

云硬盘

最佳实践

产品文档



腾讯云

【版权声明】

©2015-2016 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

文档声明.....	2
多块弹性云盘构建RAID组.....	4
多块弹性云盘构建LVM逻辑卷.....	8

多块弹性云盘构建RAID组

RAID（独立磁盘冗余阵列，Redundant Array of Independent Disks）可以将多个磁盘组合起来，构成一个磁盘阵列组，以提高数据的读写性能和可靠性。同时操作系统只会将磁盘阵列组当作一个硬盘来使用。目前RAID有多种等级，以下将介绍RAID0、RAID1、RAID01和RAID10。根据选择版本不同，磁盘阵列组相较于一块容量相当的大硬盘有增强数据集成度、增强容错功能、增加处理量或容量等优势。

以下是不同RAID版本的对比信息：

RAID等级

下面介绍如何使用4块腾讯云弹性云盘来构建RAID0阵列。Linux内核提供了md模块在底层管理RAID设备，我们可以使用mdadm工具来调用md模块。

```
[root@VM_63_126_centos ~]# fdisk -l | grep /dev/vd | grep Linux | grep -v vda
/dev/vdc1          1      20805      10485688+   83  Linux
/dev/vdc2          20806      27046      3145464     83  Linux
/dev/vdd1          1      20805      10485688+   83  Linux
/dev/vde1          1      20805      10485688+   83  Linux
/dev/vdf1          1      20805      10485688+   83  Linux
[root@VM_63_126_centos ~]#
```

注：请及时对将要到期的弹性云盘进行续费操作，以避免由于弹性云盘到期导致被系统强制隔离对RAI

D阵列产生影响。

1. 安装mdadm (以CentOS为例)

```
[root@VM_63_126_centos ~]# yum install mdadm -y
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
--> Package mdadm.x86_64 0:3.3.2-5.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch                Version             Repository           Size
=====
Installing:
mdadm                  x86_64              3.3.2-5.el6        os                   345 k
=====
Transaction Summary
=====
Install      1 Package(s)

Total download size: 345 k
Installed size: 800 k
Downloading Packages:
mdadm-3.3.2-5.el6.x86_64.rpm                | 345 kB    00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : mdadm-3.3.2-5.el6.x86_64      1/1
  Verifying  : mdadm-3.3.2-5.el6.x86_64      1/1

Installed:
  mdadm.x86_64 0:3.3.2-5.el6

Complete!
```

2. 使用mdadm创建RAID0

```
[root@VM_63_126_centos ~]# mdadm --create /dev/md0 --level=0 --raid-devices=4 /dev/vd[cdef]1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
[root@VM_63_126_centos ~]#
```

注：创建RAID1、RAID01、RAID10时最好使用相同大小的分区创建RAID，以避免对磁盘空间的浪费

。

3. 使用mkfs创建文件系统

```
[root@VM_63_126_centos ~]# mkfs.ext3 /dev/md0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=512 blocks
2621440 inodes, 10477056 blocks
523852 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
320 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

4. 挂载文件系统

```
[root@VM_63_126_centos ~]# mount /dev/md0 md0/
[root@VM_63_126_centos ~]# tree md0
md0
|-- lost+found

1 directory, 0 files
```

5. 修改mdadm配置文件

确定文件系统UUID：

```
[root@VM_63_126_centos ~]# mdadm --detail --scan
ARRAY /dev/md0 metadata=1.2 name=VM_63_126_centos:0 UUID=3c2adec2:14cf1fa7:999c29c5:7d739349
```

执行以下命令修改mdadm配置文件：

```
vi /etc/mdadm.conf
```

对于弹性云硬盘，建议写入以下配置：

```
DEVICE /dev/disk/by-id/virtio-□□□□1ID-part1
DEVICE /dev/disk/by-id/virtio-□□□□2ID-part1
DEVICE /dev/disk/by-id/virtio-□□□□3ID-part1
DEVICE /dev/disk/by-id/virtio-□□□□4ID-part1
ARRAY □□□□□□ metadata= UUID=
```

本例为：ARRAY /dev/md0 metadata=1.2 UUID=3c2adec2:14cf1fa7:999c29c5:7d739349

多块弹性云盘构建LVM逻辑卷

LVM (Logical Volume Manager , 逻辑卷管理) 通过在硬盘和分区之上建立一个逻辑层, 可以将磁盘或分区划分为相同大小的PE (Physical Extents) 单元, 不同的磁盘或分区可以划归到同一个卷组 (VG , Volume Group) , 在VG上可以创建逻辑卷 (LV , Logical Volume) , 在LV上可以创建文件系统。可以简单的把卷组与磁盘, 逻辑卷与分区概念对应起来。但是相对于直接使用磁盘分区的方式, LVM的重点在于弹性调整文件系统的容量:

- 文件系统不再受限于物理磁盘的大小, 可以分布在多个磁盘上: 比如您可以购买3个4TB的弹性云盘并使用LVM创建一个将近12TB的超大文件系统
- 可以动态调整逻辑卷大小, 不需要重新对磁盘重新分区: 当LVM卷组的空间无法满足您的需求时, 您可以单独购买弹性云盘并将其挂载在相应的云主机上, 然后参考下边的指引将其添加到LVM卷组中进行扩容操作
-

下面介绍如何使用三块腾讯云弹性云硬盘通过LVM创建可以动态调整大小的文件系统。

