一、MFS基础

1.1.分布式文件系统

分布式文件系统（Distributed File System，DFS）是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点（可简单的理解为一台计算机）相连；

DFS为分布在网络上任意位置的资源提供一个逻辑上的树形文件系统结构，从而使用户访问分布在网络上的共享文件更加简便。对于用户来说，要访问这个文件系统，只要打开这个虚拟共享文件夹，就可以看到所有链接到虚拟共享文件夹内的共享文件，用户感觉不到这些共享文件是分散在各个计算机上的。

分布式文件系统的好处是集中访问、简化操作、数据容灾，以及提高文件的存取性能。

常见的分布式文件系统有：MFS（块大小默认64M）、HDFS（块大小默认128M)

1.2.MFS分布式文件系统

MFS：（MooseFS:驯鹿文件系统) 是一个具备冗余容错功能的分布式网络文件系统，它将数据分别存放在多个物理服务器或单独磁盘或分区上，确保一份数据有多个备份副本，然而对于访问 MFS 的客户端或者用户来说，整个分布式网络文件系统集群看起来就像一个资源一样，MFS是基于FUSE（用户空间文件系统）机制的，所以支持含有FUSE的操作方式的系统。

冗余：指的是一份数据会备份到多个服务器内，可以手动指定（默认2份副本）

优点：

> 1.高可靠性：每一份数据可以设置多个备份（多分数据），并可以存储在不同的主机上
>
> 2.高可扩展性：可以很轻松的通过增加主机的磁盘容量或增加主机数量来动态扩展整个文件系统的存储量
>
> 3.高可容错性：可以通过对mfs进行系统设置，实现当数据文件被删除后的一段时间内，依旧存放于主机的回收站中，以备误删除恢复数据
>
> 4.高数据一致性：即使文件被写入、访问时，我们依然可以轻松完成对文件的一致性快照

缺点：

> 1.Master目前是单点，企业版的不是单点的，数据会很安全；对于社区版来说是单点，有数据安全风险，这里我们可以使用keepalived+master的方式，实现master节点的高可用；虽然会把数据信息同步到备份服务器，但是恢复需要时间
>
> 2.Master服务器对主机的内存要求略高，是因为Master节点在开启状态时，会将所有的元数据全部加装到内存中
>
> 3.默认Metalogger复制元数据时间较长

1.3.MFS文件系统的组成

> - **元数据服务器（Master）：在整个体系中负责管理文件系统，维护元数据**
> - **元数据日志服务器（MetaLogger）：备份Master服务器的变化日志文件**
> - **数据存储服务器（Chunk Server）：真正存储数据的服务器**
> - **客户端（Client）：可以像挂载NFS一样，挂载MFS文件系统**

1.4.MFS读取数据的处理过程

> 1. **客户端向元数据服务器发出读请求**
> 2. **元数据服务器把所需数据存放的位置（Chunk Server的IP地址和Chunk编号）告知客户端**
> 3. **客户端向已知的Chunk Server请求发送数据**
> 4. **Chunk Server向客户端发送数据**

1.5.MFS写入数据的处理过程

> 1. **客户端向元数据服务器发送写入请求**
> 2. **元数据服务器与Chunk Server进行交互，但元数据服务器只在某些服务器创建新的分块Chunks，创建成功后由Chunk Servers告知元数据服务器操作成功**
> 3. **元数据服务器告知客户端，可以在哪个Chunk Server的哪些Chunks写入数据**
> 4. **客户端向指定的Chunk Server写入数据**
> 5. **该Chunk Server与其他Chunk Server进行数据同步，同步成功后Chunk Server告知客户端数据写入成功**
> 6. **客户端告知元数据服务器本次写入完毕**

1.5.MFS端口号

> 9420： MFS master和MFS chunck通信端口
>
> 9421： MFS master和MFS Client端通信端口
>
> 9419： MFS master和MFS metalogger端通信端口
>
> 9422： MFS chunck 和MFS Client端通信端口
>
> 9425： MFS master web界面监听端口，查看整体运行状态

二、MFS部署应用

2.1.MFS项目说明

> 公司之前的图片服务器采用的是NFS，随着业务量增加，暴露出以下问题：
>
> 1. 多台服务器通过NFS方式共享一个服务器的存储空间，使得NFS服务器不堪重负。而且NFS存在着单点故障问题
> 2. 在当前多种常用的分布式文件系统中，我们采用了MFS（MooseFS），它是一个具有容错功能的，高可用、可扩展的海量级分布式文件系统

