OSD 进程自身（数据缓存、元数据、对象映射等）占用了约 4.5GB 内存，这是正常现象。 这个也不是正常现象吧，4.5GB已经很大了，另外如何查看数据缓存、元数据、对象映射的大小呢， 如何减少这部分对内存的占用

已深度思考（用时 21 秒）

好的，我需要帮助用户解决关于Ceph OSD内存占用的问题。用户之前已经询问过类似的问题，但现在对之前的回答提出了质疑，认为4.5GB的内存占用可能不正常，并希望知道如何查看和减少各部分内存的使用情况。

首先，我得回顾一下之前的对话。用户第一次询问时，我解释了OSD内存较高的原因，主要是tcmalloc的内存管理和Ceph的工作负载。用户现在质疑4.5GB是否正常，并想具体知道如何检查各部分的内存使用，以及如何减少内存占用。

用户提到“数据缓存、元数据、对象映射等”占用了4.5GB，这可能指的是BlueStore的后端内存使用，比如RocksDB的缓存或BlueFS的元数据。我需要确认Ceph中这些组件的内存管理机制，以及如何监控它们的具体使用情况。

用户的问题可以分为两部分：

1. 如何查看数据缓存、元数据、对象映射的具体内存使用？

2. 如何减少这些部分的内存占用？

首先，关于如何查看各部分内存使用，我需要提供具体的命令或方法。例如，使用ceph daemon命令查看BlueStore的统计信息，或者通过RocksDB的工具来检查缓存使用情况。可能还需要使用pmap或heap stats命令来更详细地分析内存分布。

其次，关于如何减少内存占用，我需要考虑调整Ceph的配置参数，比如降低bluestore\_cache\_size，调整RocksDB的缓存设置，或者优化OSD的并发参数。同时，用户可能需要升级Ceph版本，因为新版本可能在内存管理上有优化。另外，检查是否存在内存泄漏或过多的omap数据也是必要的。

需要注意的是，用户使用的是较旧的版本（Jewel 10.2.10），这可能存在已知的内存问题，建议升级到更新的版本。此外，用户可能需要平衡性能和内存使用，因为减少缓存可能会影响IO性能。

可能还需要解释为什么4.5GB在某些情况下是正常的，比如大型集群或高负载情况下，而如果集群规模较小，这个数值可能偏高。需要指导用户根据实际环境判断是否正常，并提供优化建议。

最后，确保回答清晰，步骤明确，并提醒用户在调整配置前备份数据，逐步测试变更效果。

对于 Ceph OSD 进程的内存占用问题，需要结合具体配置、数据负载和硬件环境综合判断。4.5GB 内存是否正常取决于集群规模、数据量、配置参数等因素，但在某些场景下可能存在优化空间。以下是详细分析和操作建议：

### ****一、判断 OSD 内存是否正常****

**常规内存占用范围**：

单个 OSD 进程的内存占用通常在 **2GB~8GB** 之间，具体取决于：

**数据量**：存储的对象数量、PG（Placement Group）数量、元数据规模。

**缓存配置**：BlueStore 的缓存大小（**bluestore\_cache\_size**）、RocksDB 的 Block Cache。

**负载压力**：高并发 IO 时内存需求会显著增加。

**4.5GB 内存是否正常**：

若集群规模较大（如数十 TB 数据、数千 PG），4.5GB 是合理的。

若集群规模较小（如仅几个 TB 数据、少量 PG），则可能偏高，需进一步排查。

**内存组成分析**：  
OSD 内存主要分为以下部分：

**BlueStore 缓存**：用于加速对象读写（元数据、数据块缓存）。

**RocksDB 内存**：存储对象的元数据（OMAP、对象映射）。

**tcmalloc 内存池**：动态内存分配的缓存（如线程缓存、中央缓存）。

**进程基础内存**：代码段、堆栈等固定开销。

### ****二、查看各部分内存占用的方法****

#### ****1. 查看 BlueStore 缓存和 RocksDB 内存****

通过 **ceph daemon** 命令获取 OSD 内部统计信息：

bash

复制

下载

*# 查看 BlueStore 缓存使用情况*

ceph daemon osd.<ID> perf dump | jq '.bluestore'

*# 查看 RocksDB 内存占用（BlueStore 后端元数据存储）*

ceph daemon osd.<ID> perf dump | jq '.rocksdb'

关键字段解释：

**bluestore\_ondisk**: 磁盘上的 BlueStore 数据总量。

**bluestore\_cache\_bytes**: BlueStore 缓存大小（含元数据和数据块缓存）。

**rocksdb\_block\_cache\_usage**: RocksDB 的 Block Cache 内存占用。

**rocksdb\_estimate\_table\_readers\_mem**: RocksDB 表读取器的内存占用。

#### ****2. 查看 OSD 进程内存分布****

使用 **pmap** 分析进程内存映射：

bash

复制

下载

pmap -x $(pgrep -f ceph-osd) | less

查找 **[heap]**、**[anon]** 等段的内存占用，重点关注大块内存区域。

#### ****3. 查看 tcmalloc 内存分配****

通过 **heap stats** 命令（需开启 tcmalloc 调试）：

bash

复制

下载

ceph tell osd.<ID> heap stats

输出中的 **Bytes in use by application** 是实际应用内存占用，其余为 tcmalloc 缓存。

### ****三、减少内存占用的优化方法****

#### ****1. 调整 BlueStore 缓存大小****

修改 BlueStore 的缓存上限（需重启 OSD）：

bash

复制

下载

*# 在 ceph.conf 中为 OSD 添加配置*[osd.<ID>]

