分布式文件系统moosefs

高可用、可扩展的海量级分布式文件系统

什么是分布式文件系统

- 数据/文件分散存储到不同的物理设备
- 文件/数据被分块
- 文件读写并行处理
- 较低的单位成本

分布式文件系统的优点

- 高可用：存储服务器down 掉一些，服务依然是可用的
- 读写性能提高：文件分块存储在不同的物理设备，对单个设备来说，其磁盘I/O得以降低
- 容量在线可扩充：增加物理设备（服务器）就实现不停原服务而自动扩展了容量。相对于物理的raid,没有所谓的木桶效应

传统共享文件系统的缺陷

- 无高可用性：共享文件系统在一个物理设备，一旦出现故障，服务完全不可用
- 读写性能随访问量的增加而降低：访问频繁，磁盘I/O增大
- 不易实现在线扩容：一般情况下需要停机停服务

传统共享文件系统的种类

- NFS(network file system)
- Samba
- ftp
- 其他

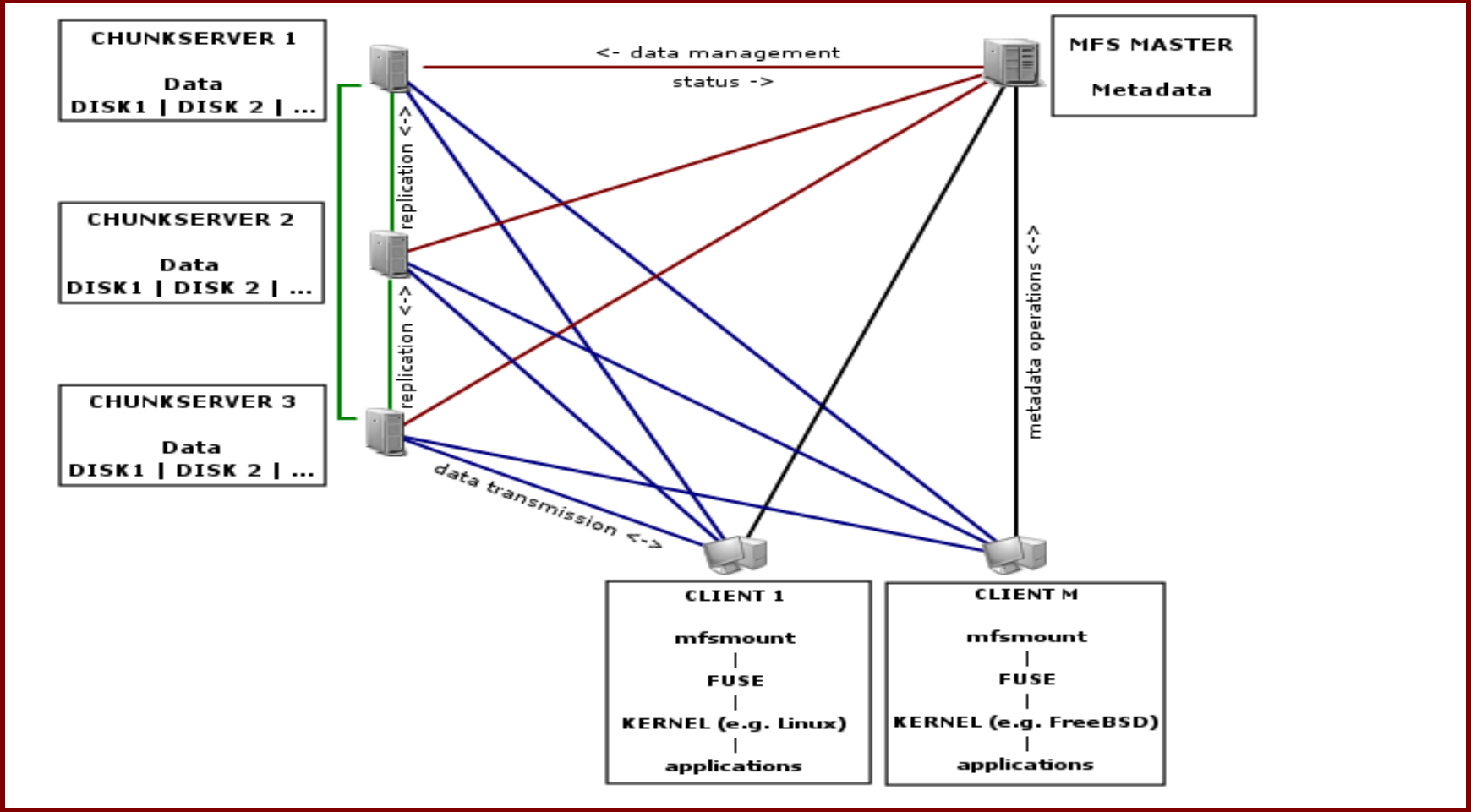
分布式文件系统的种类

- Hadoop
- FastDFS
- MooseFS
- PNFS（Parallel NFS ）
- PVFS，PVFS2
- Lustre
- 其他

为什么选Moosefs

- 实施起来简单。MFS的安装、部署、配置相对于其他几种工具来说，要简单和容易得多。看看lustre 700多页的pdf文档，让人头昏吧
- 不停服务扩容。MFS框架做好后，随时增加服务器扩充容量；扩充和减少容量皆不会影响现有的服务
- 恢复服务容易。除了MFS本身具备高可用特性外，手动恢复服务也是非常快捷的
- 我在实验过程中得到作者的帮助，这让我很是感激。

Moosefs体系结构图



MooseFS分布式文件系统的组成

- 元数据服务器（Master）
- 数据存储服务器（chunkservers）
- 客户端（clients）

元数据服务器（master）

- 分布式文件系统MooseFS的主控端：控制个数据存储服务器
- 目前只有一个master,存在单点故障
- 客户端的访问接口就是master
- 支持各种linux/unix

数据存储服务器（chunkserver）

- 数据实际存储的地方
- 由多个物理服务器组成
- 在数据存储目录，看不见实际的数据（只有带编号的目录及文件）
- 建议使用2-3个副本
- 支持各种linux/unix

Moosefs客户端

- 挂接分布式文件系统
- 一般是应用服务器
- 客户端可以是linux,freebsd等各种类unix
- 数个客户端
- Freebsd的fusefs_kmod可能会有性能问题

安装moosefs

- 元数据服务器(master)安装：配置、编译、安装。
- 数据存储服务器(chunkserver)安装：与元数据服务器相同
