# Ceph peering 相关参数测试

## 一、基本情况

针对近期某项目扩容，参考不同来源分享的配置，选取 3 个方面的参数进行优化：

### 1.1 内存使用（暂不作调整）

**（项目实际环境的内存使用情况显示在内存使用上并没明显问题，先不做调整）**

主要是 osdmap 在内存中备份数目。

在 ceph-mon 中，默认为 500 ，当集群规模变大，内存可能达到超过 500M～1G，下调到 300。

在 ceph-osd 中，默认值已经降低，为 50/40 ，相应调整到 30/20；

对于 OSD 数量更大的情况下，可以进一步降低，如 300 -> 100, 30 -> 20, 20 -> 10 。

配置项：

[global]

; default is 40

; max number of OSDMaps in a single message

osd\_map\_message\_max = 20

[mon]

; default is 500

mon\_osd\_cache\_size = 300

[osd]

; default is 50

osd\_map\_cache\_size = 30

; default is 40

osd\_map\_max\_advance = 20

; default is 40

osd\_map\_share\_max\_epochs = 20

; default is 40

osd\_pg\_epoch\_persisted\_max\_stale = 20

### 1.2 ceph-osd 的线程数

主要是 peering 和 osd\_op 的线程数量，提高 peering 和 osd 处理 op 的并发量。

peering 线程数默认是 2 ，单次处理事件数量为 20 ，单次处理过程为串行；

增加线程数到 16 ，降低单次处理事件数量为 1 ，增加并发处理，减少串行处理过程中阻塞。

osd\_op 的线程线默认为：

对于 hdd ，5 （分片） \* 1（倍数） = 5

对于 ssd ，8 （分片） \* 2（倍数） = 16

这里对 hhd 和 ssd 倍数都增加一倍。

配置项：

[osd]

; default is 2

osd\_peering\_wq\_threads = 16

; default is 20

osd\_peering\_wq\_batch\_size = 1

; related paramter

; osd\_op\_num\_shards

; default is 0, 5(hdd), 8(ssd)

; default is 0, 1(hdd), 2(ssd)

osd\_op\_num\_threads\_per\_shard\_hdd = 2

osd\_op\_num\_threads\_per\_shard\_ssd = 4

### 1.3 降低 scrub 的频率

scrub 会降低集群处理客户端 I/O 的能力。默认系统负载 loadavg 在 0.5 才允许 scrub ，当系统负载大于该值时，scrub 会被推迟，对于多核环境，这个值可以相对提高，以允许 scrub 在必要时能够顺利执行。

默认 scrub 的间隔在 1 ~ 7 天之间，deep\_scrub 为 7 天，分别调整为 14 ～ 28 天和 28 天。

默认 scrub 操作之间没有时间间隔，增加 0.1 s 的间隔让更多其它的操作（如客户端的 I/O 请求）能够执行。

[osd]

; default is 0.5

; scrubbing is allowed when loadavg is lower

; than this value

osd\_scrub\_load\_threshold = 5

; default is 7 days

; increase to 28 days

osd\_scrub\_max\_interval = 2419200

; default is 1 day

; increase to 14 days

osd\_scrub\_min\_interval = 1209600

; default is 7 days

osd\_deep\_scrub\_interval = 2419200

; default is 0

osd\_scrub\_sleep = 0.1

## 二、测试方案

### 2.1 测试目标

在大规模 Ceph 集群里，硬盘、ceph-osd 进程出现问题的情况会变得比较常见，而 ceph-osd 重新启动或者增加/删除 osd 都会导致 pg 之间重新 peering ，peering 会阻塞客户端的 I/O（相对的 backfill 和 recovery 不会阻塞客户端 I/O ，当然由于占用网络和硬盘 I/O 带宽，仍然会影响客户端的性能），因此本次优化主要目标放在了加快 peering 这个进程。

测试的目标正是验证前述参数的调整是否有效。

### 2.2 测试环境

在单独一台服务器，以开发模式进行测试。Ceph 集群的基本情况是：

3 ceph-mon + 8 ceph-osd

每次调整参数，都重新生成新的集群。

最早的测试中，ceph-osd 的数据都直接放在本地硬盘（以 hdd osd 指代）。但是服务器硬盘性能有限，测试数据可靠性不高，后面的测试中使用了基于内核 brd 模块的内存块设备（以 ramdisk osd 指代）。