bluestore\_cache\_size = 2G *# 默认值为 1~4GB，根据内存总量调整*

#### ****2. 优化 RocksDB 配置****

减少 RocksDB 的 Block Cache 大小（BlueStore 后端）：

bash

复制

下载

*# 在 ceph.conf 中配置 RocksDB 参数*[osd]

bluestore\_rocksdb\_options = "compression=kNoCompression,max\_write\_buffer\_number=4,write\_buffer\_size=256MB,block\_size=16KB"

**max\_write\_buffer\_number**: 减少写入缓冲区数量。

**write\_buffer\_size**: 降低单个缓冲区大小。

#### ****3. 控制并发和线程数****

限制 OSD 线程和并发操作：

bash

复制

下载

[osd]

osd\_op\_num\_threads = 2 *# 默认 2，避免过高*

osd\_peering\_max\_active = 2 *# 控制 PG 恢复并发数*

#### ****4. 释放 tcmalloc 内存缓存****

手动释放 tcmalloc 保留的未使用内存：

bash

复制

下载

ceph tell osd.<ID> heap release\_free\_memory

#### ****5. 检查 OMAP 数据量****

过多的 OMAP（对象扩展属性）会增大 RocksDB 内存占用：

bash

复制

下载

*# 查看 OSD 的 OMAP 数据量*

rados -p <pool-name> listomapkeys <object-id> | wc -l

若 OMAP 数据过多，可考虑合并小对象或清理无用元数据。

#### ****6. 升级 Ceph 版本****

较老版本（如 Jewel 10.2.10）可能存在内存管理缺陷，建议升级至 **Nautilus (14.2.x)** 或更高版本，新版对 BlueStore 和 RocksDB 有显著优化。