- 客户端安装：根据客户端的平台不同，安装稍有差异（个操作系统的fuse不同）

MooseFs客户端是linux时的安装

- 安装FUSE
- 设置环境变量
- export KG_CONFIG_PATH=/usr/local/lib/pkgconfig:$PKG_CONFIG_PATH
- 配置 ./configure –enable-mfsmount .....
- 编译安装 make;make install

moosefs客户端是freebsd时的安装

- 安装内核模块 fusefs-kmod :

Sysinstall→Configure→Packages→Kld→fusefs-kmod-0.3.9.p1_2

加载内核模块fusefs-kmod:

kldload /usr/local/modules/fuse.ko

安装pkg-config：

- 1、cd /usr/ports/devel/pkg-config
- 2、make install clean

moosefs客户端是freebsd时的安装（续）

- 安装MFS客户端
- 1、解包 tar zxvf mfs-1.5.12.tar.gz
- 2、切换目录 cd mfs-1.5.12
- 3、创建用户 pw useradd mfs –s /sbin/nologin
- 4、配置 ./configure --prefix=/usr/local/mfs --with-default-user=mfs --with-default-group=mfs --enable-mfsmount
- 5、编译安装 make ; make install

配置元数据服务器master

- 默认配置文件mfsmaster.cfg(不需要修改即可使用)
- # WORKING_USER = mfs
- # WORKING_GROUP = mfs
- # LOCK_FILE = /var/run/mfs/mfsmaster.pid
- # DATA_PATH = /usr/local/mfs/var/mfs
- # SYSLOG_IDENT = mfsmaster
- # BACK_LOGS = 50
- # REPLICATIONS_DELAY_INIT = 300
- # REPLICATIONS_DELAY_DISCONNECT = 3600
- # MATOCS_LISTEN_HOST = *
- # MATOCS_LISTEN_PORT = 9420
- # MATOCU_LISTEN_HOST = *
- # MATOCU_LISTEN_PORT = 9421
- # CHUNKS_LOOP_TIME = 300
- # CHUNKS_DEL_LIMIT = 100
- # CHUNKS_REP_LIMIT = 15

