

问题描述

parted命令使用实践

解决方法

1. parted的适用场景

注意：所有操作立即生效，和fdisk需要保存不同；

fdisk只支持MBR分区，MBR分区表最大支撑2T的磁盘，所以无法划分大于2T的分区。而parted工具可以划分单个分区大于2T的GPT格式的分区，也可以划分普通的MBR分区。

2. parted的2种使用方式

交互式：手动按序交互式的创建；

非交互式：可将命令行写在脚本中，运行脚本实现一键创建；适用于远程批量管理多台主机的场景。

两种方式比较：两种方法的使用和原理其实是一样的；要想实现非交互式创建，无非是在每一个具体的交互式命令前加上parted DEVICES_NAME即可。

3. 交互式创建分区

3.1. 交互式方式实现分区步骤

3.1.1. 查询磁盘的已有分区

```
# parted /dev/sdl print
```

或：

```
fdisk -l
```

```
lsblk
```

3.1.2. 选择磁盘

parted命令后跟上欲操作磁盘的名字即可选择此设备进行操作。

```
# parted /dev/sdb
```

3.1.3. 新建磁盘标签类型为GPT

要正确对2TB以上硬盘分区，需使用GPT格式。这里必须将新建的磁盘标签格式设为gpt。

```
(parted) mklabel gpt
```

3.1.4. 分区

将/dev/sdb整个空间分给同一个分区

```
(parted) mkpart primary 0 -1
```

输入i忽略警告；

输入p查看验证创建的分区的分区信息；或print；

输入q退出命令行；

3.1.5. 分区命令格式参考

语法：mkpart PART-TYPE [FS-TYPE] START END

1) PART-TYPE(分区类型)

- Primary：主分区
- Logical：逻辑分区
- Extended：扩展分区

2) FS-TYPE(文件系统类型)

- ext4
- ext3
- ext2
- xfs
- 其他.....

3) START

- 设定磁盘分区起始点；可以为0，numberMiB/GiB/TiB；
- 0：设定当前分区的起始点为磁盘的第一个扇区；
- 1G：设定当前分区的起始点为磁盘的1G处开始；
- 百分比，如0%，10%；

4) END

- 设定磁盘分区结束点；
- 1 设定当前分区的结束点为磁盘的最后一个扇区；
- 10G 设定当前分区的结束点为磁盘的10G处；

d) 百分比, 如100%;

3.1.6. 格式化分区

```
# mkfs -t ext4 /dev/sdb1
```

3.1.7. 设定分区label(非必要)

```
# e2label /dev/sdb1 /gfsdata01
```

3.2. 挂载分区

3.2.1. 查询/dev/sdb1的blkid

```
# blkid /dev/sdb1
```

3.2.2. 创建挂载目录

```
# mkdir /gfsdata01
```

3.2.3. 临时挂载分区

```
# mount /dev/sdb1 /gfsdata01  
# df -h
```

3.2.4. 开机自动挂载(永久挂载)

即修改/etc/fstab文件。

```
# echo '/dev/sdb1 /gfsdata01 ext4 defaults 0 0' >>/etc/fstab
```

3.3. 非交互式方式实现脚本

适用于需要在多台主机上进行同样的分区操作, 可以通过ansible工具调用脚本实现批量分区配置挂载等操作。

```
[root@kvm ~]# cat ./auto_partition.sh  
#!/bin/bash  
# 新建/dev/sdb的磁盘标签类型为GPT  
parted /dev/sdb mklabel gpt  
# 将/dev/sdb整个空间分给同一个分区  
parted /dev/sdb mkpart primary 0 100%  
# 忽略警告  
ignore  
# 格式化分区  
mkfs -t ext4 /dev/sdb1  
# 设定分区label(非必要)  
e2label /dev/sdb1 /gfsdata01  
# 创建挂载目录  
mkdir /gfsdata01  
# 临时挂载  
mount /dev/sdb1 /gfsdata01  
# 开机自动挂载(永久挂载)  
echo '/dev/sdb1 /gfsdata01 ext4 defaults 0 0' >>/etc/fstab  
[root@kvm ~]#
```

创建3个分区, 先删后建;

```
# parted /dev/sdb print  
# parted /dev/sdb rm 1  
# parted /dev/sdb print  
# parted /dev/sdb mkpart primary 0% 25%  
# parted /dev/sdb mkpart primary 25% 45%  
# parted /dev/sdb mkpart primary 45% 70%
```