#### ****7. 硬件层面优化****

**增加内存**：若物理内存不足，直接扩容是最直接的方案。

**使用 NVMe 或 SSD**：减少对内存缓存的依赖，提升 RocksDB 性能。

### ****四、判断是否需紧急处理****

**内存不足导致 OOM**：若 OSD 频繁被 OOM Killer 终止，需立即优化。

**内存持续增长不释放**：可能存在内存泄漏，需检查日志或升级版本。

**性能未受影响**：若集群运行稳定且内存充足，可保持观察。

### ****总结****

**4.5GB 内存是否正常**：需结合集群规模和数据量判断，小型集群可进一步优化。

**关键操作**：

通过 **ceph daemon** 和 **pmap** 定位内存占用来源。

调整 **bluestore\_cache\_size** 和 RocksDB 参数。

手动释放 tcmalloc 缓存或升级 Ceph 版本。

监控优化后的内存变化和性能影响。

建议在调整配置前备份 **ceph.conf**，并逐个参数测试，避免一次性修改过多导致性能波动。

通过pmap命令获取ceph osd.0 进程的堆内存分析 pmap -x 66863 > ceph-osd.0 输出结果如下： 66863: /usr/bin/ceph-osd -f --cluster ceph --id 0 --setuser ceph --setgroup ceph Address Kbytes RSS Dirty Mode Mapping 0000563459483000 14028 4572 0 r-x-- ceph-osd 000056345a436000 152 152 152 r---- ceph-osd 000056345a45c000 96 88 76 rw--- ceph-osd 000056345a474000 147492 28 28 rw--- [ anon ] 00005634652bb000 8518420 7365176 7365176 rw--- [ anon ] 00007f4e04f55000 2088 0 0 r--s- 3164986.sst 00007f4e06386000 2088 0 0 r--s- 3164407.sst 00007f4e06590000 2088 0 0 r--s- 3164406.sst 00007f4e0679a000 2088 0 0 r--s- 3164405.sst 00007f4e069a4000 2088 0 0 r--s- 3164404.sst 00007f4e06bae000 2088 0 0 r--s- 3164403.sst 00007f4e06db8000 2088 0 0 r--s- 3164402.sst 00007f4e071cc000 2088 0 0 r--s- 3164985.sst 00007f4e07dfc000 2088 0 0 r--s- 3165967.sst 00007f4e08006000 2088 0 0 r--s- 3164401.sst 00007f4e0841b000 2088 0 0 r--s- 3164400.sst 00007f4e08625000 2088 0 0 r--s- 3165966.sst 00007f4e0882f000 2088 0 0 r--s- 3164947.sst 00007f4e08a39000 2088 0 0 r--s- 3163569.sst 00007f4e08e48000 2092 0 0 r--s- 3165965.sst 00007f4e09467000 2088 0 0 r--s- 3164896.sst 00007f4e09671000 2092 0 0 r--s- 3164895.sst 00007f4e0987c000 2088 0 0 r--s- 3165539.sst 00007f4e09e98000 2088 0 0 r--s- 3165538.sst 00007f4e0b504000 2088 0 0 r--s- 3165964.sst 00007f4e0bb22000 2092 0 0 r--s- 3165465.sst 00007f4e0bf38000 2088 0 0 r--s- 3165327.sst 00007f4e0cf86000 2088 0 0 r--s- 3163501.sst 00007f4e0d191000 2092 0 0 r--s- 3165825.sst 00007f4e0dbc3000 2088 0 0 r--s- 3165326.sst 00007f4e0ddcd000 2088 32 0 r--s- 3164984.sst 00007f4e0e3ea000 2092 0 0 r--s- 3164945.sst 00007f4e0efb8000 4 0 0 ----- [ anon ] 00007f4e0efb9000 1024 20 20 rw--- [ anon ] 00007f4e10e33000 4 0 0 ----- [ anon ] 00007f4e10e34000 1024 20 20 rw--- [ anon ] 00007f4e1213e000 2088 40 0 r--s- 3167932.sst 00007f4e12348000 4 0 0 ----- [ anon ] 00007f4e12349000 1024 20 20 rw--- [ anon ] 00007f4e1244b000 1984 40 0 r--s- 3167830.sst 00007f4e1263b000 2088 0 0 r--s- 3167681.sst 00007f4e12845000 2088 4 0 r--s- 3165963.sst 00007f4e12a4f000 4 0 0 ----- [ anon ] 00007f4e12a50000 1024 20 20 rw--- [ anon ] 00007f4e12d32000 2088 0 0 r--s- 3167680.sst 00007f4e12f3c000 2088 0 0 r--s- 3167678.sst 00007f4e13146000 2088 0 0 r--s- 3167677.sst 00007f4e13350000 2088 0 0 r--s- 3167676.sst 00007f4e1355a000 4 0 0 ----- [ anon ] 00007f4e1355b000 1024 20 20 rw--- [ anon ] 00007f4e1365b000 4 0 0 ----- [ anon ] 00007f4e1365c000 1024 20 20 rw--- [ anon ] 00007f4e1383d000 2088 40 0 r--s- 3167829.sst 00007f4e13a47000 2088 0 0 r--s- 3167638.sst 00007f4e13c51000 2088 40 0 r--s- 3167581.sst 00007f4e13e5b000 2088 0 0 r--s- 3165325.sst 00007f4e14136000 2088 0 0 r--s- 3167636.sst 00007f4e14340000 2088 0 0 r--s- 3167635.sst 00007f4e1463d000 2088 40 0 r--s- 3167827.sst 00007f4e14847000 2088 40 0 r--s- 3167826.sst 00007f4e14a51000 2088 4 0 r--s- 3167633.sst 00007f4e14d72000 4 0 0 ----- [ anon ] 00007f4e14d73000 1024 16 16 rw--- [ anon ] 00007f4e14f05000 2088 40 0 r--s- 3167930.sst 00007f4e1510f000 1932 40 0 r--s- 3167582.sst 00007f4e152f2000 2088 0 0 r--s- 3167674.sst 00007f4e15664000 2088 40 0 r--s- 3167749.sst 00007f4e1586e000 2088 40 0 r--s- 3167747.sst 00007f4e15a78000 2084 36 0 r--s- 3167746.sst 00007f4e15c81000 4 0 0 ----- [ anon ] 00007f4e15c82000 1024 20 20 rw--- [ anon ] 00007f4e15e7c000 2084 36 0 r--s- 3167745.sst 00007f4e16085000 4 0 0 ----- [ anon ] 00007f4e16086000 1024 20 20 rw--- [ anon ] 00007f4e16572000 2088 40 0 r--s- 3167652.sst 