

Centos 7 磁盘阵列配置介绍

原创

高文龙

2018-03-14 19:28:21

评论(1)

1009人阅读

Centos 7 磁盘阵列配置介绍

每当我们提到磁盘阵列，相信广大管理员并不陌生，比如我们一般安装服务器系统的时候，先前条件是配置服务器的RAID信息，配置RAID信息可以提高服务器性能及数据的安全及稳定性，当然RAID分多少等级，最为常见的是RAID0、RAID1、RAID5、RAID1+0、RAID10等，每个RAID等级都会带来不同的效果，比如RAID0会将多个磁盘组合成一个大的磁盘空间，这样会大大增加磁盘空间的利用率，但是缺点就是只要坏一块磁盘就会导致所有数据丢失，再说说RAID1、RAID5等RAID等级对磁盘的数据有要求，同时磁盘空间利用率没有RAID0那么高，比如RAID1当配置后会将磁盘一半做为磁盘镜像，保证数据的可用性，RAID1最少两块磁盘，比如500G的两块磁盘配置RAID1，可用空间约等于500G，会将另外500G作为磁盘镜像，比如RAID5最少需要三块磁盘，会将其中一块磁盘作为镜像，所以可以根据自己的需求来进行配置。下面我们大概介绍一下RAID等级介绍：

RAID0 是一种简单的、无数据校验的数据条带化技术。实际上不是一种真正的 RAID，因为它并不提供任何形式的冗余策略。RAID0 将所在磁盘条带化后组成大容量的存储空间，将数据分散存储在所有磁盘中，以独立访问方式实现多块磁盘的并读访问。由于可以并发执行 I/O 操作，总线带宽得到充分利用。再加上不需要进行数据校验，RAID0 的性能在所有 RAID 等级中是最高的。理论上讲，一个由 n 块磁盘组成的 RAID0，它的读写性能是单个磁盘性能的 n 倍，但由于总线带宽等多种因素的限制，实际的性能提升低于理论值。RAID0 具有低成本、高读写性能、100% 的高存储空间利用率等优点，但是它不提供数据冗余保护，一旦数据损坏，将无法恢复。因此，RAID0 一般适用于对性能要求严格但对数据安全性和可靠性不高的应用，如视频、音频存储、临时数据缓存空间等。

RAID1 称为镜像，它将数据完全一致地分别写到工作磁盘和镜像 磁盘，它的磁盘空间利用率为 50%。RAID1 在数据写入时，响应时间会有所影响，但是读数据的时候没有影响。RAID1 提供了最佳的数据保护，一旦工作磁盘发生故障，系统自动从镜像磁盘读取数据，不会影响用户工作。工作原理如图 3 所示。

RAID1 与 RAID0 刚好相反，是为了增强数据安全性使两块 磁盘数据呈现完全镜像，从而达到安全性好、技术简单、管理方便。RAID1 拥有完全容错的能力，但实现成本高。RAID1 应用于对顺序读写性能要求高以及对数据保护极为重视的应用，如对邮件系统的数据保护。

RAID5 应该是目前最常见的 RAID 等级，它的原理与 RAID4 相似，区别在于校验数据分布在阵列中的所有磁盘上，而没有采用专门的校验磁盘。对于数据和校验数据，它们的写操作可以同时发生在完全不同的磁盘上。因此，RAID5 不存在 RAID4 中的并发写操作时的校验盘性能瓶颈

问题。另外，RAID5 还具备很好的扩展性。当阵列磁盘 数量增加时，并行操作量的能力也随之增长，可比 RAID4 支持更多的磁盘，从而拥有更高的容量以及更高的性能。

