curvefs 2.4.0版本总体测试

一、遗留问题列表

问题列表

	风险项	ISSUE.No	负责人	严重级别	是否解决	是否需要回归	回归人	是否回归通过	应急预案	备注

二、测试内容和结论概述

测试节点硬件配置与软件版本

环境信息	稳定性测试环境 9个机器

CPU	Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
内存	256G
网卡	1. Intel Corporation I350 Gigabit Network Connection 2. Intel Corporation 82599EB 10-Gigabit SFI/SFP+
操作系统	发行版：Debian GNU/Linux 9
内核	4.19.87-netease6-1 #2 SMP Mon Sep 7 07:50:31
用途	计算节点
curvefs版本	2.1.0

部署方式

s3	nos
镜像	harbor.cloud.netease.com/curve/curvefs:citest
disk cache	INTEL SSDSC2BB80 800G
metaserver数据	ssd 混合部署
mds	ssd 混合部署
etcd	ssd 混合部署
curveadm版本	0.1.0

三、测试要点

1、warmup相关功能、异常、性能测试

2、cto相关问题修复

3、copysets数据均衡性

4、新版本sdk稳定性和性能

四、测试结论

五、详细测试数据及监控数据

5.1 常规测试

5.1.1 文件系统POSIX 接口

5.1.1.1 pjdtest

已在ci中测试。

5.1.1.2 ltp-fsstress

测试程序：ltp-full-20220930.tar.bz2

clinet.yaml

```
kind: curvefs

s3.ak: minioadmin

s3.sk: minioadmin

s3.endpoint: 10.182.30.15:9002

s3.bucket_name: chengyi01

container_image: harbor.cloud.netease.com/curve/curvefs:release-2.4-3e38c86

mdsOpt.rpcRetryOpt.addrs: 10.182.2.46:7811,10.182.2.47:7811,10.182.2.48:7811

log_dir: /home/nbs/chengyi01/fs/logs/client

data_dir: /data/chengyi-cache

client.loglevel: 0
```

测试步骤：

```
set -ex

mkdir -p fsstress
pushd fsstress
wget -q -O ltp-full.tgz http://59.111.93.102:8080/qa/ltp-full.tgz //
tar xzf ltp-full.tgz
pushd ltp-full-20091231/testcases/kernel/fs/fsstress
make
BIN=$(readlink -f fsstress)
popd
popd
T=$(mktemp -d -p .)
"$BIN" -d "$T" -l 1 -n 1000 -p 10 -v
echo $?
rm -rf -- "$T"
```

测试结果：

success

5.1.2 元数据项& 数据属性

5.1.2.1 dbench

dbench

执行命令：

```
sudo dbench -t 600 -D ltp-full-20220930 -c /usr/share/dbench/client.txt 10
```

结果：

```
Operation      Count    AvgLat    MaxLat
 ----------------------------------------
 NTCreateX     105740    11.108  1149.330
 Close          77763    11.617  1326.275
 Rename          4473   500.666  4445.447
 Unlink         21257    10.667  1143.985
 Qpathinfo      95954     5.922  1405.727
 Qfileinfo      16758     0.019     0.138
 Qfsinfo        17514     1.216   740.514
 Sfileinfo       8567     5.179   684.553
 Find           37026    13.418  1599.009
 WriteX         52194     0.079     4.078
 ReadX         166120     0.196   628.751
 LockX            348     0.005     0.015
 UnlockX          348     0.002     0.017
 Flush           7349    39.296  1008.781

Throughput 5.50159 MB/sec  10 clients  10 procs  max_latency=4445.454 ms
```

5.1.2.2 iozone

测试步骤：

```
iozone -a -n 1g -g 4g  -i 0 -i 1 -i 2 -i 3 -i 4 -i 5 -i 8 -f  testdir -Rb log.xls
iozone -c -e -s 1024M -r 16K -t 1 -F testfile -i 0 -i 1
iozone -c -e -s 1024M -r 1M -t 1 -F testfile -i 0 -i 1
iozone -c -e -s 10240M -r 1M -t 1 -F testfile -i 0 -i 1
```