00007f4e1677c000 2088 40 0 r--s- 3167651.sst 00007f4e16986000 2088 40 0 r--s- 3167650.sst 00007f4e16b90000 4 0 0 ----- [ anon ] 00007f4e16b91000 1024 20 20 rw--- [ anon ] 00007f4e16e5f000 2088 0 0 r--s- 3167673.sst 00007f4e17069000 2088 0 0 r--s- 3167671.sst 00007f4e173c0000 4 0 0 ----- [ anon ] 00007f4e173c1000 1024 16 16 rw--- [ anon ] 00007f4e174c1000 4 0 0 ----- [ anon ] 00007f4e174c2000 1024 16 16 rw--- [ anon ] 00007f4e175c2000 2088 0 0 r--s- 3165730.sst 00007f4e177da000 4 0 0 ----- [ anon ] 00007f4e177db000 1024 16 16 rw--- [ anon ] 00007f4e178db000 4 0 0 ----- [ anon ] 00007f4e178dc000 1024 16 16 rw--- [ anon ] 00007f4e179dc000 4 0 0 ----- [ anon ] 00007f4e179dd000 1024 16 16 rw--- [ anon ] 00007f4e17add000 4 0 0 ----- [ anon ] 00007f4e17ade000 1024 12 12 rw--- [ anon ] 00007f4e17df9000 4 0 0 ----- [ anon ] 00007f4e17dfa000 1024 12 12 rw--- [ anon ] 00007f4e17efa000 4 0 0 ----- [ anon ] 00007f4e17efb000 1024 20 20 rw--- [ anon ] 00007f4e17ffb000 4 0 0 ----- [ anon ] 00007f4e17ffc000 1024 20 20 rw--- [ anon ] 00007f4e180fc000 4 0 0 ----- [ anon ] 00007f4e180fd000 1024 16 16 rw--- [ anon ] 00007f4e181fd000 2088 0 0 r--s- 3165823.sst 00007f4e18407000 2088 0 0 r--s- 3165280.sst 00007f4e18611000 4 0 0 ----- [ anon ] 00007f4e18612000 1024 20 20 rw--- [ anon ] 00007f4e1871a000 4 0 0 ----- [ anon ] 00007f4e1871b000 1024 16 16 rw--- [ anon ] 00007f4e1881b000 2088 0 0 r--s- 3165800.sst 00007f4e18a25000 2088 0 0 r--s- 3165245.sst 00007f4e18c36000 4 0 0 ----- [ anon ] 00007f4e18c37000 1024 16 16 rw--- [ anon ] 00007f4e18d37000 4 0 0 ----- [ anon ] 00007f4e18d38000 1024 16 16 rw--- [ anon ] 00007f4e18e38000 2088 0 0 r--s- 3165279.sst 00007f4e19042000 4 0 0 ----- [ anon ] 00007f4e19043000 1024 20 20 rw--- [ anon ] 00007f4e191eb000 4 0 0 ----- [ anon ] 00007f4e191ec000 1024 16 16 rw--- [ anon ] 00007f4e192ec000 4 0 0 ----- [ anon ] 00007f4e192ed000 1024 16 16 rw--- [ anon ] 00007f4e193ed000 2088 4 0 r--s- 3165729.sst 00007f4e195f7000 2088 0 0 r--s- 3164983.sst 00007f4e19809000 4 0 0 ----- [ anon ] 00007f4e1980a000 1024 16 16 rw--- [ anon ] 00007f4e1990a000 4 0 0 ----- [ anon ] 00007f4e1990b000 1024 16 16 rw--- [ anon ] 00007f4e19a0b000 2088 0 0 r--s- 3164982.sst 00007f4e19c1d000 4 0 0 ----- [ anon ] 00007f4e19c1e000 1024 12 12 rw--- [ anon ] 00007f4e19d1e000 4 0 0 ----- [ anon ] 00007f4e19d1f000 1024 16 16 rw--- [ anon ] 00007f4e19e1f000 4 0 0 ----- [ anon ] 00007f4e19e20000 1024 16 16 rw--- [ anon ] 00007f4e19f20000 2088 0 0 r--s- 3165728.sst 00007f4e1a131000 4 0 0 ----- [ anon ] 00007f4e1a132000 1024 16 16 rw--- [ anon ] 00007f4e1a232000 2092 8 0 r--s- 3164789.sst 00007f4e1a445000 4 0 0 ----- [ anon ] 00007f4e1a446000 1024 20 20 rw--- [ anon ] 00007f4e1a546000 4 0 0 ----- [ anon ] 00007f4e1a547000 1024 16 16 rw--- [ anon ] 00007f4e1a647000 2088 0 0 r--s- 3164523.sst 00007f4e1a856000 4 0 0 ----- [ anon ] 00007f4e1a857000 1024 20 20 rw--- [ anon ] 00007f4e1a957000 4 0 0 ----- [ anon ] 00007f4e1a958000 1024 16 16 rw--- [ anon ] 00007f4e1aa58000 2088 0 0 r--s- 3163568.sst 00007f4e1ac62000 2088 0 0 r--s- 3163963.sst 00007f4e1ae6c000 2092 388 0 r--s- 3164980.sst 00007f4e1b077000 2088 0 0 r--s- 3164037.sst 00007f4e1b289000 4 0 0 ----- [ anon ] 00007f4e1b28a000 1024 16 16 rw--- [ anon ] 00007f4e1b38a000 4 0 0 ----- [ anon ] 00007f4e1b38b000 1024 16 16 rw--- [ anon ] 00007f4e1b48b000 2088 0 0 r--s- 3164981.sst 00007f4e1b695000 2088 0 0 r--s- 3164128.sst 00007f4e1b89f000 2088 0 0 r--s- 3163500.sst 00007f4e1bb86000 4 0 0 ----- [ anon ] 00007f4e1bb87000 1024 20 20 rw--- [ anon ] 00007f4e1bc87000 4 0 0 ----- [ anon ] 00007f4e1bc88000 1024 16 16 rw--- [ anon ] 00007f4e1bd88000 2088 0 0 r--s- 3163533.sst 00007f4e1bf92000 4 0 0 ----- [ anon ] 00007f4e1bf93000 1024 16 16 rw--- [ anon ] 00007f4e1c093000 4 0 0 ----- [ anon ] 00007f4e1c094000 1024 20 20 rw--- [ anon ] 00007f4e1c3ab000 4 0 0 ----- [ anon ] 00007f4e1c3ac000 1024 12 12 rw--- [ anon ] 00007f4e1c4ac000 4 0 0 ----- [ anon ] 00007f4e1c4ad000 1024 16 