

基于SPDK的CurveBS PFS存储引擎

Why

- 为了减少使用cpu做内存copy，减少系统调用
- 发挥某些被操作系统屏蔽的功能，例如nvme write zero
- 根据阿里《When Cloud Storage Meets RDMA》的说法
- 在100Gbps网络带宽时，内存带宽成为瓶颈
- Intel Memory Latency Checker (MLC)测试得到的CPU内存带宽是61Gbps

RDMA可以减轻CPU负担

- 可以减少CPU操作网络通讯的开销
- 读写内存都由网卡进行offload
- 应用程序不再通过系统调用在内核和用户态来回切换

磁盘的读写

- 基于EXT4的存储引擎，依然需要通过系统调用来回切换
- 读写都需要CPU拷贝数据
- 不能发挥某些NVME的功能，例如write zero

为什么用PFS

- 对代码比较熟悉
- 找一个能管理裸盘，具有产品级可靠性的代码挺难的
- PFS支持类POSIX文件的接口，与使用EXT4的存储引擎代码很像，所以容易移植现有代码到PFS存储引擎
- CurveBS对文件系统元数据的操作非常少，对文件系统的要求不高，所以不需要元数据高性能，这方面PFS也合适

对PFS的修改

- 基于阿里开源的PFS
- 不再基于daemon模式，而是直接使用pfs core api
- 依然向外提供管理工具，例如 pfs ls、cp、rm等
- 增加spdk驱动

新增PFS接口

- 增加pfs_pwritev和pfs_preadv接口
- `ssize_t pfs_preadv(int fd, const struct iovec *iov, int iovcnt, off_t offset);`
- `ssize_t pfs_pwritev(int fd, const struct iovec *iov, int iovcnt, off_t offset);`
- IO vector的接口主要是为了与brpc的iobuf对接，iobuf由若干地址不连续的block组成，一次IO提交可以提高效率。

PFS+SPDK 的部分读写的实现

- 某些盘只支持4k单位读写，但是CurveBS支持512字节读写
- 可能存在部分写的并发冲突
- 引入并发的range lock解决冲突

PFS+SPDK 的DMA支持

- `ssize_t pfs_writev_dma(int fd, const struct iovec *iov, int iovcnt, off_t offset)`
- `ssize_t pfs_preadv_dma(int fd, const struct iovec *iov, int iovcnt, off_t offset);`
- 直接DMA读写， 要求的内存必须是DPDK的hugetlb内存
- 必须符合NVME 内存读写地址对齐要求
- `offset 512对齐`
- 为零copy提供接口

BRPC IOBuf DMA

- 修改BRPC，允许使用dpdk内存作为IOBuf的内存分配器
- BRPC接收到的数据在IOBuf中，IOBuf直接使用于NVME DMA传输
- 使用IOBuf内存读nvme，避免自己写PRP页面对齐内存分配代码

pfs_pwrite_zero

- .在初始化curvebs时，需要创建chunk pool, 每一个chunk都要填零
- .chunk不再被卷使用时，需要回归chunk pool，为了安全也需要填0。
- .使用nvme的时候，可以直接使用nvme write zero命令，不需要传递大块数据（全是0），减少了nvme传输带宽，而且nvme在垃圾回收上可以优化，例如只是标记某块为0, 而不用实际写，gc时不需要搬运。

nvme读写的内存对齐要求

- .NVME读写传输描述分两种规格：PRP和SGL。 PRP是第一个版本， SGL是后面才发展起来的
- .PRP要求内存按PAGE对齐
- .SGL要求字节/或4字节对齐(double word) , 相对宽松

PFS NVME读对齐实现

- 内存分配页面对齐，实现基于PRP严格的规定，这样SGL也可以用
- 第一个页面可以从非0的页内位置开始直到页面结束位置，必须是512字节倍数。第二个页面必须是整页，内存位置必须在页内位置0处。最后一个页面，开始位置也在页面0处，但可以是不完整的一个页面。



- 如
- 如果以iovec来表示： [page0+3584, page0+4096) , [page1, page1+4096) , [page2, page2+4096) , [page3, page3+512)