准备环境

主机名	操作系统	IP地址	主要软件
mfsmaster	CentOS 7	192.168.1.101	moosefs-v3.0.115.tar.gz
metalogger	CentOS 7	192.168.1.102	moosefs-v3.0.115.tar.gz
chunkserver1	CentOS 7	192.168.1.103	moosefs-v3.0.115.tar.gz
chunkserver2	CentOS 7	192.168.1.104	moosefs-v3.0.115.tar.gz
client	CentOS 7	192.168.1.105	moosefs-v3.0.115.tar.gz fuse-2.9.2.tar.gz

```
1、清空防火墙、关闭selinux
[root@jianfeng ~]# systemctl stop firewalld
[root@jianfeng ~]# setenforce 0
```

```
2、所有主机设置hosts文件
[root@jianfeng ~]# vim /etc/hosts
192.168.1.10    mfsmaster
192.168.1.11    metalogger
192.168.1.12    chunk1
192.168.1.13    chunk2
192.168.1.14    mfsclient

3、所有主机设置主机名
[root@jianfeng ~]# hostnamectl set-hostname mfsmaster
[root@jianfeng ~]# bash

4、安装编译工具及依赖
yum install gcc make libpcap-devel zlib-devel fuse fuse-devel fuse-libs
pkgconfig

fuse概述：用户空间文件系统（Filesystem in Userspace，简称FUSE）是操作系统中的概念，指完全在用户态实现的文件系统。目前Linux通过内核模块对此进行支持。
注：2.6以上内核，都自带fuse
```

2.2.MFS编译安装

所有主机mfs安装过程一至。

```
1、创建mfs用户
[root@jianfeng ~]# useradd -M -s /sbin/nologin moosefs

2、安装源码包
[root@jianfeng ~]# tar -zxvf moosefs-v3.0.115.tar.gz -C /usr/src/
[root@jianfeng ~]# cd /usr/src/ moosefs-3.0.115/
[root@jianfeng moosefs-3.0.115]# ./configure \
--prefix=/usr/local/moosefs \
--with-default-user=moosefs \
--with-default-group=moosefs \
--enable-mfsmount

3、编译 && 安装
[root@node1 moosefs-3.0.115]# grep processor /proc/cpuinfo | wc -l
[root@node1 moosefs-3.0.115]# nproc
以上两个命令可获取cpu数量

[root@node1 moosefs-3.0.115]# make -j 2
[root@node1 moosefs-3.0.115]# make install -j 2
-j选项后面跟cpu数量提高编译安装速度


4、复制配置文件
[root@jianfeng mfs-1.6.27]# cd /usr/local/moosefs/etc/mfs

[root@jianfeng mfs]# cp mfsmaster.cfg.sample mfsmaster.cfg              //Master
配置文件
[root@jianfeng mfs]# cp mfschunkserver.cfg.sample mfschunkserver.cfg    //数据存储
服务器配置文件
[root@jianfeng mfs]# cp mfsexports.cfg.sample mfsexports.cfg            //共享的目
录文件设置
```

```
[root@jianfeng mfs]# cp mfstopology.cfg.sample mfstopology.cfg          //定义mfs的网络访问
[root@jianfeng mfs]# cp mfsmetalogger.cfg.sample mfsmetalogger.cfg      //元数据日志服务器配置文件
[root@jianfeng mfs]# cp mfsmount.cfg.sample mfsmount.cfg                //客户端配置文件
[root@jianfeng mfs]# cp mfshdd.cfg.sample mfshdd.cfg                    //存储空间配置文件
```

