Lustre性能调优-上篇

存储内核技术交流

长按识别二维码，关注我的公众号

公众号

作者	时间	QQ技术交流群
perrynzhou@gmail.com	2022/09/02	672152841

作者

- 参数查询
  - lctl list_param *.*.* — 查询lustre第3层参数,至少有3层，最多有第5层参数
  - lctl list_param *.*.* |grep ost — 查询和ost相关参数
  - lctl list_param *.*.*.* — 查询lustre第4层参数,这个可以查询大部分参数
  - lctl list_param *.*.*.* — 查询lustre第5层参数,这个可以查询大部分参数
- 服务线程调优
  - 说明 — oss/mds的IO服务线程定义，每个服务线程消耗1.5MB的内存资源
  - oss
    - 参数名称
      - {service}.threads_min — oss启动后最小的服务线程数
      - {service}.threads_max — oss启动后最大的服务线程数
      - {service}.threads_started — oss启动后已经启动的服务线程数
    - 参数查询 — lctl list_param *.*.*.* |grep threads_*|grep ost
    - 参数设置 — lctl set_param ost.OSS.ost.threads_max 65
    - 参数查看 — lctl get_param ost.OSS.ost.threads_max
  - mds
    - 参数名称
      - {service}.threads_min — oss启动后最小的服务线程数
      - {service}.threads_max — oss启动后最大的服务线程数
      - {service}.threads_started — oss启动后已经启动的服务线程数
    - 参数查询 — lctl list_param *.*.*.* |grep threads_*|grep mds
- 绑核调优
  - 说明 — mds服务线程绑定CPU核可以提高缓存命中和充分利用内存的局部性
  - 语法 — options mdt mds_num_cpts=[0]:绑定在cpu0上
options mds_rdpg_num_cpts=[0-1]:read_page服务线程绑定在cpu0~cpu1上
options lnet networks=tcp0(eth0):绑定网卡eth0到tcp0
  - 设置 — 在/etc/modprobe.d/lustre.conf文件中添加一行:options mdt mds_num_cpts=[0-6],含义是mds的线程绑定在cpu0~cpu5
- LNet调优
  - 网络传输和接送缓冲调优
    - 说明 — options ksocklnd tx_buffer_size 和rx_buffer_size都设置为0，lustre会自动调整缓冲区大小，这也能够得到最佳的性能
    - 语法 — options ksocklnd tx_buffer_size=0 rx_buffer_size=0
    - 设置 — 在/etc/modprobe.d/lustre.conf文件中append一行:options ksocklnd tx_buffer_size=0 rx_buffer_size=0
  - LNet服务绑定CPU
    - 说明 — LNet服务绑定到一个或者多个CPU上，所有的消息处理线程都会在这些CPU上，提供处理效率
    - 语法 — options lnet networks=tcp0(enp0s5)
options tcp0(enp0s5)[0]/options tcp0(enp0s5)[0,1]：前者绑定在cpu0上，后者绑定到cpu0和cpu1上
    - 设置 — 在/etc/modprobe.d/lustre.conf文件中append一行:options tcp0(enp0s5)[0,1]
  - LNet包分发和处理
    - 说明 — 默认情况LNet分发消息到CPU核处理根据nid的哈希，存在所有消息可能被同一个cpu核处理，这样就降低了处理消息效率，可以设置消息分发和处理按照round-robin方式提高消息处理效率
    - 参数

```
/******取值说明******/
// OFF-禁用round-robin模式
// ON-开启RR模式
// RR_RT-开启路由消息的RR模式
// HASH_RT-按照源端NID哈希分发消息
portal_rotor
```

    - 设置

```
$ lctl get_param portal_rotor
portal_rotor=
{
        portals: all
        rotor: ON
        description: round-robin dispatch all PUT messages
for wildcard portals
}
```

    - 参数查看

```
$ lctl get_param portal_rotor
portal_rotor=
{
        portals: all
        rotor: HASH_RT
        description: dispatch routed PUT message by hashing
source NID for wildcard portals
}
```

- 大IO优化
  - 说明
    - 后端zfs-osd/ldiskfs-osd之间的的IO最大不能超过obdfilter.{fsname}-OST000{index}.brw_size.客户端的参数osc.*.max_pages_per_rpc控制客户端RPC大小，这个参数永远小于等于obdfilter.{fsname}-OST000{index}.brw_size
    - 当客户端连接到OST,客户端会读取brw_size然后设置自身的max_pages_per_rpc的RPC大小
  - 参数
    - osd端

```
$ lctl list_param *.*.* |grep brw_size
obdfilter.bigfs-OST0001.brw_size
obdfilter.bigfs-OST0002.brw_size

// 这里默认是4M
$ lctl get_param obdfilter.bigfs-OST0001.brw_size
obdfilter.bigfs-OST0001.brw_size=4
```

    - client端

```
// 查看mdc的max_pages_per_rpc，这里参数单位是字节，每个page默认是4k, 256*4*1024=1M
$ lctl get_param mdc.bigfs-MDT0000-mdc-ffff995143192000.max_pages_per_rpc
mdc.bigfs-MDT0000-mdc-ffff995143192000.max_pages_per_rpc=256
// 这里客户端请求OST的rpc大小是 1024*4*1024 = 4M，和后端brw_size保持一致
$ lctl get_param osc.bigfs-OST0001-osc-ffff995143192000.max_pages_per_rpc
osc.bigfs-OST0001-osc-ffff995143192000.max_pages_per_rpc=1024
```

  - 设置
    - osd端

```
// lctl set_param 添加参数-P 会持久化这个参数配置
$ lctl set_param -P  obdfilter.bigfs-OST0001.brw_size=16M
$ lctl set_param -P  obdfilter.bigfs-OST0002.brw_size=16M
// 设置后生效，但是针对新连接到的客户端生效，已经连接客户端需要重新设置
$ lctl get_param obdfilter.bigfs-OST0001.brw_size
obdfilter.bigfs-OST0001.brw_size=16
```

    - client端

```
$ lctl get_param osc.bigfs-OST0001-osc-ffff9951567b9000.max_pages_per_rpc
osc.bigfs-OST0001-osc-ffff9951567b9000.max_pages_per_rpc=4096
```