
curvefs 2.4.0版本总体测试

- 一、遗留问题列表
- 二、测试内容和结论概述
- 三、测试要点
- 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
 - 5.1.1.2 ltp-fsstress
 - 测试结果:
 - 5.1.2 元数据项& 数据属性
 - 5.1.2.1 dbench
 - 5.1.2.2 iozone
 - 5.1.2.3 mdtest
 - 5.1.2.4 rename 测试用例集
 - 5.1.2.5 xfstest
 - 5.1.3 数据一致性测试
 - 5.1.3.1 编译项目或者内核
 - 5.1.3.2 vdbench读写一致性测试
 - 5.2 异常测试
 - 5.3 新增功能测试
 - 5.3.1 warmup测试
 - 5.3.1.1 cto open
 - 5.3.1.1.1 静态warmup
 - 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
 - 5.3.1.1.2.1.1 大文件（根据缓存盘容量）并发操作
 - 5.3.1.1.2.1.2 大规模目录（1000w+）
 - 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
 - 5.3.1.1.2.2.2 大规模目录（1000w+）
 - 5.4 回归测试

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](#)

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	111856	12.154	910.045
Close	82175	9.680	413.501
Rename	4735	422.706	2669.516
Unlink	22549	11.286	824.908
Qpathinfo	101390	6.440	747.853
Qfileinfo	17699	0.019	0.108
Qfsinfo	18549	1.130	149.004
Sfileinfo	9053	5.291	290.058
Find	39162	14.795	877.050
WriteX	55122	0.076	4.072
ReadX	175813	0.119	56.392
LockX	366	0.005	0.019
UnlockX	366	0.002	0.023
Flush	7764	33.910	156.470

Throughput 5.82144 MB/sec 10 clients 10 procs max_latency=2669.523 ms

5.1.2.2 iозone

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 client
```

```

W 2022-11-29T20:33:23.822531+0800 100 mds_client.cpp:495] GetLatestTxId fail, errCode = 1008, errorText = [E1008]Reached timeout=500ms @10.182.2.46:7811, logId = 4308728
I 2022-11-29T20:33:24.409754+0800 100 client_operator.cpp:394] UpdateInodeParent oldParent = 3148032, newParent = 3148032
I 2022-11-29T20:33:24.595553+0800 94 client_operator.cpp:394] UpdateInodeParent oldParent = 3148025, newParent = 3148025
I 2022-11-29T20:33:24.852859+0800 90 client_operator.cpp:394] UpdateInodeParent oldParent = 7342288, newParent = 7342288
I 2022-11-29T20:33:24.986894+0800 89 client_operator.cpp:394] UpdateInodeParent oldParent = 2099505, newParent = 2099505
I 2022-11-29T20:33:25.041456+0800 88 client_operator.cpp:394] UpdateInodeParent oldParent = 5245188, newParent = 5245188
I 2022-11-29T20:33:25.133960+0800 95 client_operator.cpp:394] UpdateInodeParent oldParent = 8390943, newParent = 8390943
I 2022-11-29T20:33:25.249828+0800 81 client_operator.cpp:394] UpdateInodeParent oldParent = 9439520, newParent = 9439520
I 2022-11-29T20:33:25.324363+0800 101 client_operator.cpp:394] UpdateInodeParent oldParent = 3148025, newParent = 3148025
I 2022-11-29T20:33:25.475733+0800 93 client_operator.cpp:394] UpdateInodeParent oldParent = 8390938, newParent = 8390938
I 2022-11-29T20:33:25.569933+0800 96 client_operator.cpp:394] UpdateInodeParent oldParent = 3148032, newParent = 3148032
E 2022-11-29T20:33:25.668931+0800 93 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:25.687059+0800 90 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:25.766714+0800 102 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:25.794623+0800 89 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:25.825882+0800 101 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:25.887320+0800 93 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:26.021159+0800 90 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:26.101945+0800 89 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:26.125624+0800 88 fuse_client.cpp:731] rmdir not empty
E 2022-11-29T20:33:26.132129+0800 102 fuse_client.cpp:731] rmdir not empty

```

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

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

Iozone: Performance Test of File I/O

Version \$Revision: 3.429 \$

Compiled for 64 bit mode.

Build: linux-AMD64

Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins

Al Slater, Scott Rhine, Mike Wisner, Ken Goss

Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,

Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,

Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
Vangel Bojaxhi, Ben England, Vikentsi Lapa.

Run began: Thu Dec 1 10:18:38 2022

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 = 91503.61 kB/sec

Parent sees throughput for 1 initial writers = 91501.71 kB/sec

Min throughput per process = 91503.61 kB/sec

Max throughput per process = 91503.61 kB/sec

Avg throughput per process = 91503.61 kB/sec

Min xfer = 1048576.00 kB

Children see throughput for 1 rewriters	= 90770.86 kB/sec
Parent sees throughput for 1 rewriters	= 90768.49 kB/sec
Min throughput per process	= 90770.86 kB/sec
Max throughput per process	= 90770.86 kB/sec
Avg throughput per process	= 90770.86 kB/sec
Min xfer	= 1048576.00 kB
Children see throughput for 1 readers	= 167221.64 kB/sec
Parent sees throughput for 1 readers	= 167213.43 kB/sec
Min throughput per process	= 167221.64 kB/sec
Max throughput per process	= 167221.64 kB/sec
Avg throughput per process	= 167221.64 kB/sec
Min xfer	= 1048576.00 kB
Children see throughput for 1 re-readers	= 1047344.44 kB/sec
Parent sees throughput for 1 re-readers	= 1047067.22 kB/sec
Min throughput per process	= 1047344.44 kB/sec
Max throughput per process	= 1047344.44 kB/sec
Avg throughput per process	= 1047344.44 kB/sec
Min xfer	= 1048576.00 kB

iozone test complete.