16 rw--- [ anon ] 00007f4e1c5ad000 4 0 0 ----- [ anon ] 00007f4e1c5ae000 1024 16 16 rw--- [ anon ] 00007f4e1c6ae000 2092 0 0 r--s- 3165196.sst 00007f4e1c8b9000 2088 0 0 r--s- 3165244.sst 00007f4e1cac3000 2088 0 0 r--s- 3165243.sst 00007f4e1cccd000 2088 0 0 r--s- 3163372.sst 00007f4e1cefa000 2088 0 0 r--s- 3164036.sst 00007f4e1d104000 2088 0 0 r--s- 3163371.sst 00007f4e1d30e000 2088 0 0 r--s- 3164894.sst 00007f4e1d5fd000 4 0 0 ----- [ anon ] 00007f4e1d5fe000 1024 16 16 rw--- [ anon ] 00007f4e1d6fe000 2088 0 0 r--s- 3165278.sst 00007f4e1d916000 4 0 0 ----- [ anon ] 00007f4e1d917000 1024 12 12 rw--- [ anon ] 00007f4e1da17000 4 0 0 ----- [ anon ] 00007f4e1da18000 1024 16 16 rw--- [ anon ] 00007f4e1db18000 4 0 0 ----- [ anon ] 00007f4e1db19000 1024 12 12 rw--- [ anon ] 00007f4e1dc19000 4 0 0 ----- [ anon ] 00007f4e1dc1a000 1024 12 12 rw--- [ anon ] 00007f4e1dd1a000 2088 40 0 r--s- 3167580.sst 00007f4e1df24000 2088 40 0 r--s- 3167579.sst 00007f4e1e12e000 2088 0 0 r--s- 3164035.sst 00007f4e1e338000 2088 0 0 r--s- 3165084.sst 00007f4e1e542000 2088 0 0 r--s- 3165822.sst 00007f4e1e74c000 2088 0 0 r--s- 3165684.sst 00007f4e1e956000 2088 0 0 r--s- 3163532.sst 00007f4e1eb60000 2088 0 0 r--s- 3164946.sst 00007f4e1ed6b000 2088 0 0 r--s- 3165962.sst 00007f4e1ef75000 2088 0 0 r--s- 3165083.sst 00007f4e1f19e000 1960 40 0 r--s- 3167828.sst 00007f4e1f388000 2088 0 0 r--s- 3163369.sst 00007f4e1f592000 2088 0 0 r--s- 3163499.sst 00007f4e1f7d3000 4 0 0 ----- [ anon ] 00007f4e1f7d4000 1024 12 12 rw--- [ anon ] 00007f4e1f8d4000 2088 0 0 r--s- 3163962.sst 00007f4e1fae0000 2088 40 0 r--s- 3167928.sst 00007f4e1fcea000 2088 40 0 r--s- 3167938.sst 00007f4e1fef4000 2088 0 0 r--s- 3164399.sst 00007f4e201fc000 2088 0 0 r--s- 3164944.sst 00007f4e20408000 2084 36 0 r--s- 3167927.sst 00007f4e20611000 2088 40 0 r--s- 3167937.sst 00007f4e2081b000 2088 0 0 r--s- 3165277.sst 00007f4e20a25000 2088 0 0 r--s- 3163901.sst 00007f4e20c30000 1608 36 0 r--s- 3167919.sst 00007f4e20dc2000 1280 0 0 r--s- 3167682.sst 00007f4e21042000 2092 0 0 r--s- 3167370.sst 00007f4e2128c000 1324 28 0 r--s- 3167918.sst 00007f4e213d7000 1524 32 0 r--s- 3167917.sst 00007f4e2175d000 2088 0 0 r--s- 3165609.sst 00007f4e21967000 2088 0 0 r--s- 3164790.sst 00007f4e21b71000 2088 0 0 r--s- 3164127.sst 00007f4e21d7b000 2088 0 0 r--s- 3164750.sst 00007f4e22059000 4 0 0 ----- [ anon ] 00007f4e2205a000 1024 16 16 rw--- [ anon ] 00007f4e2215a000 2088 44 0 r--s- 3167916.sst 00007f4e22364000 2088 0 0 r--s- 3164641.sst 00007f4e2256e000 2088 0 0 r--s- 3163900.sst 00007f4e22778000 2088 40 0 r--s- 3167936.sst 00007f4e22982000 2088 0 0 r--s- 3163899.sst 00007f4e22bf0000 2088 0 0 r--s- 3164640.sst 00007f4e22e53000 1732 36 0 r--s- 3167825.sst 00007f4e23004000 2088 0 0 r--s- 3163898.sst 00007f4e2320e000 2088 0 0 r--s- 3165537.sst 00007f4e234b8000 4 0 0 ----- [ anon ] 00007f4e234b9000 1024 12 12 rw--- [ anon ] 00007f4e235b9000 2088 40 0 r--s- 3167925.sst 00007f4e237c3000 2088 0 0 r--s- 3164522.sst 00007f4e239cf000 2084 36 0 r--s- 3167744.sst 00007f4e23bd8000 2088 0 0 r--s- 3163897.sst 00007f4e23de9000 4 0 0 ----- [ anon ] 00007f4e23dea000 1024 12 12 rw--- [ anon ] 00007f4e23eea000 4 0 0 ----- [ anon ] 00007f4e23eeb000 1024 12 12 rw--- [ anon ] 00007f4e23fed000 2088 40 0 r--s- 3167924.sst 00007f4e241f7000 2084 36 0 r--s- 3167923.sst 00007f4e24609000 2088 0 0 r--s- 3163531.sst 00007f4e24813000 2088 0 0 r--s- 3166065.sst 00007f4e24a1d000 2088 0 0 r--s- 3163421.sst 00007f4e24d07000 2092 44 0 r--s- 3167934.sst 00007f4e24f12000 2092 0 0 r--s- 3163961.sst 00007f4e251dc000 2088 0 0 r--s- 3165536.sst 00007f4e253e6000 2088 0 0 r--s- 3164094.sst 00007f4e255f2000 2088 40 0 r--s- 3167835.sst 00007f4e257fc000 2088 0 0 r--s- 3164126.sst 00007f4e25a0d000 4 0 0 ----- [ anon ] 00007f4e25a0e000 1024 12 12 rw--- [ anon ] 00007f4e25b0e000 4 0 0 ----- [ anon ] 00007f4e25b0f000 1024 12 12 rw--- [ anon ] 00007f4e25c0f000 2088 4 0 r--s- 3166091.sst 00007f4e25e19000 2088 0 0 r--s- 3164943.sst

