但是单节点的两块缓存盘890GB， 一拖5，没有做分区，是不是和没有做分区有关系，才导致的缓存盘成为瓶颈了？

已深度思考（用时 35 秒）

好的，用户提到他们的单节点使用了两块890GB的缓存盘，采用“一拖五”的配置，并且没有对缓存盘进行分区。现在他们怀疑缓存盘成为瓶颈是否与未分区有关。我需要深入分析这个问题。

首先，我需要理解“一拖五”是什么意思。在Ceph中，通常一个缓存盘（如SSD）可能会被多个OSD共享，尤其是当使用HDD作为主存储时。这里的“一拖五”可能是指每块缓存盘服务于五个OSD，而这两块缓存盘总共支持十个OSD。但用户没有对缓存盘进行分区，可能所有OSD共享同一块缓存盘的全部容量，没有隔离。

接下来，用户观察到缓存盘的I/O使用率接近100%，这成为性能瓶颈。问题可能出在多个OSD共享同一块物理磁盘，导致I/O竞争。每个OSD的RocksDB（用于元数据存储）和BlueStore的写操作可能同时访问同一块缓存盘，导致延迟增加和吞吐量下降。

未分区的情况下，所有OSD共享同一块磁盘的I/O带宽和IOPS。如果每个OSD都有大量的元数据操作（如OMAP），或者数据写入频繁，这些并发请求会在同一磁盘上排队，导致高延迟和高利用率。尤其是当多个OSD同时进行compaction（RocksDB的压缩操作）时，I/O压力会更大。

此外，单块缓存盘的性能有限。比如，一块SATA SSD的IOPS可能在几万到十万左右，而如果多个OSD同时进行随机写入，很容易达到磁盘的上限。分区虽然不能增加磁盘的物理性能，但可以隔离不同OSD的I/O，避免相互影响。例如，为每个OSD分配独立的分区，每个分区有自己的文件系统，可能会减少I/O冲突，提高整体效率。

不过，用户使用的是Intel CAS缓存盘。Intel CAS（Cache Acceleration Software）可能用于将SSD作为缓存加速HDD，但具体配置方式会影响性能。如果CAS配置不当，可能会导致缓存效率低下，或者缓存盘本身的I/O没有被有效管理，从而成为瓶颈。

可能的解决方案包括：

1. \*\*分区隔离\*\*：为每个OSD分配独立的分区，减少I/O竞争。

2. \*\*增加缓存盘数量\*\*：如果每个OSD有独立的缓存盘，性能会更好，但成本高。

3. \*\*优化RocksDB和BlueStore配置\*\*：减少对缓存盘的I/O压力，比如调整compaction策略，减少写入放大。

4. \*\*升级硬件\*\*：使用更高性能的NVMe SSD，提供更高的IOPS和吞吐量。

5. \*\*检查Intel CAS配置\*\*：确保缓存策略合理，如缓存模式、刷新策略等。

需要验证用户是否确实在多个OSD间共享同一块缓存盘而没有隔离。如果是这样，分区或使用独立设备可能会缓解问题。另外，检查Ceph的日志和性能数据，确认是否有大量的I/O等待或队列长度过长的情况，这可以进一步确认I/O竞争的存在。

总结来说，未分区导致的多个OSD共享同一缓存盘的I/O资源，可能是瓶颈的主要原因。通过分区隔离或优化配置，可以减轻这种情况。

从您的描述来看，单节点使用 **两块 890GB 缓存盘** 以 **“一拖五”**（即每块缓存盘承载 5 个 OSD 的 RocksDB 元数据存储）且未分区的配置，确实可能是缓存盘 I/O 瓶颈的核心原因。以下是详细分析和解决方案：