### 2.3 测试方法

开发环境的 Ceph 集群，默认建 2 个 pool ，共 16 pg。测试的方法是：

在 Ceph 集群中所有 pg 都处于 active+clean 情况下，新建一个 1024 pg ，复本数为 3 的 pool 。

由于 ceph-osd 的数量是 8 ，新建 pool 成功之后，平均每个 osd 的 pg 数量在 400 左右，和某项目的情况类似。

观察所有新建 pg 的状态变化，记录创建 pool 到所有 pg 都处于 active+clean 状态的耗时。

## 三、 测试数据

### 3.1 8 hdd osd

#### 3.1.1 数据

|  |  |
| --- | --- |
| 配置参数 | peering 耗时 |
| 默认 | 41m |
| osd\_peering\_wq\_threads = 16 | > 50m |
| osd\_peering\_wq\_threads = 16  osd\_peering\_wq\_batch\_size = 1 | > 41m |
| osd\_peering\_wq\_threads = 16  osd\_peering\_wq\_batch\_size = 1  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 62m |

#### 3.1.2 数据说明

8个 osd 同时开始 I/O ，服务器硬盘所限，难以得到有效的数据。

### 3.2 8 ramdisk osd

#### 3.2.1 数据

|  |  |
| --- | --- |
| 配置参数 | peering 耗时 |
| 默认 | 14s |
| osd\_peering\_wq\_threads = 16  osd\_peering\_wq\_batch\_size = 1  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 15s |
| osd\_peering\_wq\_threads = 8  osd\_peering\_wq\_batch\_size = 4  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 16s |

#### 3.2.1 数据说明

可以看到，当硬盘性能非常好时，上述参数调整的效果很小。

但是默认参数的表现仍是最好的。

### 3.3 6 ramdisk od + 2 hdd osd

#### 3.3.1 数据

|  |  |
| --- | --- |
| 配置参数 | peering 耗时 |
| 默认 | 16m |
| osd\_peering\_wq\_threads = 16  osd\_peering\_wq\_batch\_size = 1  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 20m57s |
| osd\_peering\_wq\_threads = 8  osd\_peering\_wq\_batch\_size = 4  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 17m48s |
| osd\_peering\_wq\_threads = 6  osd\_peering\_wq\_batch\_size = 8  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 21m07s |
| osd\_peering\_wq\_threads = 8  osd\_peering\_wq\_batch\_size = 6  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 22m12s |
| osd\_peering\_wq\_threads = 8  osd\_peering\_wq\_batch\_size = 2  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 21m |

#### 3.3.2 数据说明

仍然是默认配置表现最好。

这里针对 osd\_peering\_wq\_threads 和 osd\_peering\_wq\_batch\_size 的不同取值的组合进行测试，发现：

* osd\_peering\_wq\_threads 的取值超过系统 cpu 核心的数量时反而性能降低，测试服务器共有 16 颗逻辑 cpu ，当两个 hdd osd 都配置为线程数量都配置为 16 时，性能反而不如配置为 8 ；
* ossd\_peering\_wq\_batch\_size 的取值比较难确定，较大时的确可能降低性能。但是默认配置是 20 ，却是性能最好的，目前没有好的解释。

### 3.4 7 ramdisk osd + 1 hdd osd

#### 3.4.1 数据

|  |  |
| --- | --- |
| 配置参数 | peering 耗时 |
| 默认 | 15m42s |
| osd\_peering\_wq\_threads = 16  osd\_peering\_wq\_batch\_size = 1  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 15m22s |
| osd\_peering\_wq\_threads = 8  osd\_peering\_wq\_batch\_size = 4  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 14m24s |
| osd\_peering\_wq\_threads = 8  osd\_peering\_wq\_batch\_size = 6  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 13m27s |
| osd\_peering\_wq\_threads = 6  osd\_peering\_wq\_batch\_size = 8  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 12m59s |
| osd\_peering\_wq\_threads = 8  osd\_peering\_wq\_batch\_size = 8  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 14m04s |
| osd\_peering\_wq\_threads = 8  osd\_peering\_wq\_batch\_size = 2  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 16m07s |