IOPortal实现读对齐支持

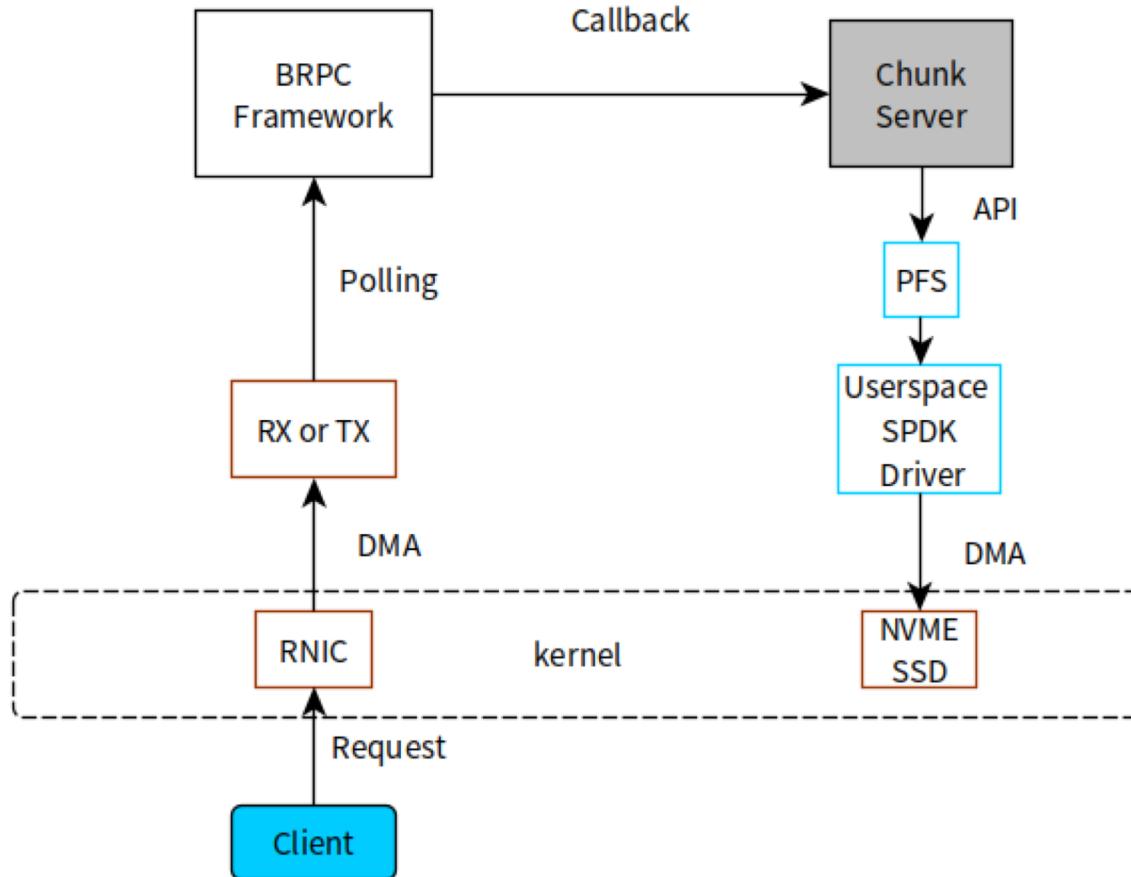
`.ssize_t IOPortal::pappend_from_dev_descriptor(int fd, off_t offset, size_t max_count)`

- 这个函数的实现是按PRP的规定来做的
- IOPortal就是IOBuf，是BRPC存放数据的类
- CurveBS 使用brpc::Controller 的attachment 发送数据
- attachment是一个IOBuf
- IOBuf直接通过rdma发送出去
- 以上过程在读chunk的代码路径上实现了零copy。

写入nvme的零copy实现

- 当前只支持到SGL，对于PRP的实现不支持。
- 对于只支持PRP的nvme，可以开启读数据零copy功能
- 对于支持SGL的nvme，可以同时开启读、写零copy功能
- RDMA模块当前还未完成修改，需要修改ucx内存分配，使用dpdk内存，才可以完成DMA写NVME

PFS DMA 总体架构



TCP也可以部分零copy

- 读写盘的部分是零copy的
- 网络部分依赖内核tcp，不是零copy

进展

- 还在测试CurveBS
- 布置、监控等工具需要更新

性能测试

- 使用pfs daemon测试
- 估计非daemon模式的会更快一点，因为没有跨进程开销

Write,DMA write,Write-zero测试

```
TELEMETRY: No legacy callbacks, legacy socket not created
[PFS_LOG] Oct  8 16:53:13.249156 INF [161786] spdk rpc address:/var/tmp/spdk.sock
[2022-10-08 16:53:13.249594] accel_engine.c: 969:sw_accel_engine_init: *NOTICE*: Accel framework software eng
[PFS_LOG] Oct  8 16:53:13.600975 INF [161786] Found devices:
[PFS_LOG] Oct  8 16:53:13.601066 INF [161786]     Name: 0000:3e:00.0n1, Size: 1920383410176, BlockSize: 512, Ph
Local CPUs: 0-13,28-41
Setup successful
initializing...
initialized.
testing pfs_write
pfs_write elapsed time: 16.6655s, used cpu time:1.32149s
testing pfs_write_dma
pfs_write_dma elapsed time: 16.6651s, used cpu time:1.15668s
testing pfs_write_zero
pfs_write_zero elapsed time: 13.099s, used cpu time:1.14355s
```