2.3.Master配置

```
1、首次安装master 时，会自动生成一个名为metadata.mfs.empty的元数据文件metadata,该文件是空的。MooseFS master 必须有文件metadata.mfs
[root@jianfeng ~]# cd /usr/local/moosefs/var/mfs
[root@jianfeng mfs]# mv metadata.mfs.empty metadata.mfs

2、mfsmaster.cfg配置文件介绍
[root@jianfeng mfs]# cd /usr/local/moosefs/etc/mfs
[root@jianfeng mfs]# vim mfsmaster.cfg
凡是用#注释掉的变量均使用其默认值：对于这个文件不需要做任何修改就可以工作。
#WORKING_USER和WORKING_GROUP：是运行master server的用户和组；
#SYSLOG_IDENT：是master server在syslog中的标识，也就是说明这是由master serve产生的；
#LOCK_MEMORY：是否执行mlockall()以避免mfsmaster 进程溢出（默认为0）；
#NICE_LEVE：运行的优先级(如果可以默认是 -19; 注意: 进程必须是用root启动)；
#EXPORTS_FILENAME：被挂接目录及其权限控制文件的存放位置
#DATA_PATH：数据存放路径，此目录下大致有三类文件，changelog，sessions和stats；
#BACK_LOGS：metadata的改变log文件数目(默认是 50)；
#REPLICATIONS_DELAY_INIT：延迟复制的时间（默认是300s）；
#REPLICATIONS_DELAY_DISCONNECT：chunkserver断开的复制延迟（默认是3600）；
# MATOML_LISTEN_HOST：metalogger监听的IP地址(默认是*，代表任何IP)；
# MATOML_LISTEN_PORT：metalogger监听的端口地址(默认是9419)；
# MATOCS_LISTEN_HOST：用于chunkserver连接的IP地址（默认是*，代表任何IP）；
# MATOCS_LISTEN_PORT：用于chunkserver连接的端口地址（默认是9420）；
# MATOCU_LISTEN_HOST：用于客户端挂接连接的IP地址(默认是*，代表任何IP)；
# MATOCU_LISTEN_PORT：用于客户端挂接连接的端口地址（默认是9421）；
# CHUNKS_LOOP_TIME ：chunks的回环频率（默认是：300秒）；
# CHUNKS_DEL_LIMIT :在一个loop设备中可以删除chunks的最大数 (默认：100)
# REPLICATIONS_DELAY_DISCONNECT chunkserver断开后的复制延时（默认：3600秒）
# CHUNKS_WRITE_REP_LIMIT：在一个循环里复制到一个chunkserver的最大chunk数目（默认是1）
# CHUNKS_READ_REP_LIMIT ：在一个循环里从一个chunkserver复制的最大chunk数目（默认是5）
# REJECT_OLD_CLIENTS：弹出低于1.6.0的客户端挂接（0或1，默认是0）


3、mfsexports.cfg配置文件:注意mfsexports访问控制对于那些老客户是没用的
[root@jianfeng mfs]# vim mfsexports.cfg
*    .    rw
192.168.3.138    /    rw,alldirs,maproot=0,password=111111

配置文件内容注解：
    该文件每一个条目分为三部分：
        第一部分：客户端的ip地址
        第二部分：被挂接的目录
        第三部分：客户端拥有的权限

地址可以指定的几种表现形式：
    *                    所有的ip地址
```

```
    n.n.n.n             单个ip地址
    n.n.n.n/b           IP网络地址/位数掩码
    n.n.n.n/m.m.m.m     IP网络地址/子网掩码
    f.f.f.f-t.t.t.t     IP范围
```

目录部分需要注意两点：

```
    /   标识MooseFS 根;
    .   表示MFSMETA 文件系统
```

权限部分：

```
    ro          只读模式共享
    rw          可写方式共享
    alldirs     许挂载任何指定的子目录
    maproot     映射为root，还是指定的用户
    password    指定客户端密码
```

4、启动Master

```
[root@jianfeng mfs]# ln -s /usr/local/moosefs/sbin/mfsmaster /usr/local/sbin/
[root@jianfeng mfs]# mfsmaster start
open files limit has been set to: 16384
working directory: /usr/local/moosefs/var/mfs
lockfile created and locked
initializing mfsmaster modules ...
exports file has been loaded
topology file has been loaded
loading metadata ...
loading sessions data ... ok (0.0001)
loading storage classes data ... ok (0.0000)
loading objects (files,directories,etc.) ... ok (0.9192)
loading names ... ok (0.0000)
loading deletion timestamps ... ok (0.0000)
loading quota definitions ... ok (0.0000)
loading xattr data ... ok (0.0000)
loading posix_acl data ... ok (0.0000)
loading open files data ... ok (0.0000)
loading flock_locks data ... ok (0.0000)
loading posix_locks data ... ok (0.0000)
loading chunkservers data ... ok (0.0000)
loading chunks data ... ok (0.0000)
checking filesystem consistency ... ok
connecting files and chunks ... ok
all inodes: 1
directory inodes: 1
file inodes: 0
chunks: 0
metadata file has been loaded
stats file has been loaded
master <-> metaloggers module: listen on *:9419
master <-> chunkservers module: listen on *:9420
main master server module: listen on *:9421
mfsmaster daemon initialized properly
```

