通过iostat 命令观察输出如下： sdj 0.00 0.00 1.00 25.00 0.00 0.08 6.77 0.04 1.85 16.00 1.28 1.19 3.10 sdo 0.00 752.00 1.00 280.00 0.00 4.00 29.18 0.67 2.38 0.00 2.39 0.31 8.70 sdn 0.00 0.00 1.00 0.00 0.00 0.00 8.00 0.01 10.00 10.00 0.00 10.00 1.00 intelcas1-1 0.00 0.00 40.00 159.00 0.60 1.59 22.51 2078464.04 2.42 0.20 2.98 5.04 100.20 intelcas1-2 0.00 0.00 7.00 0.00 0.03 0.00 8.00 1104311.64 0.29 0.29 0.00 143.14 100.20 intelcas1-3 0.00 0.00 8.00 0.00 0.03 0.00 8.00 3682871.97 0.00 0.00 0.00 125.25 100.20 intelcas1-4 0.00 0.00 31.00 69.00 0.24 1.07 26.96 3003900.27 0.27 0.16 0.32 10.02 100.20 intelcas1-5 0.00 0.00 11.00 44.00 0.04 0.00 1.60 3119582.66 0.40 1.55 0.11 18.22 100.20 intelcas2-1 0.00 0.00 7.00 0.00 0.03 0.00 8.00 4069497.51 0.43 0.43 0.00 143.29 100.30 intelcas2-2 0.00 0.00 10.00 98.00 0.04 9.14 174.07 2310571.54 5.90 0.30 6.47 9.29 100.30 intelcas2-3 0.00 0.00 6.00 0.00 0.02 0.00 8.00 482905.99 0.33 0.33 0.00 167.17 100.30 intelcas2-4 0.00 0.00 6.00 0.00 0.04 0.00 12.00 295925.73 1.67 1.67 0.00 167.17 100.30 intelcas2-5 0.00 0.00 11.00 0.00 0.04 0.00 8.00 822.13 0.00 0.00 0.00 91.18 100.30 可以看到缓存盘几乎跑满， 该如何解决

从 **iostat** 输出可以看出，**Intel CAS 缓存盘（如 intelcas1-1 到 intelcas2-5）的 I/O 使用率（%util）均接近 100%**，且部分设备的响应时间（**await**）较高（如 **intelcas1-2** 的 **await=143.14ms**），这表明缓存盘已成为性能瓶颈。以下是解决方案：

### ****一、问题根因分析****

**缓存盘过载表现**：

**%util=100%**：设备持续满负载运行。

**高 await（如 >50ms）**：I/O 请求排队严重，响应延迟高。

**高读写吞吐量**（如 **intelcas1-1** 的 **rkB/s=2078464**）：缓存盘承受大量数据读写。

**可能原因**：

**缓存盘性能不足**：硬件性能（如 IOPS、带宽）无法满足 Ceph 需求。

**缓存比例不合理**：缓存盘与慢速盘（如 HDD）的容量或性能配比失衡。

**数据热点**：大量请求集中在少数对象或 PG 上。

**Ceph 配置不当**：如缓存层策略、刷写阈值设置不合理。

### ****二、解决方案****

#### ****1. 升级缓存盘硬件（根本解决）****

**替换为高性能 NVMe SSD**：  
Intel CAS 缓存盘如果是 SATA SSD 或低端 NVMe，建议升级为高性能 NVMe（如 Intel Optane 或企业级 SSD）。

**增加缓存盘数量**：  
通过多盘分担负载（如从 5 盘扩展到 10 盘）。

#### ****2. 优化 Ceph 缓存层配置****

**调整缓存模式**（若使用 Cache Tiering）：  
在 **ceph osd pool set** 中优化缓存策略：

bash

复制

下载

*# 设置缓存层为 writeback（需根据场景选择）*

ceph osd tier cache-mode <cache-pool> writeback*# 调整刷写和驱逐阈值*

ceph osd pool set <cache-pool> hit\_set\_count 12

ceph osd pool set <cache-pool> hit\_set\_period 14400

ceph osd pool set <cache-pool> target\_max\_bytes 1000000000000 *# 1TB*

**禁用 Cache Tiering（若性能反而下降）**：  
Cache Tiering 在 Jewel 版本中可能存在效率问题，直接使用 BlueStore 的自动缓存管理更优：

bash

复制

下载

ceph osd tier remove <base-pool> <cache-pool>

#### ****3. 调整 BlueStore 和 RocksDB 参数****

**减少 RocksDB 对缓存盘的压力**：

ini

复制

下载

[osd]bluestore\_rocksdb\_options = "max\_background\_compactions=4,max\_background\_flushes=2"

**限制 BlueStore 刷写速率**：

ini

复制

下载

bluestore\_sync\_submit\_transaction = false # 异步提交（提高吞吐）bluestore\_prefer\_deferred\_size = 32768 # 增加延迟写入缓冲区

#### ****4. 数据分布优化****

**检查 PG 分布是否均衡**：

bash

复制

下载

ceph pg dump | grep ^[0-9] | awk '{print $1,$15}' | sort -k2 -n

若存在 PG 倾斜（如某些 PG 的 **object\_count** 远高于其他），调整 PG 数量：

bash

复制

下载

ceph osd pool set <pool> pg\_num <new\_pg\_num>

#### ****5. 监控与扩容****

**识别热点对象**：  
使用 **ceph daemon osd.<id> perf dump** 查看高负载对象。

**扩容 OSD**：  
增加 OSD 节点或更换更大容量缓存盘。

#### ****6. 内核与文件系统优化****

**调整内核 I/O 调度器**（对 NVMe 推荐 **none**）：

bash

复制

下载

echo none | sudo tee /sys/block/nvme0n1/queue/scheduler

**增大队列深度**：

bash

复制

下载

echo 1024 | sudo tee /sys/block/nvme0n1/queue/nr\_requests

### ****三、应急措施（临时缓解）****

**手动触发缓存刷写**：  
将缓存池数据强制刷写到基础池：

bash

复制

下载

ceph osd tier flush <cache-pool>

**限制客户端流量**：  
通过 QoS 限制客户端 IOPS 或带宽：

bash

复制

下载

ceph osd set-backfill-full

ceph osd set-nearfull

### ****四、验证优化效果****

**重新运行 iostat**：

bash

复制

下载

iostat -x 1 10

目标：**%util** 降至 80% 以下，**await** 降至 10ms 以内。

**检查 OSD 延迟**：

bash

复制

下载

ceph osd perf

### ****五、总结****

| **问题** | **解决方案** | **操作命令示例** |
| --- | --- | --- |
| 缓存盘硬件性能不足 | 升级为高性能 NVMe SSD | - |
| Cache Tiering 配置不合理 | 调整缓存模式或禁用 Cache Tiering | **ceph osd tier remove** |
| RocksDB 压力过大 | 减少后台压缩/刷写任务 | **bluestore\_rocksdb\_options** |
| 数据分布不均 | 调整 PG 数量或均衡分布 | **ceph pg dump** |
| 内核 I/O 调度不佳 | 切换调度器为 **none**（NVMe） | **echo none > /sys/block/nvme0n1/queue/scheduler** |

**推荐优先级**：

**硬件升级**（最快见效） → 2. **调整 Ceph 配置** → 3. **优化数据分布**。