已深度思考（用时 22 秒）

嗯，用户给了一个关于Ceph OSD内存分析的pmap输出，需要我帮忙分析堆内存的情况，以及如何减少内存占用。首先，我得仔细看看pmap的输出内容，找出哪些部分占用了最多的内存。

从输出中，我看到有一个非常大的匿名内存区域，地址是00005634652bb000，占用了8518420 KB，也就是大约8.1 GB，其中RSS（实际驻留内存）是7365176 KB，约7 GB，而且都是脏页。这说明这个区域是OSD进程的主要堆内存占用，可能包含了BlueStore的缓存、RocksDB的数据结构等。

接下来，用户提到之前讨论过OSD内存占用高的问题，现在想通过pmap进一步确认堆内存的使用情况，并找到减少内存的方法。我需要回忆一下Ceph的内存管理机制，特别是BlueStore和RocksDB的内存使用情况。

BlueStore使用内存来缓存元数据和数据块，这部分由bluestore\_cache\_size参数控制。RocksDB作为BlueStore的后端存储元数据，其Block Cache也会占用较多内存。此外，tcmalloc的内存池机制会保留一些内存，不立即释放给操作系统，可能导致内存看起来较高。

用户提供的pmap输出中，大量的.sst文件映射可能是RocksDB的SSTable文件，这些文件被映射到内存中，但RSS显示为0或很小，说明它们可能没有被频繁访问，只是被mmap到地址空间，实际占用物理内存较少。主要的内存占用还是来自那个8GB的匿名区域。

接下来，用户想知道如何减少这部分内存。我需要考虑调整BlueStore和RocksDB的配置，比如减少缓存大小，优化RocksDB的参数，或者手动释放tcmalloc的内存。另外，检查是否有过多的OMAP数据，或者是否需要升级Ceph版本，因为旧版本可能存在内存管理的问题。

需要注意的是，调整这些参数可能会影响性能，比如减少缓存可能导致IO延迟增加。所以建议逐步调整，并监控效果。另外，手动释放tcmalloc的内存虽然可以暂时减少占用，但可能会影响后续性能，因为tcmalloc需要重新分配内存。

最后，用户可能是在运维一个Ceph集群，遇到了内存使用过高的问题，需要优化配置以确保集群的稳定运行。可能需要综合使用配置调整、版本升级和硬件优化来解决。

从提供的 **pmap** 输出来看，OSD 进程 **osd.0** 的内存分配主要集中在 **堆内存（Heap）** 和 **内存映射文件（如 RocksDB 的 SST 文件）**。以下是详细分析和优化建议：

### ****一、堆内存分析****

#### ****关键内存段定位****

在输出中，以下两段匿名内存（**[anon]**）占用了绝大部分物理内存（RSS）：

text

复制

下载

Address Kbytes RSS Dirty Mode Mapping

000056345a474000 147492 28 28 rw--- [ anon ] # ~144MB（几乎未使用）

00005634652bb000 8518420 7365176 7365176 rw--- [ anon ] # ~7.3GB（主要堆内存）

**地址 00005634652bb000 的匿名内存段**：

占用 **8,518,420 KB（约 8.1GB）虚拟内存**，实际物理内存（RSS）为 **7,365,176 KB（约 7GB）**。

**Dirty 内存** 同样为 7.3GB，表明这部分内存已被修改且未同步到磁盘（如 BlueStore 缓存、RocksDB 元数据等）。

这是 OSD 进程的主要堆内存区域，包含：

**BlueStore 缓存**（数据块和元数据缓存）。

**RocksDB 内存占用**（对象元数据、OMAP 数据）。

**tcmalloc 内存池**（动态分配的内存缓存）。

#### ****其他内存段****

**.sst 文件映射**（如 **00007f4e04f55000** 等）：

虽然虚拟内存占用较大（每个约 2MB），但 **RSS 接近 0**，说明这些 RocksDB 的 SST 文件未被频繁访问，仅通过 **mmap** 映射到内存，物理内存占用极少。