5、检查是否启动mfs

```
[root@jianfeng mfs]# netstat -anpt | grep mfsmaster
tcp        0      0 0.0.0.0:9419                0.0.0.0:*
LISTEN      39617/mfsmaster
tcp        0      0 0.0.0.0:9420                0.0.0.0:*
LISTEN      39617/mfsmaster
```

```
tcp        0      0 0.0.0.0:9421                0.0.0.0:*
LISTEN      39617/mfsmaster
```

2.4.MetaLogger配置

```
1、配置文件修改
[root@node2 mfs]# vim /usr/local/moosefs/etc/mfs/mfsmetalogger.cfg
MASTER_HOST = mfsmaster     //msaster服务器主机名也可使用IP

2、启动MetaLogger
[root@jianfeng mfs]# ln -s /usr/local/mfs/sbin/mfsmetalogger /usr/local/sbin/
[root@jianfeng mfs]# mfsmetalogger start
open files limit has been set to: 4096
working directory: /usr/local/moosefs/var/mfs
lockfile created and locked
initializing mfsmetalogger modules ...
mfsmetalogger daemon initialized properly

3、检查
[root@jianfeng mfs]# netstat -anpt | grep mfs
tcp        0      0 192.168.1.101:42031         192.168.1.100:9419    ESTABLISHED
17901/mfsmetalogger
```

2.5.ChunkServer配置

```
1、配置文件修改
[root@jianfeng ~]# vim /usr/local/moosefs/etc/mfs/mfschunkserver.cfg
MASTER_HOST = mfsmaster     //msaster服务器主机名也可使用IP

2、磁盘空间配置文件
[root@node1 ~]# vim /usr/local/moosefs/etc/mfs/mfshdd.cfg
/data   //末尾添加，表示使用该文件夹存储文件

3、将新增加磁盘分区并挂载至/data目录
[root@jianfeng ~]# fdisk /dev/sdx  //过程省略
[root@jianfeng ~]# mkdir /data
[root@jianfeng ~]# chown -R moosefs:moosefs  /data
[root@jianfeng ~]# mount /dev/sdx /data

4、启动ChunkServer
[root@jianfeng ~]# ln -s /usr/local/moosefs/sbin/mfschunkserver /usr/local/sbin/
[root@jianfeng ~]# mfschunkserver start
open files limit has been set to: 16384
working directory: /usr/local/moosefs/var/mfs
lockfile created and locked
setting glibc malloc arena max to 4
setting glibc malloc arena test to 4
initializing mfschunkserver modules ...
hdd space manager: path to scan: /data/
hdd space manager: start background hdd scanning (searching for available
chunks)
main server module: listen on *:9422
no charts data file - initializing empty charts
mfschunkserver daemon initialized properly

3、检查
```

```
[root@jianfeng ~]# netstat -anpt | grep mfs
tcp        0      0 0.0.0.0:9422                0.0.0.0:*            LISTEN
12976/mfschunkserve
tcp        0      0 192.168.1.102:42268         192.168.1.100:9420   ESTABLISHED
12976/mfschunkserve
```

2.6.Client配置

```
1、创建文件夹用于挂在mfs文件系统
[root@jianfeng ~]# modprobe fuse
[root@jianfeng ~]# mkdir /mnt/mfs
[root@jianfeng ~]# ln -s /usr/local/moosefs/bin/mfsmount /usr/local/bin

2、客户端挂载MFS服务器硬盘
[root@jianfeng ~]# mfsmount /mnt/mfs/ -H mfsmaster
-H选项也可以跟ip

3、验证
[root@jianfeng ~]# df -Th
Filesystem         Type      Size  Used Avail Use% Mounted on
/dev/sda3          ext4       17G  4.0G   12G  25% /
tmpfs              tmpfs     238M  224K  238M   1% /dev/shm
/dev/sda1          ext4     1008M   58M  900M   6% /boot
/dev/sr0           iso9660   4.2G  4.2G     0 100% /media/CentOS_6.5_Final
mfsmaster:9421     fuse.mfs   37G     0   37G   0% /mnt/mfs                  //看
这里
```