### ****一、当前架构的问题分析****

#### ****1. 未分区的弊端****

**I/O 竞争**：所有 5 个 OSD 的 RocksDB（元数据存储）和 BlueStore 操作共享同一块物理磁盘的 I/O 资源，导致：

**高延迟**：多个 OSD 的随机读写请求在磁盘队列中堆积（**await** 指标高）。

**低吞吐**：单块磁盘的 IOPS 和带宽被多个 OSD 争抢，无法满足并发需求。

**缺乏隔离**：某个 OSD 的高负载操作（如 Compaction）可能阻塞其他 OSD 的 I/O。

#### ****2. 缓存盘性能瓶颈的表现****

**%util=100%**：磁盘持续满负荷运行。

**高 await（如 >50ms）**：请求排队时间过长。

**低效的 RocksDB Compaction**：多 OSD 的 Compaction 操作并发执行，进一步加剧 I/O 压力。

#### ****3. “一拖五”设计的局限性****

**硬件性能限制**：单块 SATA SSD/NVMe 的 IOPS 和带宽无法支撑 5 个 OSD 的 RocksDB 元数据操作。

**容量浪费**：未分区可能导致空间分配不均，部分 OSD 占用过多空间，其他 OSD 受限制。

### ****二、解决方案****

#### ****1. 分区隔离（推荐）****

**为每个 OSD 分配独立分区**：

bash

复制

下载

*# 使用 parted 或 fdisk 对每块缓存盘划分 5 个独立分区*parted /dev/sdj mklabel gptparted /dev/sdj mkpart primary 1MiB 178GiB *# 每个分区约 178GB（890GB/5）*parted /dev/sdj mkpart primary 178GiB 356GiB... *# 重复创建 5 个分区*

**优势**：

**I/O 隔离**：每个 OSD 的 RocksDB 使用独立分区，减少竞争。

**资源可控**：避免单个 OSD 占用过多磁盘空间。

#### ****2. 优化 RocksDB 配置****

**减少 Compaction 压力**：

ini

复制

下载

[osd]bluestore\_rocksdb\_options = "max\_background\_compactions=2,max\_background\_flushes=1"

**降低写入放大**：

ini

复制

下载

bluestore\_rocksdb\_options = "level\_compaction\_dynamic\_level\_bytes=true"

#### ****3. 调整 BlueStore 参数****

**控制刷写频率**：

ini

复制

下载

bluestore\_sync\_submit\_transaction = false # 启用异步提交bluestore\_prefer\_deferred\_size = 65536 # 增大延迟写入缓冲区

#### ****4. 硬件级优化****

**升级缓存盘**：

将 SATA SSD 更换为 **高性能 NVMe**（如 Intel Optane P5800X，随机读写 IOPS 可达百万级）。

**增加缓存盘数量**：

改为 “一拖二” 或 “一拖三”（每块盘承载更少 OSD，降低负载）。

#### ****5. 检查 Intel CAS 配置****

**确认缓存策略**：

若使用 Intel CAS 作为缓存加速层，确保缓存模式（Write-Through/Write-Back）与 Ceph 需求匹配。

**调整缓存比例**：

bash

复制

下载

*# 示例：调整 CAS 缓存分配策略*

casadm --set-param --cache-mode wt --cache-id 1

### ****三、操作步骤示例（分区隔离）****

#### ****1. 分区并格式化****

bash

复制

下载

*# 以 /dev/sdj 为例，划分为 5 个分区*sudo parted /dev/sdj mklabel gptsudo parted /dev/sdj mkpart primary 1MiB 178GiBsudo parted /dev/sdj mkpart primary 178GiB 356GiBsudo parted /dev/sdj mkpart primary 356GiB 534GiBsudo parted /dev/sdj mkpart primary 534GiB 712GiBsudo parted /dev/sdj mkpart primary 712GiB 890GiB

*# 格式化分区为 XFS（推荐）*for i in {1..5}; do

sudo mkfs.xfs /dev/sdj${i}done

#### ****2. 更新 OSD 配置****

ini

复制

下载

*# 修改每个 OSD 的 bluestore\_block\_db\_path 和 bluestore\_block\_wal\_path*[osd.0]bluestore\_block\_db\_path = /dev/sdj1bluestore\_block\_wal\_path = /dev/sdj1

[osd.1]bluestore\_block\_db\_path = /dev/sdj2bluestore\_block\_wal\_path = /dev/sdj2

...

#### ****3. 重启 OSD****

bash

复制

下载

sudo systemctl restart ceph-osd@0sudo systemctl restart ceph-osd@1...

### ****四、验证优化效果****

**检查 I/O 负载**：

bash

复制

下载

iostat -x 1 | grep sdj

目标：**%util** 降至 70% 以下，**await** < 10ms。

**观察 RocksDB 状态**：

bash

复制

下载

ceph daemon osd.0 perf dump | jq '.rocksdb'

关注 **stall\_micros** 和 **compaction** 相关指标是否下降。

### ****五、总结****

| **问题根源** | **解决方案** | **预期效果** |
| --- | --- | --- |
| 多 OSD 共享未分区缓存盘 | 为每个 OSD 分配独立分区 | 减少 I/O 竞争，降低延迟 |
| RocksDB Compaction 压力大 | 限制后台 Compaction 线程数 | 降低 I/O 峰值 |
| 硬件性能不足 | 升级为高性能 NVMe 或增加缓存盘数量 | 提升 IOPS 和带宽 |
| Intel CAS 配置不合理 | 调整缓存模式和分配策略 | 提高缓存效率 |

**建议优先执行分区隔离和硬件升级**，若仍存在瓶颈，进一步优化 RocksDB 和 BlueStore 参数。