

小组会记录
20071114
author: pb

参加：yzf,pb,fankai,tqc,tsunami,zhulei

投 SEWM2008 的文章

内容：

yzf: 围绕与 GFS 的设计差别讨论，包括变长 chunksize，append 执行流程，open 流程，不支持并发 write 等。通过性能测试，与 google GFS 论文中的数据比较，支持这些设计的有效性。

background, related works 材料收集，大家发到 mailing list 里去。

测试：

master ops	<p>fankai: report getChunkInfo 瓶颈</p> <p>测量 ops，有如下参数</p> <ol style="list-style-type: none">1. client 的并发数。1 个 client 不能使 master 满负荷工作。提高 workload 并发数，来测试带宽（最大可用吞吐率）2. 应用中的典型文件大小（决定 Chunk 数目，从而决定典型 op 的执行比例，getFileInfo .vs. addChunk etc.）3. ChunkSize 的大小 <p>重要的关键 OP，针对 Read 操作，是 getFileInfo, getChunksInfo，addChunk, commitChunk，它们是重点测试对象。</p> <p>测试结果，可以画出吞吐率在各种参数变化情况下的图</p>
client append	<p>tqc: 18G*10 client, 4M every append call(buffer size)</p> <p>zhulei: 现在测试 append 2G，运行时间 540s，why?</p> <p>tsunami: 运行环境，在 build1 上的 script 目录下</p> <p>cexec ...</p> <p>测量参数</p> <ol style="list-style-type: none">1. client 的并发数。2. ChunkSize 的大小3. replica 数目

client read	<p>tqc: 1G*10 clients, 4M every read randomly 还需要顺序的 read 测试，这是主要访问形式。</p> <p>测量参数</p> <ol style="list-style-type: none"> 1 . client 的并发数。 2 . ChunkSize 的大小 3 . replica 数目
-------------	--

测试报告	
fankai	getChunksInfo 太慢
	chunkserver 在 chunk_tmp 底下记录了 25GB 的数据
	append call 执行慢
	master 只选择了前 3 个 chunkserver 存储数据

1. 功能测试状况

fankai:	(#2)测试计划 tfs 文件系统基本操作功能测试(?)
	testClient
zhulei:	(#3)append 并发写正确性测试(?)
	test_append

2. 待完成的工作列表

http://docs.google.com/Doc?docid=dwvtcdq_14gfcwmc&hl=zh_CN

Test	http://docs.google.com/Doc?docid=dwvtcdq_152k4z7v
	<ul style="list-style-type: none"> ● 各个模块补齐单元测试 ● (#2)测试计划 tfs 文件系统基本操作功能测试(done) ● (#3)append 并发写正确性测试(done) ● (#4)模拟 master crash，检查系统 consistency(done) ● (#5)模拟 client crash，检查有否资源泄露 ● (#6)chunkserver 功能测试

Terminal	tsunami
	<ul style="list-style-type: none"> ● 支持 put/get (done) ● 支持 history (done) ● 支持 arrow-key , tab 增强输入 (done) ● 支持!, 外部 shell 命令 (done) ● 按 shell 样式输出显示美化
Client Library	fankai
	<ul style="list-style-type: none"> ● 异常管理 , (done) ● 名字空间 api (done) ● Buffer 管理 (done) ● Client Lib 的注释/文档, 使用指南(参数设置说明) ● 编码转换, 从 client 编码到 utf-8 ● write 操作 ● stream 类支持基本数据类型的输入输出, 并支持扩展 (<<,>>操作符重载, 特定的功能函数 readLine)
ChunkServer	tqc
	<ul style="list-style-type: none"> ● 限制 tcp 连接数 ● workload 统计 ● checksum ● 数据传输 (双通道, 流水线错误逐级报告) <p>checklist:</p> <ul style="list-style-type: none"> ● pipeline 目前 read 然后 write, 能够重叠, 可能是以后性能测试的一个地方 ● 修改 datatransfer 接口, addChunk 不需要 chunkSize, ● 客户端 write 后要等待一会才能 read, why? ● master 重起后, chunkserver 必须要重起吗?
Master::FsDirectory	pb
	<p>支持 write 操作, 文件锁(done)</p> <p>支持 statistics 接口</p> <p>支持 recovery mode 的目录数据检查</p>
Master	yzf
	<ul style="list-style-type: none"> ● 支持系统状态访问接口, Monitor (done) ● Garbage collection(done) ● chunk 复制和迁移, 不同优先级的策略

--	--