#### 3.4.2 数据说明

之所以进行这一部份测试，是因为在 6 ramdisk osd + 2 hdd osd 的测试过程仍然发现，服务器硬盘的性能仍然是瓶颈，对测试数据的解读造成困难。

只使用一个 hdd osd 就比较接近实际的部署环境了，实际部署时每个 osd 都是独占硬盘的（journal 可能共享性能更好的 ssd ）。

将测试结果和之前 6 ramdisk osd + 2 hdd osd 的测试对比，得到的结论有：

* osd\_peering\_wq\_threads 的取值也不完全需要参照 cpu 核心数据，因为取值为 8 时，性能都要好点取值为 16 时；
* osd\_peering\_wq\_batch\_size 的取值仍然难以确定，对于 osd\_peering\_wq\_threads 为 8 时，取值为 6 时表现最好。

### 3.5 7 ramdisk + 1 hdd osd （测试 filestore\_op\_threads ）

#### 3.5.1 测试说明

通过上面的测试，关于 peering 相关的参数进一步调整，选了 2 种方案：

* 优化配置1

osd\_peering\_wq\_threads = 8

osd\_peering\_wq\_batch\_size = 6

* 优化配置2

osd\_peering\_wq\_threads = 6

osd\_peering\_wq\_batch\_size = 8

但是 peering 过程的性能改善仍不明显，考虑到可能的瓶颈仍然是硬盘 I/O ，测试另一个配置项 filestore\_op\_threads 的作用。

filestore\_op\_threads 用来控制对 filestore 进行 I/O 的线程的数量。

#### 3.5.2 数据

下表对比了默认配置和上述经过调整的配置，使用不同 filestore\_op\_threads 取值时，peering 完成的耗时，每一列对应一个 filestore\_op\_threads 取值（默认值为 2）。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 默认 | 4 | 8 | 16 | 20 | 32 | 64 |
| 默认 | 15m42s | 8m01s | 4m57s | 4m02s | 3m12s | 2m14s | 2m40s |
| 优化1 | 13m27s | 9m22s | 5m46s | 4m18s | - | 3m15s | 2m20s |
| 优化2 | 12m59s | 10m38s | 5m41s | 4m02s | - | 3m | 2m19s |

#### 3.5.3 数据说明

和前面各项测试对比，可以看到 filestore\_op\_threads 的作用非常大，但是和前面得出的优化配置一起使用时，相对于默认配置， peering 性能反而可能下降，对于这一状况目前仍没有找到合理的解释。

## 四、 结论

虽然前面得出的优化配置效果不明显，但是从配置项的作用以及对 Ceph 代码的简单分析，仍然认为是值得尝试的，关于 filestore\_op\_threads 的测试情况参考”Ceph filestore\_op\_threads 测试报告”：

考虑到对于生产环境的影响，取值如下：

* filestore\_op\_threads = 6
* osd\_peering\_wq\_threads = 6
* osd\_peering\_wq\_batch\_size = 8

单独进行测试：

|  |  |
| --- | --- |
| 配置参数 | peering 耗时 |
| 默认 | 15m42s |
| filestore\_op\_threads = 6  osd\_peering\_wq\_threads = 6  osd\_peering\_wq\_batch\_size = 8  osd\_op\_num\_threads\_per\_shard\_hdd = 2  osd\_op\_num\_threads\_per\_shard\_ssd = 4 | 8m23s |

优化方案第二部份的配置项调整为：

[osd]

; default is 2

filestore\_op\_threads = 6

; default is 2

osd\_peering\_wq\_threads = 6

; default is 20

osd\_peering\_wq\_batch\_size = 8

; related paramter

; osd\_op\_num\_shards

; default is 0, 5(hdd), 8(ssd)

; default is 0, 1(hdd), 2(ssd)

osd\_op\_num\_threads\_per\_shard\_hdd = 2

osd\_op\_num\_threads\_per\_shard\_ssd = 4