```
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 = 96521.59 kB/sec

Parent sees throughput for 1 initial writers = 96519.08 kB/sec

Min throughput per process	=	96521.59 kB/sec
Max throughput per process	=	96521.59 kB/sec
Avg throughput per process	=	96521.59 kB/sec
Min xfer	=	1048576.00 kB

Children see throughput for 1 rewriters	=	96529.34 kB/sec
Parent sees throughput for 1 rewriters	=	96526.59 kB/sec
Min throughput per process	=	96529.34 kB/sec
Max throughput per process	=	96529.34 kB/sec
Avg throughput per process	=	96529.34 kB/sec
Min xfer	=	1048576.00 kB

Children see throughput for 1 readers	=	182745.86 kB/sec
Parent sees throughput for 1 readers	=	182734.62 kB/sec
Min throughput per process	=	182745.86 kB/sec
Max throughput per process	=	182745.86 kB/sec
Avg throughput per process	=	182745.86 kB/sec
Min xfer	=	1048576.00 kB

Children see throughput for 1 re-readers	= 1692895.88 kB/sec
Parent sees throughput for 1 re-readers	= 1691925.45 kB/sec
Min throughput per process	= 1692895.88 kB/sec
Max throughput per process	= 1692895.88 kB/sec

Avg throughput per process	= 1692895.88 kB/sec
Min xfer	= 1048576.00 k

```
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	= 99574.78 kB/sec
Parent sees throughput for 1 initial writers	= 99574.60 kB/sec
Min throughput per process	= 99574.78 kB/sec
Max throughput per process	= 99574.78 kB/sec
Avg throughput per process	= 99574.78 kB/sec
Min xfer	= 10485760.00 kB
Children see throughput for 1 rewriters	= 104966.91 kB/sec

Parent sees throughput for 1 rewriters	= 104966.63 kB/sec
Min throughput per process	= 104966.91 kB/sec
Max throughput per process	= 104966.91 kB/sec
Avg throughput per process	= 104966.91 kB/sec
Min xfer	= 10485760.00 kB

Children see throughput for 1 readers	= 183532.78 kB/sec
Parent sees throughput for 1 readers	= 183532.05 kB/sec
Min throughput per process	= 183532.78 kB/sec
Max throughput per process	= 183532.78 kB/sec
Avg throughput per process	= 183532.78 kB/sec
Min xfer	= 10485760.00 kB

Children see throughput for 1 re-readers	= 1674970.38 kB/sec
Parent sees throughput for 1 re-readers	= 1674905.61 kB/sec
Min throughput per process	= 1674970.38 kB/sec
Max throughput per process	= 1674970.38 kB/sec

Avg throughput per process	= 1674970.38 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/nbs/failover/test2/iozone;done  
  
#  
mpirun --allow-run-as-root -np mdtest -C -F -L -z 4 -b 10 -I 10000 -d /home/nbs/failover/test1 -w 1024
```

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
```

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 http://59.111.93.102:8080/qa/linux-5.4.tar.gz
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*
```

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,rdp
ct=50
rd=rd1,fwd=fwd*,fwdrate=max,format=restart,elapsed=2000000,interval=1

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

5.2 异常测试

操作	影响
1个etcd\mds\metaserver 网络拔出	
client 网络拔出	
client节点丢包	
kill etcd 后重启	
kill mds 后重启	
kill metaserver 后重启	
metaserver 数据迁出	
一个metasever掉电	
丢包10%	
丢包30%	
主etcd掉电	
主mds掉电	
增加metaserver数据迁入	
网络延时300ms	

5.3 新增功能测试

5.3.1 warmup测试

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

可以预先在缓存盘里创建一个文件占据缓存盘容量，人为制造缓存盘容量不足，校验文件md5一致性

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	
其他文件并发读写	
单metaserver异常（kill）	
多挂载，不共用缓存盘，并发warmup同一文件	
多挂载，共用缓存盘，并发warmup同一文件	
多挂载，共用缓存盘，并发warmup不同文件	

5.3.1.1.2.1.2 大规模目录（1000w+）

操作	结论
挂卸载 fuse	
其他目录并发读写	
单metaserver异常（kill）	
多挂载，不共用缓存盘，并发warmup同一目录	
多挂载，共用缓存盘，并发warmup同一目录	
多挂载，共用缓存盘，并发warmup不同目录	

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	

其他文件并发读写	
单metaserver异常（kill）	
多挂载，不共用缓存盘，并发warmup同一文件	
多挂载，共用缓存盘，并发warmup同一文件	
多挂载，共用缓存盘，并发warmup不同文件	

5.3.1.1.2.2.2 大规模目录（1000w+）

操作	结论
挂卸载 fuse	
其他目录并发读写	
单metaserver异常（kill）	
多挂载，不共用缓存盘，并发warmup同一目录	
多挂载，共用缓存盘，并发warmup同一目录	
多挂载，共用缓存盘，并发warmup不同目录	

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>