三、MFS常用操作

3.1.副本操作

副本设置的最高数量等于chunkserver的数量，如果存储副本的机器损坏，默认会再选择一台存储副本，如果损坏修复，会再删除一个副本

```
#查看副本数量
[root@jianfeng mfs]# mfsgetgoal a.txt
a.txt: 2   #默认两份

#设置副本数目
[root@jianfeng mfs]# mfssetgoal 2 a.txt #设置a.txt的副本数目

#查看副本详细信息
[root@jianfeng mfs]# mfsfileinfo a.txt
a.txt:
    chunk 0: 0000000000000003_00000001 / (id:3 ver:1)
    copy 1: 192.168.4.112:9422 (status:VALID)
    copy 2: 192.168.4.115:9422 (status:VALID)
```

3.2.MFS元数据日志

在日志服务器查看日志

```
[root@jianfeng mfs]# cd /usr/local/moosefs/var/mfs
[root@jianfeng mfs]# cat changelog_ml.0.mfs
0: 1526698816|SESSION():1
1: 1526699254|SETGOAL(1,0,3,4):1,0,0
2: 1526699277|ACCESS(1)
3: 1526699277|ATTR(1,511,500,500,1526699277,1526695191)
4: 1526699291|SETGOAL(1,0,3,4):0,1,0
....
```

在master服务器查看日志

```
[root@jianfeng mfs]# cd /usr/local/mfs/var/mfs
[root@jianfeng mfs]# cat changelog_ml.0.mfs
0: 1526698816|SESSION():1
1: 1526699254|SETGOAL(1,0,3,4):1,0,0
2: 1526699277|ACCESS(1)
3: 1526699277|ATTR(1,511,500,500,1526699277,1526695191)
4: 1526699291|SETGOAL(1,0,3,4):0,1,0
....
```

3.3.Master服务器恢复

将主服务器进行还原操作，模拟主服务器崩溃。拷贝元数据备份服务器的changelog文件，该文件在metalogger服务器中的/usr/local/moosefs/var/mfs目录下，将该目录下的备份文件拷贝至对应的主服务器的/usr/local/moosefs/var/mfs/下。

```
1、metalogger元数据日志服务器
[root@jianfeng ~]# cd /usr/local/moosefs/var/mfs
[root@jianfeng mfs]# scp -rp ./* root@mfsmaster:/usr/local/moosefs/var/mfs/

2、Master服务器，启动Master
[root@jianfeng ~]# mfsmaster start

3、Master服务器检查
[root@jianfeng ~]# netstat -anpt | grep mfsmaster
```

3.4.MFS监控页面

```
1、Master服务器上启动
[root@jianfeng ~]# /usr/local/moosefs/sbin/mfscgiserv
[root@jianfeng ~]# netstat -anpt | grep 9425
tcp        0      0 0.0.0.0:9425                0.0.0.0:*
LISTEN      39890/python

2、使用浏览器访问http://192.168.1.100:9425
Info                显示mfs基本信息
Servers             列出现有Chunk Server
Disks               列出现有Chunk Server 硬盘信息
Exports             列出可被挂载的目录
mounts              列出被挂载的目录
Operator            显示在执行的操作
Master charts       显示Master Server 的操作情况,读写删除等操作
Server charts       显示Chunk Server的操作情况,数据传输率及系统状态
```

3.5.MFS集群维护

启动MFS集群

> 安全的启动MooseFS 集群（避免任何读或写的错误数据或类似的问题）的方式是按照以下命令步骤：
>
> 1. 启动mfsmaster 进程
> 2. 启动所有的mfschunkserver 进程
> 3. 启动mfsmetalogger 进程（如果配置了mfsmetalogger）
> 4. 当所有的chunkservers 连接到MooseFS master 后，任何数目的客户端可以利用mfsmount去挂接被export 的文件系统。（可以通过检查master 的日志或是CGI 监视器来查看是否所有的chunkserver被连接）。

停止MFS集群

> 安全的停止MooseFS 集群：
>
> 1. 在所有的客户端卸载MooseFS 文件系统（用umount 命令或者是其它等效的命令）
> 2. 用mfschunkserver stop 命令停止chunkserver 进程
> 3. 用mfsmetalogger stop 命令停止metalogger 进程
> 4. 用mfsmaster stop 命令停止master 进程