fio 4k 1个并发+单深度

```
seqwrite: (g=0): rw=randwrite, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=pfs, iodepth=1
fio-3.29-36-g607a-dirty
Starting 1 process
Jobs: 1 (f=1): [w(1)][100.0%][w=56.6MiB/s][w=14.5k IOPS][eta 00m:00s]
seqwrite: (groupid=0, jobs=1): err= 0: pid=164218: Sat Oct  8 17:00:27 2022
    write: IOPS=14.5k, BW=56.7MiB/s (59.4MB/s)(5120MiB/90356msec); 0 zone resets
        clat (usec): min=36, max=3178, avg=68.44, stdev= 3.92
        lat (usec): min=36, max=3178, avg=68.51, stdev= 3.93
        clat percentiles (usec):
|  1.00th=[   68],  5.00th=[   69], 10.00th=[   69], 20.00th=[   69],
| 30.00th=[   69], 40.00th=[   69], 50.00th=[   69], 60.00th=[   69],
| 70.00th=[   69], 80.00th=[   69], 90.00th=[   70], 95.00th=[   70],
| 99.00th=[   84], 99.50th=[   88], 99.90th=[   93], 99.95th=[   95],
| 99.99th=[  108]
        bw ( KiB/s): min=57640, max=59984, per=100.00%, avg=58028.40, stdev=163.28, samples=180
        iops       : min=14410, max=14996, avg=14507.10, stdev=40.82, samples=180
        lat (usec)  : 50=0.10%, 100=99.88%, 250=0.01%, 500=0.01%, 750=0.01%
        lat (usec)  : 1000=0.01%
        lat (msec)   : 4=0.01%
        cpu         : usr=5.22%, sys=11.41%, ctx=1310782, majf=0, minf=530
        IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
                      submit      : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
                      complete    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
        issued rwt: total=0,1310720,0,0 short=0,0,0,0 dropped=0,0,0,0
        latency     : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
    WRITE: bw=56.7MiB/s (59.4MB/s), 56.7MiB/s-56.7MiB/s (59.4MB/s-59.4MB/s), io=5120MiB (5369MB), run=90356-90356
```

Fio 4k 16并发单深度

```
...
fio-3.29-36-g607a-dirty
Starting 16 processes
Jobs: 16 (f=16): [w(16)][100.0%][w=818MiB/s][w=209k IOPS][eta 00m:00s]
seqwrite: (groupid=0, jobs=16): err= 0: pid=166239: Sat Oct  8 17:05:59 2022
  write: IOPS=210k, BW=820MiB/s (860MB/s)(80.0GiB/99938msec); 0 zone resets
    clat (usec): min=35, max=2565, avg=75.52, stdev= 7.33
      lat (usec): min=35, max=2566, avg=75.59, stdev= 7.33
    clat percentiles (usec):
    | 1.00th=[   61], 5.00th=[   67], 10.00th=[   69], 20.00th=[   72],
    | 30.00th=[   74], 40.00th=[   75], 50.00th=[   76], 60.00th=[   77],
    | 70.00th=[   78], 80.00th=[   80], 90.00th=[   84], 95.00th=[   88],
    | 99.00th=[   98], 99.50th=[  102], 99.90th=[  114], 99.95th=[  121],
    | 99.99th=[  149]
    bw ( KiB/s): min=834272, max=845544, per=100.00%, avg=839998.79, stdev=171.23, samples=3184
    iops       : min=208568, max=211386, avg=209999.68, stdev=42.81, samples=3184
    lat (usec)  : 50=0.03%, 100=99.30%, 250=0.67%, 500=0.01%, 750=0.01%
    lat (usec)  : 1000=0.01%
    lat (msec)   : 2=0.01%, 4=0.01%
    cpu         : usr=8.37%, sys=8.79%, ctx=21429828, majf=0, minf=12942
    IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
      submit     : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
      complete   : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
    issued rwt: total=0,20971520,0,0 short=0,0,0,0 dropped=0,0,0,0
    latency     : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
 WRITE: bw=820MiB/s (860MB/s), 820MiB/s-820MiB/s (860MB/s-860MB/s), io=80.0GiB (85.9GB), run=99938-99938msec
```

谢谢 !



Curve 小助手



扫一扫上面的二维码图案，加我为朋友