

kvm虚拟机数据盘扩容方案

在传统情况下, 虚拟机数据盘都要进行分区操作, 这样无法方便地扩容虚拟磁盘并更新分区偏移, 导致扩容磁盘难度很大.

此前使用增加虚拟磁盘的方式来解决, 当一块磁盘用满, 则添加新的虚拟磁盘.

下面提供一种新的扩容办法, 制备好独立的数据盘后, 扩容期间只需关闭并启动(重启虚拟机不会更新配置)一次虚拟机, 以及几条命令, 即可完成虚拟机扩容.

- 添加数据盘
 - 创建并添加数据盘
 - 格式化并挂载数据盘
- 数据盘扩容
 - 注释fstab挂载项并关机
 - 开启虚拟机并调整文件系统大小
 - 扩容结束, 开启程序
- 技术要点

添加数据盘

创建并添加数据盘

用来存放数据的虚拟磁盘, 以下称之为数据盘.

首先需要创建虚拟磁盘:

```
[root@kvm-host domid]# qemu-img create -f qcow2 test.data.img 50G
```

然后编辑虚拟机: `virsh edit domid`, 找到<file>节点, 在其后添加一个新的file节点, 内容如下所示:

```
<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2' cache='writeback' />
  <source file='/path/to/test.data.img' /> <!-- 填写虚拟磁盘在主机路径 -->
  <target dev='vdb' bus='virtio' /> <!-- 虚拟磁盘的设备名, 一般为vdb -->
  <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' /> <!-- 本行可以删掉,
系统会自动创建 -->
</disk>
```

格式化并挂载数据盘

此时您可以`virsh start domid`来启动虚拟机.

进入系统后, 开始格式化新添加的数据盘. 注意, 这是直接格式化, 没有以前的分区操作. 而系统盘还是需要分区的, 不然没法安装bootloader.

```
mkfs.ext4 /dev/vdb -L voldata
```

然后您可以挂载数据盘了, 然后查看下数据盘的大小:

```
# df -h
...
/dev/vdb  49G  1G  48G  2% /opt
```

至于后续的写入 /etc/fstab , 则不再赘述.

数据盘扩容

在添加完数据盘后, 可能需要对其进行扩容.

注释fstab挂载项并关机

为了避免下次开机后有程序使用数据盘时卸载不便, 预先注释掉 /etc/fstab 里的条目. 然后关闭系统.

在主机调整数据盘大小

```
[root@kvm-host domid]# qemu-img resize test.data.img 90G # 当然, 您可以使用+size或者-size这种格式,
不过qcow2格式只能增不能减.
```

开启虚拟机并调整文件系统大小

注意: 以下操作须在数据盘卸载状态下进行. 如为挂载状态, 请使用lsof找出锁定文件的进程, 结束之, 然后尝试卸载数据盘.

```
[root@vm ~]# e2fsck -f /dev/vdb # 注意, 现在是在虚拟机里操作.
[root@vm ~]# resize2fs /dev/vdb # 程序会自动识别磁盘大小, 并占满.
[root@vm ~]# vim /etc/fstab # 如果您之前有调整fstab, 此时可以恢复了
[root@vm ~]# mount -a # 自动挂载所有磁盘
[root@vm ~]# df -h # 检查下数据盘大小
...
/dev/vdb  89G  16G  69G  19% /opt
```

扩容结束, 开启程序

磁盘扩容到此结束, 您可以手动开启您的程序了.

声明: 您可以选择重启系统来自动启动程序, 但这次重启不是必要操作. 我们不对这次重启负责.

技术要点

如果虚拟磁盘先行分区, 则需要调整mbr里的分区表信息, 操作繁琐. 但是不必添加额外磁盘.

但直接把虚拟磁盘当作分区使用, 则不需维护分区表信息. 操作方便. 不过多加块虚拟磁盘罢了.