无需重点关注，除非 RSS 显著增长。

### ****二、内存占用原因****

**BlueStore 缓存**：

BlueStore 默认使用内存缓存热数据块和元数据，由 **bluestore\_cache\_size** 参数控制（默认值通常为 1~4GB）。

如果未显式限制缓存大小，可能根据系统内存自动增长。

**RocksDB 内存占用**：

RocksDB 作为 BlueStore 的元数据存储引擎，其 Block Cache 和 Memtable 会占用内存。

OMAP（对象扩展属性）数据量较大时，RocksDB 的内存需求显著增加。

**tcmalloc 内存池**：

tcmalloc 为提升性能会保留空闲内存（见 **heap stats** 中的 **freelist**），未及时释放给操作系统。

### ****三、优化建议****

#### ****1. 限制 BlueStore 缓存大小****

在 **ceph.conf** 中显式设置 BlueStore 缓存上限（需重启 OSD）：

ini

复制

下载

[osd.0]bluestore\_cache\_size = 2G # 根据系统内存调整（如分配 2~4GB）

#### ****2. 优化 RocksDB 配置****

减少 RocksDB 的 Block Cache 和写入缓冲区大小：

ini

复制

下载

[osd]*# 调整 RocksDB 参数*bluestore\_rocksdb\_options = "block\_cache\_size=1GB,max\_write\_buffer\_number=4,write\_buffer\_size=256MB"

**block\_cache\_size**: Block Cache 大小（默认占 **bluestore\_cache\_size** 的 25%）。

**max\_write\_buffer\_number**: 写入缓冲区数量（默认 4，无需调整）。

**write\_buffer\_size**: 单个写入缓冲区大小（默认 64MB，可适当降低）。

#### ****3. 释放 tcmalloc 空闲内存****

手动释放 tcmalloc 保留的未使用内存：

bash

复制

下载

ceph tell osd.0 heap release\_free\_memory

#### ****4. 检查 OMAP 数据量****

过多的 OMAP 数据会增大 RocksDB 内存压力：

bash

复制

下载

*# 列出所有对象的 OMAP 数量*

rados -p <pool-name> listomapkeys <object-id> | wc -l

若 OMAP 数据过多，考虑合并小对象或清理无用元数据。

#### ****5. 升级 Ceph 版本****

Ceph Jewel (10.2.10) 是较老版本，存在已知内存管理问题。建议升级至 **Nautilus (14.2.x)** 或更高版本，新版本对 BlueStore 和 RocksDB 有显著优化。

#### ****6. 监控内存变化****

使用以下命令持续监控内存：

bash

复制

下载

*# 实时内存统计*

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

*# tcmalloc 内存详情*

ceph tell osd.0 heap stats

### ****四、关键操作验证****

#### ****1. 调整 BlueStore 缓存后验证****

重启 OSD 后观察堆内存变化：

bash

复制

下载

systemctl restart ceph-osd@0

pmap -x $(pgrep -f ceph-osd) | grep '[ anon ]'