测试结果：

`iozone -a -n 1g -g 4g  -i 0 -i 1 -i 2 -i 3 -i 4 -i 5 -i 8 -f  testdir -Rb log.xls`

log.xls

`iozone -c -e -s 1024M -r 16K -t 1 -F testfile -i 0 -i 1`

```
Include close in write timing
        Include fsync in write timing
        File size set to 1048576 kB
        Record Size 16 kB
        Command line used: iozone -c -e -s 1024M -r 16K -t 1 -F testfile -i 0 -i 1
        Output is in kBytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 kBytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
        Throughput test with 1 process
        Each process writes a 1048576 kByte file in 16 kByte records

        Children see throughput for  1 initial writers  =   87558.38 kB/sec
        Parent sees throughput for  1 initial writers   =   87554.11 kB/sec
        Min throughput per process                      =   87558.38 kB/sec
        Max throughput per process                      =   87558.38 kB/sec
        Avg throughput per process                      =   87558.38 kB/sec
        Min xfer                                        = 1048576.00 kB

        Children see throughput for  1 rewriters        =   85795.09 kB/sec
        Parent sees throughput for  1 rewriters         =   85790.68 kB/sec
        Min throughput per process                      =   85795.09 kB/sec
        Max throughput per process                      =   85795.09 kB/sec
        Avg throughput per process                      =   85795.09 kB/sec
        Min xfer                                        = 1048576.00 kB

        Children see throughput for  1 readers          =  320914.25 kB/sec
        Parent sees throughput for  1 readers           =  320818.59 kB/sec
        Min throughput per process                      =  320914.25 kB/sec
        Max throughput per process                      =  320914.25 kB/sec
        Avg throughput per process                      =  320914.25 kB/sec
        Min xfer                                        = 1048576.00 kB

        Children see throughput for 1 re-readers        =  983651.06 kB/sec
        Parent sees throughput for 1 re-readers         =  983069.33 kB/sec
        Min throughput per process                      =  983651.06 kB/sec
        Max throughput per process                      =  983651.06 kB/sec
        Avg throughput per process                      =  983651.06 kB/sec
        Min xfer                                        = 1048576.00 kB
```

```
iozone -c -e -s 1024M -r 1M -t 1 -F testfile -i 0 -i 1
```

```
Include close in write timing
        Include fsync in write timing
        File size set to 1048576 kB
        Record Size 1024 kB
        Command line used: iozone -c -e -s 1024M -r 1M -t 1 -F testfile -i 0 -i 1
        Output is in kBytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 kBytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
        Throughput test with 1 process
        Each process writes a 1048576 kByte file in 1024 kByte records

        Children see throughput for  1 initial writers  =   96138.94 kB/sec
        Parent sees throughput for  1 initial writers   =   96132.17 kB/sec
        Min throughput per process                      =   96138.94 kB/sec
        Max throughput per process                      =   96138.94 kB/sec
        Avg throughput per process                      =   96138.94 kB/sec
        Min xfer                                        = 1048576.00 kB

        Children see throughput for  1 rewriters        =   99591.31 kB/sec
        Parent sees throughput for  1 rewriters         =   99584.72 kB/sec
        Min throughput per process                      =   99591.31 kB/sec
        Max throughput per process                      =   99591.31 kB/sec
        Avg throughput per process                      =   99591.31 kB/sec
        Min xfer                                        = 1048576.00 kB

        Children see throughput for  1 readers          =  326366.78 kB/sec
        Parent sees throughput for  1 readers           =  326293.75 kB/sec
        Min throughput per process                      =  326366.78 kB/sec
```

```
Max throughput per process                      =  326366.78 kB/sec
Avg throughput per process                      =  326366.78 kB/sec
Min xfer                                        = 1048576.00 kB

Children see throughput for 1 re-readers        = 1567921.62 kB/sec
Parent sees throughput for 1 re-readers         = 1566385.40 kB/sec
Min throughput per process                      = 1567921.62 kB/sec
Max throughput per process                      = 1567921.62 kB/sec
Avg throughput per process                      = 1567921.62 kB/sec
```

```
        Min xfer                                        = 1048576.00 kB
```