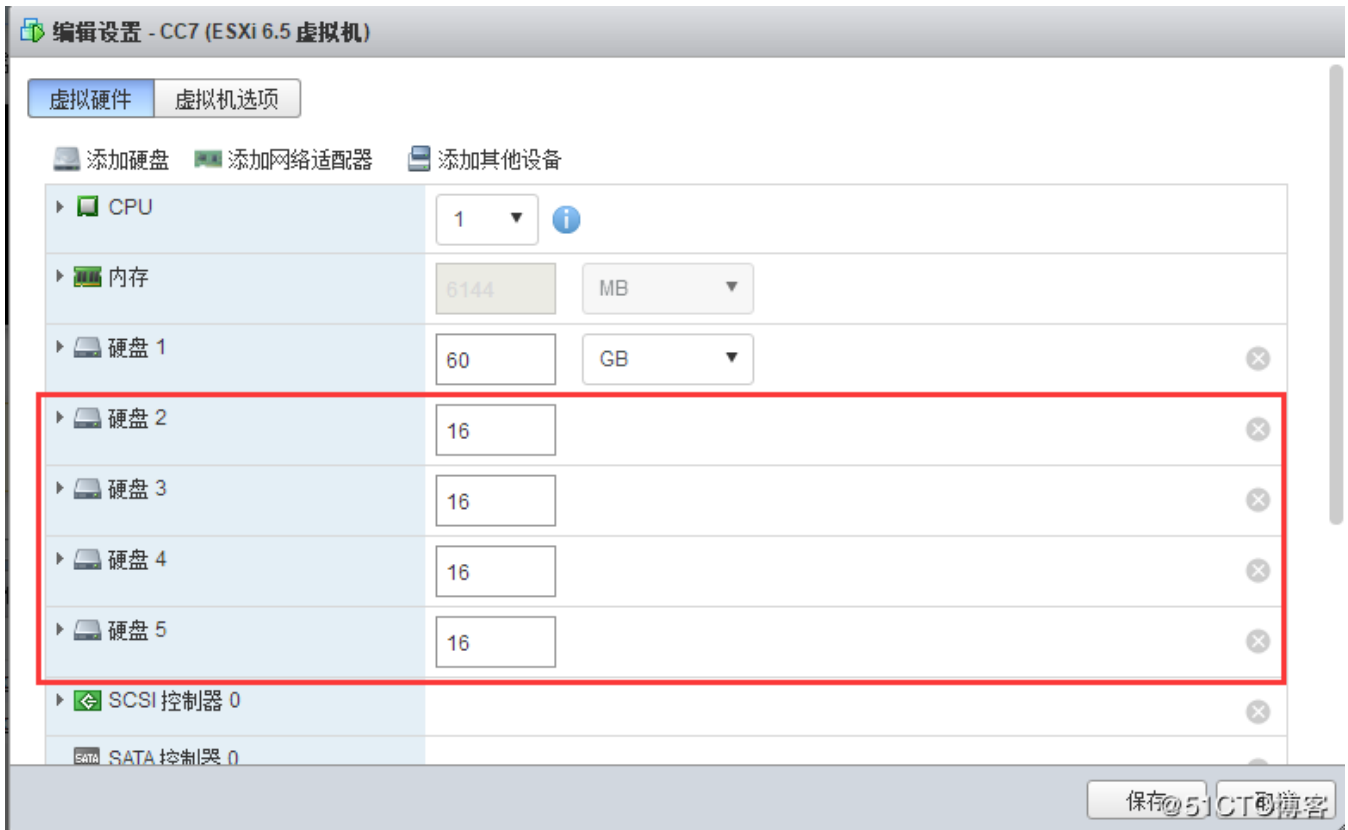
RAID5 兼顾存储性能、数据安全和存储成本等各方面因素，它可以理解为 RAID0 和 RAID1 的折中方案，是目前综合性能最佳的数据保护解决方案。RAID5 基本上可以满足大部分的存储应用需求，数据中心大多采用它作为应用数据的保护方案。

RAID 等级	RAID0	RAID1	RAID3	RAID5	RAID6	RAID10
别名	条带	镜像	专用奇偶校验条带	分布奇偶校验条带	双重奇偶校验条带	镜像加条带
容错性	无	有	有	有	有	有
冗余类型	无	有	有	有	有	有
热备份选择	无	有	有	有	有	有
读性能	高	低	高	高	高	高
随机写性能	高	低	低	一般	低	一般
连续写性能	高	低	低	低	低	一般
需要磁盘数	$n \geq 1$	$2n$ ($n \geq 1$)	$n \geq 3$	$n \geq 3$	$n \geq 4$	$2n(n \geq 2) \geq 4$
可用容量	全部	50%	$(n-1)/n$	$(n-1)/n$	$(n-2)/n$	50%