数据存储服务器chunkserver配置

- 2个配置文件：主配置文件mfschunkserver.cfg及共享磁盘配置文件mfshdd.cfg

数据存储服务器chunkserver配置(续)

- Mfschunkserver.cfg
- #WORKING_USER = mfs
- #WORKING_GROUP = mfs
- # DATA_PATH = /usr/local/mfs/var/mfs
- # LOCK_FILE = /var/run/mfs/mfschunkserver.pid
- # SYSLOG_IDENT = mfschunkserver
- # BACK_LOGS = 50
- # MASTER_RECONNECTION_DELAY = 30
- MASTER_HOST = 192.168.0.19
- MASTER_PORT = 9420
- # MASTER_TIMEOUT = 60
- # CSSERV_LISTEN_HOST = *
- # CSSERV_LISTEN_PORT = 9422
- # CSSERV_TIMEOUT = 60
- # CSTOCS_TIMEOUT = 60
- # HDD_CONF_FILENAME = /usr/local/mfs/etc/mfshdd.cfg

数据存储服务器chunkserver配置(续)

- 共享磁盘配置文件mfshdd.cfg
- /data1
- /data2

Moosefs分布式文件系统启/停

- 元数据服务器启动：mfsmaster start（mfsmaster -s）
- 数据存储服务器启动：mfschunkserver start(mfschunkserver -s)
- 客户端启动：mfsmount挂接moosefs的元数据服务器

挂接和使用moosefs

- Mfsmount –h ip (客户机操作)
- 默认的挂接点是/mnt/mfs.可以用选项-w 改变挂接点
- 在客户端执行ls mkdir cp等操作测试

MooseFS客户端常用工具

```
[root@mysql-bk ~]# ll /usr/local/mfs/bin/
total 292
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfscheckfile -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfsdirinfo -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfsfileinfo -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfsgetgoal -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfsgettrashtime -> mfstools
-rwxr-xr-x 1 root root 185509 Mar  6 11:43 mfsmount
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfsrgetgoal -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfsrgettrashtime -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfsrsetgoal -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfsrsettrashtime -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfssetgoal -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfssettrashtime -> mfstools
lrwxrwxrwx 1 root root      8 Mar  6 11:43 mfssnapshot -> mfstools
-rwxr-xr-x 1 root root  49121 Mar  6 11:43 mfstools
```

查看挂接情况

- [root@mysql-bk ~]# df -h
- Filesystem Size Used Avail Capacity Mounted on
- /dev/ad4s1a 26G 570M 24G 2% /
- devfs 1.0K 1.0K 0B 100% /dev
- /dev/ad4s1g 356G 157G 170G 48% /data
- /dev/ad4s1f 17G 215M 15G 1% /home
- /dev/ad4s1d 28G 1.1G 25G 4% /usr
- /dev/ad4s1e 24G 362M 21G 2% /var
- /dev/fuse0 2.5T 256G 2.2T 11% /mnt/mfs

Moosefs分布是系统状态查看

- 主要查看元数据服务器master系统日志

```
mfs-ctrl# tail -f /var/log/messages
Mar 27 08:13:00 mfs-ctrl mfsmaster[29647]: inodes: 5902006
Mar 27 08:13:00 mfs-ctrl mfsmaster[29647]: dirnodes: 5005394
Mar 27 08:13:00 mfs-ctrl mfsmaster[29647]: filenodes: 896612
Mar 27 08:13:00 mfs-ctrl mfsmaster[29647]: chunks: 899177
Mar 27 08:13:00 mfs-ctrl mfsmaster[29647]: chunks to delete: 0
Mar 27 08:13:00 mfs-ctrl mfsmaster[29647]: chunkservers status:
```

监控

- 服务监控：元数据服务器tcp端口9420，9421；数据存储服务器tcp端口9422
- 服务器主机资源监控：最主要的是磁盘空间监控

完毕，谢谢

- 田逸（sery@163.com）

- 2009-8-29