```
[root@VM_63_126_centos ~]# fdisk -l | grep vd | grep -v vda | grep -v vdb
Disk /dev/vdc: 10.7 GB, 10737418240 bytes
Disk /dev/vdd: 10.7 GB, 10737418240 bytes
Disk /dev/vde: 10.7 GB, 10737418240 bytes
```

1. 创建物理卷 (PV)

执行以下命令创建一个物理卷:

```
pvcreate □□□□1 ... □□□□N
```

```
[root@VM_63_126_centos ~]# pvcreate /dev/vdc /dev/vdd /dev/vde
Physical volume "/dev/vdc" successfully created
Physical volume "/dev/vdd" successfully created
Physical volume "/dev/vde" successfully created
```

执行 `pvscan`、`lvmdiskscan`、`pvs`、`pvdisplay` 物理卷路径等命令查看现在系统中的物理卷:

```
[root@VM_63_126_centos ~]# lvm diskscan | grep LVM
/dev/vdc [ 10.00 GiB] LVM physical volume
/dev/vdd [ 10.00 GiB] LVM physical volume
/dev/vde [ 10.00 GiB] LVM physical volume
3 LVM physical volume whole disks
0 LVM physical volumes
```

2. 创建卷组 (VG)

执行以下命令创建卷组：

```
vgcreate [-s 物理卷大小] 卷组名 物理卷路径
```

```
[root@VM_63_126_centos ~]# vgcreate lvm_demo0 /dev/vdc /dev/vdd
Volume group "lvm_demo0" successfully created
```

创建完成后可以使用vgextend 卷组名 新物理卷路径来向卷组中添加新的物理卷：

```
[root@VM_63_126_centos ~]# vgextend lvm_demo0 /dev/vdf
Volume group "lvm_demo0" successfully extended
```

使用vgs、vgdisplay等命令查看当前系统中的卷组：

```
[root@VM_63_126_centos ~]# vgs
VG          #PV #LV #SN Attr   VSize  VFree
lvm_demo    3   0   0 wz--n- 29.99g 29.99g
```

3. 创建逻辑卷 (LV)

创建出大卷组后，接下来可以开始建立分割区 (LV) 了，执行以下命令创建逻辑卷：

```
lvcreate [-L 逻辑卷大小] [ -n 逻辑卷名 ] VG名
```

```
[root@VM_63_126_centos ~]# lvcreate -L 8G -n lv_0 lvm_demo
Logical volume "lv_0" created
```

此时使用pvs命令可以发现只有vdc的PE被使用了：

```
[root@VM_63_126_centos ~]# pvs
PV          VG          Fmt  Attr PSize  PFree
/dev/vdc    lvm_demo    lvm2 a--  10.00g  2.00g
/dev/vdd    lvm_demo    lvm2 a--  10.00g 10.00g
/dev/vde    lvm_demo    lvm2 a--  10.00g 10.00g
```

4. 创建文件系统

执行以下命令在创建好的逻辑卷上创建文件系统：

```
mkfs
```

```
[root@VM_63_126_centos ~]# mkfs.ext3 /dev/lvm_demo/lv_0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
524288 inodes, 2097152 blocks
104857 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2147483648
64 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

使用mount命令挂载该文件系统：

```
[root@VM_63_126_centos ~]# mount /dev/lvm_demo/lv_0 vg0/
[root@VM_63_126_centos ~]# mount | grep lvm
/dev/mapper/lvm_demo-lv_0 on /root/vg0 type ext3 (rw)
```

5. 动态扩展逻辑卷及文件系统大小

当VG容量有剩余时，LV容量可动态扩展。执行以下命令扩展逻辑卷大小：

```
lvextend [-L +/- 容量] 容量
```

```
[root@VM_63_126_centos vg0]# lvextend -L +4G /dev/lvm_demo/lv_0
Size of logical volume lvm_demo/lv_0 changed from 8.00 GiB (2048 extents) to 12.00 GiB (3072 extents).
Logical volume lv_0 successfully resized
```

此时使用pvs命令可以发现vdc已被完全使用，vdd被使用了2G空间：

```
[root@VM_63_126_centos vg0]# pvs
PV          VG      Fmt Attr PSize  PFree
/dev/vdc    lvm_demo lvm2 a-- 10.00g  0
/dev/vdd    lvm_demo lvm2 a-- 10.00g  7.99g
/dev/vde    lvm_demo lvm2 a-- 10.00g 10.00g
```

此时只是扩展的逻辑卷的大小，在其之上的文件系统也要随之进行扩展才能使用，这里使用resize2fs来扩展文件系统大小：

```
[root@VM_63_126_centos vg0]# resize2fs /dev/lvm_demo/lv_0
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/lvm_demo/lv_0 is mounted on /root/vg0; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/lvm_demo/lv_0 to 3145728 (4k) blocks.
The filesystem on /dev/lvm_demo/lv_0 is now 3145728 blocks long.

[root@VM_63_126_centos vg0]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       7.9G 1019M  6.5G  14% /
/dev/mapper/lvm_demo-lv_0
                12G  549M  11G   5% /root/vg0
```