@51CTO博客

言归正传，我们今天主要介绍在Linux系统下如何配置磁盘阵列。具体见下：

我们今天在Centos7下做介绍，我们首先给系统增加4块磁盘



然后通过fdisk -查看磁盘信息

```
[root@oa-nginx ~]# fdisk -l
```

```
Disk /dev/sda: 64.4 GB, 64424509440 bytes, 125829120 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk label type: dos
```

```
Disk identifier: 0x0000f635
```

```
Device Boot Start End Blocks Id System
```

```
/dev/sda1 * 2048 2099199 1048576 83 Linux
```

```
/dev/sda2 2099200 125829119 61864960 8e Linux LVM
```

```
Disk /dev/mapper/cl-root: 38.3 GB, 38323355648 bytes, 74850304 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk /dev/mapper/cl-swap: 6308 MB, 6308233216 bytes, 12320768 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Disk /dev/mapper/cl-home: 18.7 GB, 18710790144 bytes, 36544512 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdb: 17.2 GB, 17179869184 bytes, 33554432 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdc: 17.2 GB, 17179869184 bytes, 33554432 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdd: 17.2 GB, 17179869184 bytes, 33554432 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sde: 17.2 GB, 17179869184 bytes, 33554432 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

磁盘准备好后，我们准备配置磁盘阵列，因为linux下配置磁盘阵列需要使用mdadm命令，所以我们需要安装

我们首先安装mdadm工具

```
yum install -y mdadm
```

```

Package Arch Version Repos
Installing:
mdadm x86_64 4.0-5.el7 base
Installing for dependencies:
libreport-fs x86_64 2.1.11-38.el7.centos base
Transaction Summary
Install 1 Package (+1 Dependent package)

Total download size: 467 k
Installed size: 1.0 M
Downloading packages:
(1/2): libreport-fs-2.1.11-38.el7.centos.x86_64.rpm
(2/2): mdadm-4.0-5.el7.x86_64.rpm
-----
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : libreport-fs-2.1.11-38.el7.centos.x86_64
Installing : mdadm-4.0-5.el7.x86_64
Verifying : libreport-fs-2.1.11-38.el7.centos.x86_64
Verifying : mdadm-4.0-5.el7.x86_64

Installed:
mdadm.x86_64 0:4.0-5.el7

Dependency Installed:
libreport-fs.x86_64 0:2.1.11-38.el7.centos

```

@51CTO博客

接着我们介绍一下命令---mdadm参数

- a 检测设备名称
- n 指定设备数量
- l 指定RAID等级
- C 创建
- v 显示过程
- f 模拟设备损坏
- r 移除设备
- a 增加设备
- Q 查看摘要信息
- D 查看详细信息
- S 停止

我们使用mdadm命令创建RAID10名称为:dev/md0:

C代表创建操作, -v显示创建过程, -a yes检查RAID没名称, -n是用到硬盘个数, -l定义RAID的级别而后面写上要增加整列的硬盘名称

```
mdadm -Cv /dev/md0 -a yes -n 4 -l 10 /dev/sdb /dev/sdc /dev/sdd /dev/sde
```

```
[root@oa-nginx ~]# mdadm -Cv /dev/md0 -a yes -n 4 -l 10 /dev/sdb /dev/sdc /dev/sdd /dev/sde
mdadm: layout defaults to n2
mdadm: layout defaults to n2
mdadm: chunk size defaults to 512K
mdadm: /dev/sdb appears to be part of a raid array:
        level=raid10 devices=5 ctime=Wed Mar 14 14:49:39 2018
mdadm: /dev/sdc appears to be part of a raid array:
        level=raid10 devices=5 ctime=Wed Mar 14 14:49:39 2018
mdadm: /dev/sdd appears to be part of a raid array:
        level=raid10 devices=5 ctime=Wed Mar 14 14:49:39 2018
mdadm: /dev/sde appears to be part of a raid array:
        level=raid10 devices=5 ctime=Wed Mar 14 14:49:39 2018
mdadm: size set to 16760832K
Continue creating array? █
```

@51CTO博客

格式化RAID磁盘整理为ext4格式