iozone -c -e -s 10240M -r 1M -t 1 -F testfile -i 0 -i 1

```
         Include close in write timing
        Include fsync in write timing
        File size set to 10485760 kB
        Record Size 1024 kB
        Command line used: iozone -c -e -s 10240M -r 1M -t 1 -F testfile -i 0 -i 1
        Output is in kBytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 kBytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
        Throughput test with 1 process
        Each process writes a 10485760 kByte file in 1024 kByte records

        Children see throughput for  1 initial writers  =   97156.06 kB/sec
        Parent sees throughput for  1 initial writers   =   97155.48 kB/sec
        Min throughput per process                      =   97156.06 kB/sec
        Max throughput per process                      =   97156.06 kB/sec
        Avg throughput per process                      =   97156.06 kB/sec
        Min xfer                                        = 10485760.00 kB

        Children see throughput for  1 rewriters        =   96146.71 kB/sec
        Parent sees throughput for  1 rewriters         =   96145.96 kB/sec
        Min throughput per process                      =   96146.71 kB/sec
        Max throughput per process                      =   96146.71 kB/sec
        Avg throughput per process                      =   96146.71 kB/sec
        Min xfer                                        = 10485760.00 kB

        Children see throughput for  1 readers          =  341906.62 kB/sec
        Parent sees throughput for  1 readers           =  341898.45 kB/sec
        Min throughput per process                      =  341906.62 kB/sec
        Max throughput per process                      =  341906.62 kB/sec
```

```
Avg throughput per process                      =  341906.62 kB/sec
Min xfer                                        = 10485760.00 kB

Children see throughput for 1 re-readers        = 1456958.62 kB/sec
Parent sees throughput for 1 re-readers         = 1456786.91 kB/sec
Min throughput per process                      = 1456958.62 kB/sec
Max throughput per process                      = 1456958.62 kB/sec
Avg throughput per process                      = 1456958.62 kB/sec
```

```
          Min xfer                                        = 10485760.00 kB
```

5.1.2.3 mdtest

测试步骤：

```
#
for i in 4 8 16;do mpirun --allow-run-as-root -np $i  mdtest -z 2 -b 3 -I 10000 -d 
/home/chengyi01/test/mnt/mnt-1/mdtest;done

#
mpirun --allow-run-as-root -np 16  mdtest -C -F -L -z 4 -b 10 -I 10000 -d mdtest -w 1024
```