预期效果：**00005634652bb000** 的 RSS 应降至与 **bluestore\_cache\_size** 接近。

#### ****2. 释放 tcmalloc 内存后验证****

执行释放命令后检查 **heap stats**：

bash

复制

下载

ceph tell osd.0 heap stats

预期效果：**Bytes in page heap freelist** 和 **Bytes in thread cache freelists** 显著减少。

### ****五、总结****

**主要内存占用**：集中在 BlueStore 缓存和 RocksDB 元数据（约 7.3GB）。

**优化方向**：

显式限制 **bluestore\_cache\_size**。

优化 RocksDB 参数减少内存占用。

定期释放 tcmalloc 空闲内存。

检查并清理冗余 OMAP 数据。

升级到新版本 Ceph。

建议逐步实施优化措施，并监控内存和性能变化，避免一次性调整过多参数导致服务不稳定

重启osd.0 前后内存变化关系 重启前： [root@rg2-ceph001 ~]# ceph tell osd.0 heap stats osd.0 tcmalloc heap stats:------------------------------------------------ MALLOC: 4760644592 ( 4540.1 MiB) Bytes in use by application MALLOC: + 138149888 ( 131.8 MiB) Bytes in page heap freelist MALLOC: + 1492099536 ( 1423.0 MiB) Bytes in central cache freelist MALLOC: + 290816 ( 0.3 MiB) Bytes in transfer cache freelist MALLOC: + 1113121344 ( 1061.6 MiB) Bytes in thread cache freelists MALLOC: + 72744960 ( 69.4 MiB) Bytes in malloc metadata MALLOC: ------------ MALLOC: = 7577051136 ( 7226.0 MiB) Actual memory used (physical + swap) MALLOC: + 1143054336 ( 1090.1 MiB) Bytes released to OS (aka unmapped) MALLOC: ------------ MALLOC: = 8720105472 ( 8316.1 MiB) Virtual address space used MALLOC: MALLOC: 865144 Spans in use MALLOC: 5004 Thread heaps in use MALLOC: 8192 Tcmalloc page size ------------------------------------------------ Call ReleaseFreeMemory() to release freelist memory to the OS (via madvise()). Bytes released to the OS take up virtual address space but no physical memory. 重启后： [root@rg2-ceph001 ~]# ceph tell osd.0 heap stats osd.0 tcmalloc heap stats:------------------------------------------------ MALLOC: 4619533576 ( 4405.5 MiB) Bytes in use by application MALLOC: + 51912704 ( 49.5 MiB) Bytes in page heap freelist MALLOC: + 80496112 ( 76.8 MiB) Bytes in central cache freelist MALLOC: + 6662912 ( 6.4 MiB) Bytes in transfer cache freelist MALLOC: + 46690824 ( 44.5 MiB) Bytes in thread cache freelists MALLOC: + 50331648 ( 48.0 MiB) Bytes in malloc metadata MALLOC: ------------ MALLOC: = 4855627776 ( 4630.7 MiB) Actual memory used (physical + swap) MALLOC: + 0 ( 0.0 MiB) Bytes released to OS (aka unmapped) MALLOC: ------------ MALLOC: = 4855627776 ( 4630.7 MiB) Virtual address space used MALLOC: MALLOC: 554089 Spans in use MALLOC: 4905 Thread heaps in use MALLOC: 8192 Tcmalloc page size ------------------------------------------------

从重启 OSD.0 前后的 **heap stats** 对比可以看出，**重启后内存占用显著下降**，尤其是 **tcmalloc 缓存（freelist）和应用程序实际使用的内存**。以下是详细分析：

### ****一、重启前后关键指标对比****

| **指标** | **重启前** | **重启后** | **变化量** |
| --- | --- | --- | --- |
| **应用内存（Bytes in use）** | 4540.1 MiB | 4405.5 MiB | **减少 134.6 MiB** |
| **Page Heap Freelist** | 131.8 MiB | 49.5 MiB | **减少 82.3 MiB** |
| **Central Cache Freelist** | 1423.0 MiB | 76.8 MiB | **减少 1346.2 MiB** |
| **Thread Cache Freelists** | 1061.6 MiB | 44.5 MiB | **减少 1017.1 MiB** |
| **实际内存（Actual used）** | 7226.0 MiB (物理+Swap) | 4630.7 MiB (物理+Swap) | **减少 2595.3 MiB** |
| **虚拟内存（Virtual）** | 8316.1 MiB | 4630.7 MiB | **减少 3685.4 MiB** |

### ****二、重启优化的核心原因****

**tcmalloc 缓存释放**：

**Freelist 内存大幅减少**：

**central cache freelist** 从 1423 MiB 降至 76.8 MiB，说明重启后 tcmalloc 的中央缓存池被清空。

**thread cache freelists** 从 1061.6 MiB 降至 44.5 MiB，线程本地缓存也被释放。

**原因**：重启会强制释放 tcmalloc 保留的空闲内存（类似 **heap release\_free\_memory** 的效果）。

**应用内存减少**：

**BlueStore/RocksDB 缓存重置**：

重启后，BlueStore 的数据块缓存和 RocksDB 的 Block Cache 被清空，需重新预热。

应用内存从 4540.1 MiB 降至 4405.5 MiB，说明部分长期未释放的元数据（如 OMAP）被清理。

**虚拟内存回归正常**：

虚拟内存从 8316.1 MiB 降至 4630.7 MiB，表明重启后进程的地址空间占用更合理。

### ****三、重启后的潜在问题****

**性能短暂下降**：

重启后缓存失效，需重新加载热数据和元数据，可能导致短期内 IO 延迟增加（尤其是 RocksDB 的读性能）。

**内存可能逐步增长**：

随着 OSD 运行时间增长，tcmalloc 的 freelist 和 BlueStore/RocksDB 缓存会再次积累，最终接近重启前的水平（需定期监控）。

### ****四、长期优化建议****

**定期释放 tcmalloc 内存**（避免频繁重启）：

bash

复制

下载

*# 手动释放空闲内存（效果类似重启的缓存清理）*

ceph tell osd.0 heap release\_free\_memory

**限制 BlueStore 缓存大小**（需重启生效）：

ini

复制

下载

[osd.0]bluestore\_cache\_size = 2G # 根据物理内存调整（如 64GB 内存设为 4~8GB）

**升级 Ceph 版本**：

Jewel 10.2.10 的 tcmalloc 和 BlueStore 内存管理较粗糙，建议升级至 **Luminous (12.2.x)** 或更高版本。

**监控内存增长趋势**：

bash

复制

下载

*# 定期记录内存状态*watch -n 60 "ceph tell osd.0 heap stats | grep -A 10 'MALLOC:'"

### ****五、总结****

**重启的作用**：强制释放 tcmalloc 缓存和 BlueStore/RocksDB 内存，短期内显著降低内存占用。

**根本问题**：Jewel 版本的内存管理机制导致缓存无法自动回收，需手动干预或升级版本。

**建议操作**：

定期执行 **heap release\_free\_memory**。

限制 **bluestore\_cache\_size**。

规划升级至新版本 Ceph。