```
mkfs.ext4 /dev/md0
```

```
mdadm: array /dev/md0 started.
[root@oa-nginx ~]# mkfs.ext4 /dev/md0
mke2fs 1.42.9 (28-Dec-2013)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=256 blocks
2097152 inodes, 8380416 blocks
419020 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2155872256
256 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

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

@51CTO博客

创建挂载目录

```
mkdir /RAID
```

```
[root@oa-nginx ~]# mkdir /RAID
```

@51CTO博客

```
mount /dev/md0 /RAID
```

```
[root@oa-nginx ~]# mount /dev/md0 /RAID
```

@51CTO博客

```
[root@oa-nginx ~]#
```

```
df -h
```

```
[root@oa-nginx ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/cl-root 36G  1.2G   35G   4% /
devtmpfs        2.9G     0   2.9G   0% /dev
tmpfs           2.9G     0   2.9G   0% /dev/shm
tmpfs           2.9G  8.6M   2.9G   1% /run
tmpfs           2.9G     0   2.9G   0% /sys/fs/cgroup
/dev/sda1       1014M  139M   876M  14% /boot
/dev/mapper/cl-home 18G   33M   18G   1% /home
tmpfs          581M     0   581M   0% /run/user/0
/dev/md0        32G   49M   30G   1% /RAID
```

@51CTO博客

设置为系统重启也自动生效

```
echo "/dev/md0 /RAID ext4 default 0 0" >> /etc/fstab
```

```
[root@oa-nginx ~]# echo "/dev/md0 /RAID ext4 default 0 0" >> /etc/fstab
[root@oa-nginx ~]# cat /etc/fstab
```

```
#
# /etc/fstab
# Created by anaconda on Thu Feb 22 15:00:53 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/cl-root    /                    xfs     defaults        0 0
UUID=8a26e06d-e8a0-4453-a60a-18f9f8a703b7 /boot               xfs     defaults        0 0
/dev/mapper/cl-home    /home               xfs     defaults        0 0
/dev/mapper/cl-swap     swap                swap    defaults        0 0
/dev/md0 /RAID ext4 default 0 0
```

@51CTO博客

查看/dev/md0设备信息

```
mdadm -D /dev/md0
```

```
/dev/md0 /RAID ext4 default 0 0
[root@oa-nginx ~]# mdadm -D /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Wed Mar 14 14:59:09 2018
    Raid Level : raid10
    Array Size : 33521664 (31.97 GiB 34.33 GB)
  Used Dev Size : 16760832 (15.98 GiB 17.16 GB)
    Raid Devices : 4
    Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Wed Mar 14 15:02:36 2018
      State : clean
 Active Devices : 4
Working Devices : 4
 Failed Devices : 0
 Spare Devices : 0


    Layout : near=2
   Chunk Size : 512K

Consistency Policy : unknown

        Name : oa-nginx:0 (local to host oa-nginx)
        UUID : baa28a59:ae5ccd44:3f348032:c682ad8a
        Events : 19

   Number   Major   Minor   RaidDevice State
     0         8        16         0   active sync set-A   /dev/sdb
     1         8        32         1   active sync set-B   /dev/sdc
     2         8        48         2   active sync set-A   /dev/sdd
     3         8        64         3   active sync set-B   /dev/sde
```

@51CTO博客

接下来我们模拟一块硬盘损坏

```
mdadm /dev/md0 -f /dev/sdb
```

```
[root@oa-nginx ~]# mdadm /dev/md0 -f /dev/sdb
mdadm: set /dev/sdb faulty in /dev/md0
[root@oa-nginx ~]#
```