测试结果：

`for i in 4 8 16;do mpirun --allow-run-as-root -np $i  mdtest -z 2 -b 3 -I 10000 -d /home/chengyi01/test/mnt/mnt-1/mdtest;done`

```
-- started at 12/16/2022 17:37:53 --

mdtest-1.9.3 was launched with 4 total task(s) on 1 node(s)
Command line used: mdtest -z 2 -b 3 -I 10000 -d /home/chengyi01/test/mnt/mnt-1/mdtest
Path: /home/chengyi01/test/mnt/mnt-1
FS: 10240.0 TiB   Used FS: 0.0%   Inodes: 1024.0 Mi   Used Inodes: 0.0%

4 tasks, 520000 files/directories



SUMMARY: (of 1 iterations)
   Operation                      Max            Min           Mean        Std Dev
   ---------                      ---            ---           ----        -------
   Directory creation:        659.362        659.362        659.362          0.000
   Directory stat    :        640.632        640.632        640.632          0.000
   Directory removal :        376.895        376.895        376.895          0.000
   File creation     :        573.503        573.503        573.503          0.000
   File stat         :        984.815        984.815        984.815          0.000
   File read         :        916.783        916.783        916.783          0.000
   File removal      :        439.710        439.710        439.710          0.000
   Tree creation     :        313.125        313.125        313.125          0.000
   Tree removal      :         12.875         12.875         12.875          0.000

-- finished at 12/16/2022 19:20:39 --
-- started at 12/16/2022 19:20:39 --

mdtest-1.9.3 was launched with 8 total task(s) on 1 node(s)
Command line used: mdtest -z 2 -b 3 -I 10000 -d /home/chengyi01/test/mnt/mnt-1/mdtest
Path: /home/chengyi01/test/mnt/mnt-1
FS: 10240.0 TiB   Used FS: 0.0%   Inodes: 1024.0 Mi   Used Inodes: 0.0%

8 tasks, 1040000 files/directories

SUMMARY: (of 1 iterations)
   Operation                      Max            Min           Mean        Std Dev
   ---------                      ---            ---           ----        -------
   Directory creation:        676.693        676.693        676.693          0.000
   Directory stat    :       1673.279       1673.279       1673.279          0.000
   Directory removal :        436.321        436.321        436.321          0.000
   File creation     :        597.750        597.750        597.750          0.000
   File stat         :       1698.615       1698.615       1698.615          0.000
   File read         :       1622.404       1622.404       1622.404          0.000
   File removal      :        494.133        494.133        494.133          0.000
   Tree creation     :        310.338        310.338        310.338          0.000
   Tree removal      :          6.949          6.949          6.949          0.000

-- finished at 12/16/2022 22:01:21 --
 -- started at 12/16/2022 22:01:21 --

mdtest-1.9.3 was launched with 16 total task(s) on 1 node(s)
Command line used: mdtest -z 2 -b 3 -I 10000 -d /home/chengyi01/test/mnt/mnt-1/mdtest
Path: /home/chengyi01/test/mnt/mnt-1
FS: 10240.0 TiB   Used FS: 0.0%   Inodes: 1024.0 Mi   Used Inodes: 0.0%

16 tasks, 2080000 files/directories
```

```
SUMMARY: (of 1 iterations)
   Operation                      Max            Min           Mean        Std Dev
   ---------                      ---            ---           ----        -------
   Directory creation:        782.754        782.754        782.754          0.000
   Directory stat    :       2277.546       2277.546       2277.546          0.000
   Directory removal :        481.781        481.781        481.781          0.000
   File creation     :        662.049        662.049        662.049          0.000
   File stat         :       2276.689       2276.689       2276.689          0.000
   File read         :       2164.328       2164.328       2164.328          0.000
   File removal      :        544.474        544.474        544.474          0.000
   Tree creation     :        315.641        315.641        315.641          0.000
   Tree removal      :          3.526          3.526          3.526          0.000
```

```
-- finished at 12/17/2022 02:40:09 --
```

5.1.2.4 rename 测试用例集

暂无

5.1.2.5 xfstest

测试步骤：

```
#!/bin/sh -x

set -e

wget http://59.111.93.102:8080/qa/fsync-tester.c
gcc -D_GNU_SOURCE fsync-tester.c -o fsync-tester

./fsync-tester

echo $PATH
whereis lsof
lsof
```

测试结果：

success

5.1.3 数据一致性测试

5.1.3.1 编译项目或者内核

测试步骤：

```
# linux
#!/usr/bin/env bash

set -e

wget -O linux.tar.gz https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.4.tar.xz
sudo apt-get install libelf-dev bc -y
mkdir t
cd t
tar xzf ../linux.tar.gz
cd linux*
make defconfig
make -j`grep -c processor /proc/cpuinfo`
cd ..
if ! rm -rv linux* ; then
    echo "uh oh rm -r failed, it left behind:"
    find .
    exit 1
fi
cd ..
rm -rv t linux*
```

测试结果：

success

5.1.3.2 vdbench读写一致性测试

测试步骤：

```
fsd=fsd1,anchor=/home/nbs/failover/test1,depth=1,width=10,files=10,sizes=(100m,0),shared=yes,openflags=o_direct
fwd=fwd1,fsd=fsd1,threads=10,xfersize=(512,20,4k,20,64k,20,512k,20,1024k,20),fileio=random,fileselect=random,rdpct=50
rd=rd1,fwd=fwd*,fwdrate=max,format=restart,elapsed=10000,interval=1

exec :  ./vdbench -f profile -jn
```

