistics	Count	%	Units
Read hits	132199632	95.0	Requests
Read partial misses	0	0.0	Requests
Read full misses	6919431	5.0	Requests
Read total	139119063	100.0	Requests

从以上的数据对比图可以看出，加了缓存之后，不管是 Write-Through 模式还是 Write-Back 模式，数据读取的延迟都有了非常大的改善。其中平均延迟从 2006.2 微秒减少到了 52.04 微秒（Write-Through）以及 51.01 微秒（Write-Back），减少了约 97%。P99 延迟数据从 2343 微秒减少到了 67 微秒（Write-Through 和 Write-Back 均是），缩小了约 97%，与此同时，每秒读请求数 IOPS 从 19142.84 提升到了 118046.38（Write-Through）以及 124237.81（Write-Back），提升了约 6 倍。

实验数据显示，Write-Through 和 Write-Back 模式在随机读取时性能非常接近，Write-Back 模式表现略好于 Write-Through 模式。

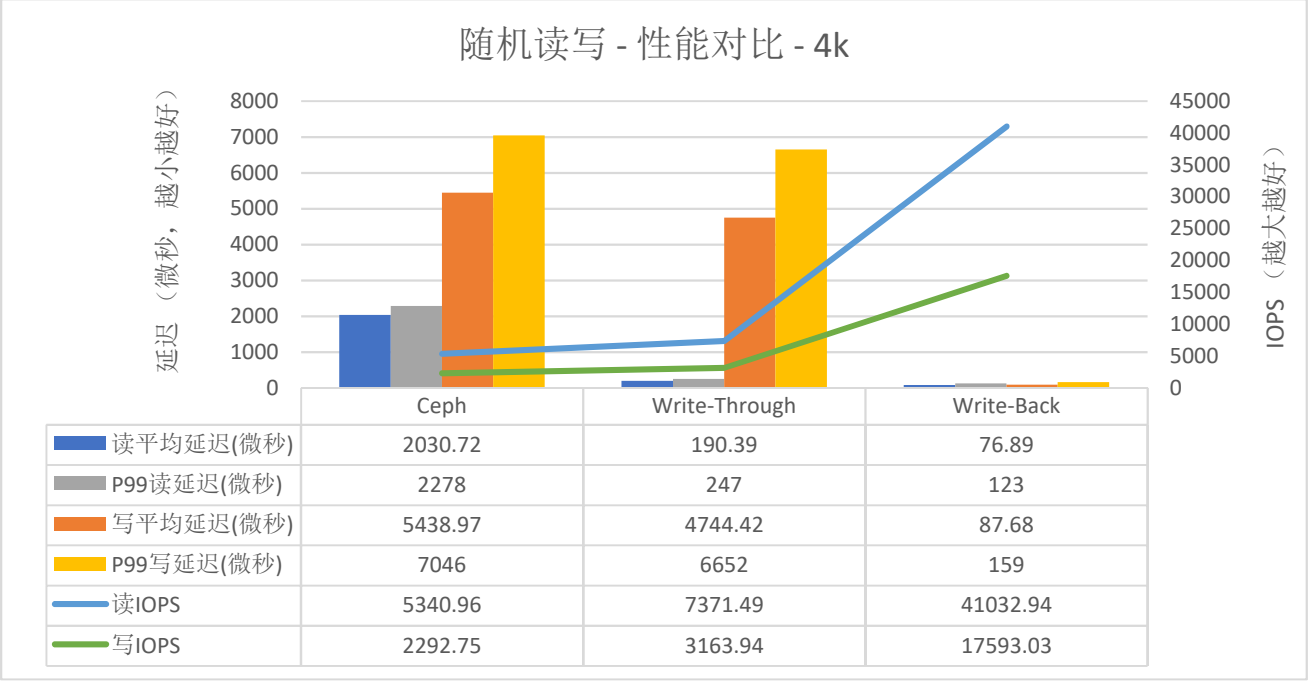
- **随机写性能对比分析(4k)**



	Ceph	Write-Through	Write-Back
平均延迟(微秒)	4991.5	4543.23	118.22
P99延迟(微秒)	6390	6063	194
IOPS	2496.16	3185.91	53463.16

对于随机写数据来说，不同缓存模式性能差距非常巨大。如果是 Write-Through 模式，如上数据显示不管是延迟还是吞吐率都没有明显提升。这也是在预期之中的，因为这种模式下所有数据都会在远程盘中落盘，和直接写远程盘差别不大。但 Open CAS 这种缓存方案要求远程 Ceph 盘必须在计算节点本地有挂载点，比如远程盘在本地挂载成/dev/rbd0，然后再对/dev/rbd0 进行缓存。所以 Write-Through 模式采用的是 krbd，而普通的 Ceph 远程盘是由 QEMU 直接挂载的，采用的是 librbd，所以性能还是略微有所区别，实验数据显示 krbd 性能略好于 librbd。但是如果改用 Write-Back 模式，那么写数据的性能会有极大的提升，平均延迟和 P99 都能减少 97%以上，每秒写请求数 IOPS 提升了 21 倍。

- **随机读写（7/3）性能对比(4k)**



	Ceph	Write-Through	Write-Back
读平均延迟(微秒)	2030.72	190.39	76.89
P99读延迟(微秒)	2278	247	123
写平均延迟(微秒)	5438.97	4744.42	87.68
P99写延迟(微秒)	7046	6652	159
读IOPS	5340.96	7371.49	41032.94
写IOPS	2292.75	3163.94	17593.03

在随机读写 7/3 比例混合的场景中，Write-Through 模式写延迟仅有 12.7%减少（5438.97 VS 4744.42），但读性能表现良好，读请求的平均延迟缩小了约 90%（2030.72 VS 190.39），P99 缩小了 89%（2278 VS 247），有效抑制了读请求的延迟波动率。而 Write-Back 模式则表现出了卓越的性能优势，读延迟减少了 96.2%（2030.72 VS 76.89），写延迟减少了 98.3%（5438.97 VS 87.68），而每秒读写请求数提升了约 7.6 倍。

总结和展望

使用 Open CAS 结合 Optane™ SSD 来做缓存，在特定场景极大提升了 OpenStack 系统的读写性能。如果应用场景有数据一致性的要求，那么选择 Write-Through 模式。此时对于读密集的应用能带来很大的性能提升。而如果应用场景是写性能要求高，或者磁盘读写综合性能要求高，但数据一致性要求不高，或者有其他的补偿措施来保证数据一致性的，那么可以选择 Write-Back 模式，此时不管读还是写的性能都可以得到非常大的提升。

联通沃云海量数据检索平台主要为读密集的使用场景，对读请求的延迟比较敏感。本次实验对其调优决策和平台升级有非常大的参考意义。

附件及参考资料

- Intel® Optane™ SSD：https://www.intel.com/content/www/us/en/products/memory-storage/solid-state-drives/data-center-ssds.html
- Ceph 官网：https://ceph.io
- Open CAS 官网：https://open-cas.github.io/index.html