@51CTO博客

```
mdadm -D /dev/md0
```



```
[root@oa-nginx ~]# mdadm -D /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Wed Mar 14 14:59:09 2018
    Raid Level : raid10
    Array Size : 33521664 (31.97 GiB 34.33 GB)
  Used Dev Size : 16760832 (15.98 GiB 17.16 GB)
    Raid Devices : 4
  Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Wed Mar 14 15:23:04 2018
      State : clean, degraded
  Active Devices : 3
 Working Devices : 3
 Failed Devices : 1
  Spare Devices : 0

    Layout : near=2
   Chunk Size : 512K

Consistency Policy : unknown

        Name : oa-nginx:0 (local to host oa-nginx)
        UUID : baa28a59:ae5ccd44:3f348032:c682ad8a
        Events : 21

   Number   Major   Minor   RaidDevice State
    -   -   -   -   -   -
     0         0         0         0   removed
     1         8         32         1   active sync set-B   /dev/sdc
     2         8         48         2   active sync set-A   /dev/sdd
     3         8         64         3   active sync set-B   /dev/sde
     0         8         16         -   faulty   /dev/sdb
```

@51CTO博客

损坏后依然可以正常使用

因为RAID10允许一组RAID1硬盘中存在一个故障而不影响使用，所以我们任然可以使用正常点的创建或者删除文件，现在就把新的硬盘增加进去吧，当然也可以让硬盘sdb恢复使用,重启后执行以下命令

```
mdadm /dev/md0 -a /dev/sdb
```

我们重启系统后发现无法启动


```

CC7
0.997311] sd 0:0:0:0: [sda] Assuming drive cache: write through
0.998909] sd 0:0:1:0: [sdb] Assuming drive cache: write through
1.002472] sd 0:0:2:0: [sdc] Assuming drive cache: write through
1.005435] sd 0:0:3:0: [sdd] Assuming drive cache: write through
1.007448] sd 0:0:4:0: [sde] Assuming drive cache: write through
2.958044] piix4_smbus 0000:00:07.3: Host SMBus controller not enabled!
3.541465] EXT4-fs (md0): Unrecognized mount option "default" or missing value
Welcome to emergency mode! After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or ^D to
try again to boot into default mode.
Give root password for maintenance
(or type Control-D to continue):

```

@51CTO博客

自动挂载的那个fstab文件有问题，你在这个界面直接输入密码，然后把你增加的删除，重启就OK

注释后，保存重启

```

CC7
#
# /etc/fstab
# Created by anaconda on Thu Feb 22 15:00:53 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/cl-root    /                    xfs     defaults        0 0
UUID=8a26e06d-e8a0-4453-a60a-18f9f8a703b7 /boot                xfs     defaults        0 0
/dev/mapper/cl-home    /home               xfs     defaults        0 0
/dev/mapper/cl-swap     swap                swap    defaults        0 0
# /dev/md0 /RAID ext4 default 0 0

```

@51CTO博客

重启后恢复正常

```
CC7
CentOS Linux 7 (Core)
Kernel 3.10.0-514.el7.x86_64 on an x86_64

oa-nginx login: _

@51CTO博客
```

我们再次执行上面的操作

```
mdadm /dev/md0 -a /dev/sdb
```

```
[root@oa-nginx ~]# mdadm /dev/md0 -a /dev/sdb
mdadm: added /dev/sdb
[root@oa-nginx ~]#
```

@51CTO博客

我们再次查看

```
mdadm -D /dev/md0
```

```
[root@oa-nginx ~]# mdadm -D /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Wed Mar 14 14:59:09 2018
    Raid Level : raid10
    Array Size : 33521664 (31.97 GiB 34.33 GB)
  Used Dev Size : 16760832 (15.98 GiB 17.16 GB)
    Raid Devices : 4
    Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Wed Mar 14 15:41:19 2018
      State : clean, degraded, recovering
  Active Devices : 3
 Working Devices : 4
 Failed Devices : 0
  Spare Devices : 1

    Layout : near=2
   Chunk Size : 512K

Consistency Policy : unknown

Rebuild Status : 15% complete

    Name : oa-nginx:0 (local to host oa-nginx)
    UUID : baa28a59:ae5ccd44:3f348032:c682ad8a
    Events : 27

   Number Major Minor RaidDevice State
     4       8      16         0  spare rebuilding /dev/sdb
     1       8      32         1  active sync set-B /dev/sdc
     2       8      48         2  active sync set-A /dev/sdd
     3       8      64         3  active sync set-B /dev/sde

[root@oa-nginx ~]#
```

@51CTO博客

版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任