测试结果：

totals.html

5.2 异常测试

操作	影响
kill mds 后重启	无影响
kill etcd 后重启	无影响
kill metaserver 后重启	无影响
主etcd掉电	etcd数据如果在系统盘，数据会丢失

主mds掉电	无影响
一个metasever掉电	概率性 io卡顿 0-10s，需要配置数据盘自动挂载
1个etcd\mds\metaserver 网络拔出	无影响
client 网络拔出	网络恢复后恢复
网络延时300ms	性能下降
丢包30%	性能卡顿
丢包10%	性能卡顿
client节点丢包	性能卡顿
metaserver 数据迁出	正常
增加metaserver数据迁入	新增metaserver上无copysets。迁入需要等原故障的metaserver迁出

5.3 新增功能测试

5.3.1 warmup测试

缓存盘 1.5TiB,

5.3.1.1 cto open

5.3.1.1.1 静态warmup

参考 http://eq.hz.netease.com//#/useCaseManag/list?projectId=1155&moduleid=9870838 中的 fs文件系统/2.4.0版本自测用例/预热数据

5.3.1.1.2 同时有读写时warmup

5.3.1.1.2.1 缓存盘容量不足时

调整缓存盘的容量（fdisk），制造缓存盘容量不足。

缓存盘大小保持一致。

5.3.1.1.2.1.1 大文件（根据缓存盘容量）并发操作

操作	结论
挂卸载 fuse	md5一致
其他文件并发读写	md5一致
单metaserver异常（kill）	md5一致
多挂载，不共用缓存盘，并发warmup同一文件	md5一致
多挂载，共用缓存盘，并发warmup同一文件	md5一致
多挂载，共用缓存盘，并发warmup不同文件	md5一致

5.3.1.1.2.1.2 大规模目录（1000w+）

操作	结论
挂卸载 fuse	缓存盘内s3对象文件数量不一致
其他目录并发读写	缓存盘内s3对象文件数量不一致
单metaserver异常（kill）	缓存盘内s3对象文件数量不一致
多挂载，不共用缓存盘，并发warmup同一目录	缓存盘内s3对象文件数量不一致
多挂载，共用缓存盘，并发warmup同一目录	缓存盘内s3对象文件数量不一致
多挂载，共用缓存盘，并发warmup不同目录	缓存盘内s3对象文件数量不一致

5.3.1.1.2.2 缓存盘容量足够时

5.3.1.1.2.2.1 大文件（根据缓存盘容量）并发操作

操作	结论
挂卸载 fuse	md5一致
其他文件并发读写	md5一致
单metaserver异常（kill）	md5一致
多挂载，不共用缓存盘，并发warmup同一文件	md5一致
多挂载，共用缓存盘，并发warmup同一文件	md5一致

多挂载，共用缓存盘，并发warmup不同文件	md5一致

5.3.1.1.2.2.2　大规模目录（1000w+）

操作	结论
挂卸载 fuse	缓存盘内s3对象文件数量一致
其他目录并发读写	缓存盘内s3对象文件数量一致
单metaserver异常（kill）	缓存盘内s3对象文件数量一致
多挂载，不共用缓存盘，并发warmup同一目录	缓存盘内s3对象文件数量一致
多挂载，共用缓存盘，并发warmup同一目录	缓存盘内s3对象文件数量一致
多挂载，共用缓存盘，并发warmup不同目录	缓存盘内s3对象文件数量一致

5.4 回归测试

- https://github.com/opencurve/curve/issues/1833
- https://github.com/opencurve/curve/issues/1841
- https://github.com/opencurve/curve/issues/1842
- https://github.com/opencurve/curve/issues/1881
- https://github.com/opencurve/curve/issues/1851
- https://github.com/opencurve/curve/issues/1854

5.5 2.3升级2.4 测试

1. 集群先升级2.4
2. client再测试升级2.4