**文档版本：V0.1**

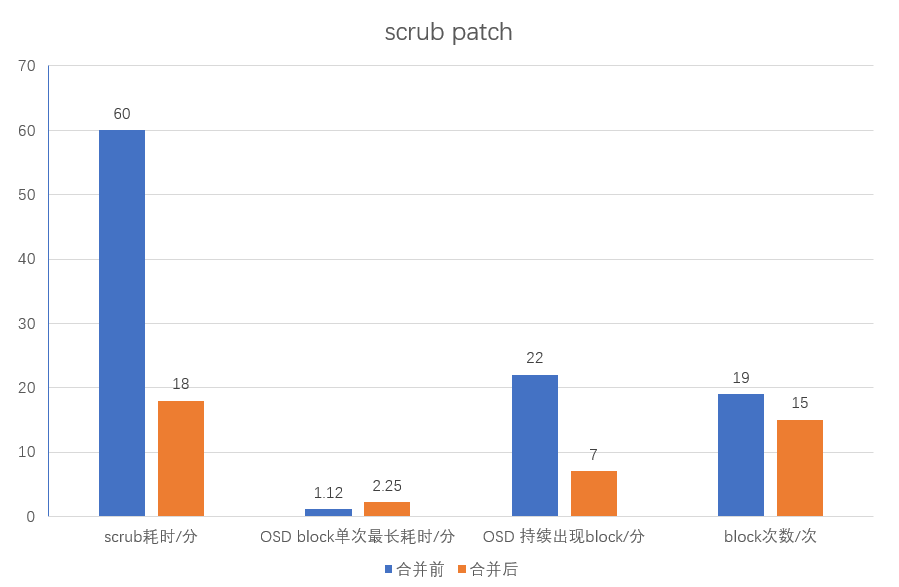
**背景**

部分生产环境使用的ceph版本为jewel-v10.2.2，目前该版本在运行时偶尔会出现slow request的现象，直接导致客户虚拟机的存储出现”kernel [sdb] abort”的日志。尤其在deep-scrub的时候，slow request更为频繁。ceph社区的两个patch优化了该问题，可以降低slow request出现的频率。下面针对这两个patch，分别解释其优化原理和测试数据。

**Deep-scrub优化**

Patch地址：[[link](https://github.com/ceph/ceph/commit/3cc29c6736007c97f58ba3a77ae149225e96d42a#diff-02e8b93bd43f8aff5a4c3ef2cd17e319)]，原先，程序在遍历PG下面的object时会递归遍历所有的子目录，大多数情况下会重复循环遍历子目录的object，该patch会在满足条件的情况下中断部分循环，减少获取PG下object信息的时间。

合并patch前后数据对比：



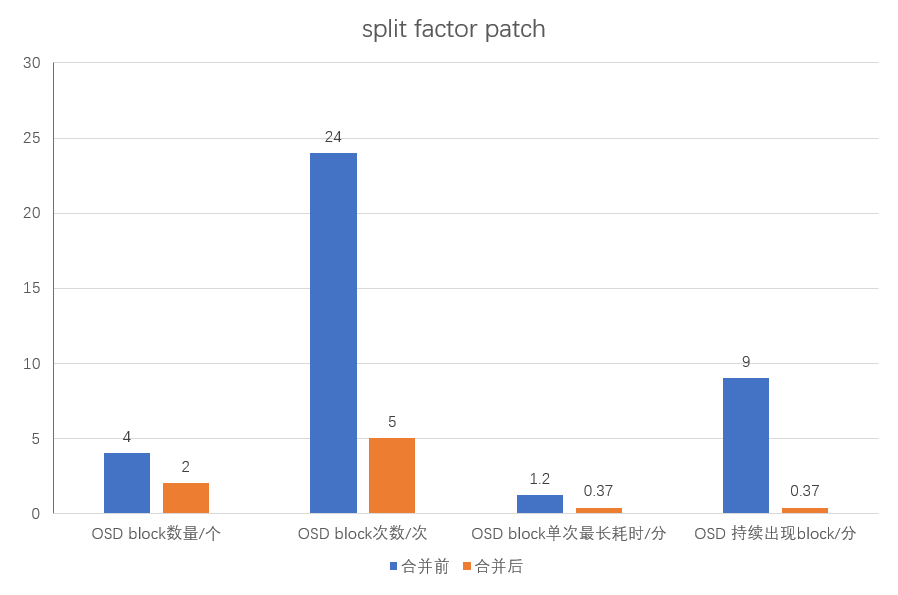
\*最明显的优化是完成一次scrub的时长

**Split-factor优化**

Patch地址：[[link](https://github.com/ceph/ceph/commit/e52ae3664a2914af4208b3adce8b5215e24626ab#diff-02e8b93bd43f8aff5a4c3ef2cd17e319)]，ceph为避免目录下面拥有太多object而导致读写性能下降，默认参数时，PG下面的子目录结构生成规则是：每当一个目录的object数量达到320个时，程序会分裂出新的子目录来存放object。但是，这样的设计会有一个问题，就是数据量越大，IOPS越高，主从PG的文件夹的同时分裂会导致读写卡顿，耗时过长而出现slow request。该patch的 优化思路是让分裂的时机变得随机，即分别设置主从PG的随机因子，让目录下文件数的上限变得不一致，避免同时分裂导致block。不过，由于随机因子过于随机，有可能主从PG的随机因子过于接近而导致优化效果有限，因此，我在测试的时候固定配置了主从PG的随机因子，刻意间隔。

由于压测前期数据量小，出现slow request的概率都比较小，所以我择取了集群写满前15分钟的数据，对比下来，这个问题没有测试解决，但是slow request的频率下降了很多

合并patch前后数据对比：



下图是每分钟出现的最多block的数量：

