LINUX FOUNDATION

TRAINING

Assembling Partitions as RAID Devices

A Linux Foundation Training Publication
www.training.linuxfoundation.org

Overview

A Redundant Array of Independent Disks (RAID) is the practice of treating several hard disks as a single disk in order to provide improved performance, reliability, availability, and capacity. Data is distributed across the combined disks in a way that optimizes for the desired combination of these improvements. The data distribution choices are known as RAID levels; there are 4 commonly used ones, and a number of proprietary and non-standard levels as well.

There are full hardware raid controllers that present the operating system with an assembled, configured RAID device. There are also what are known as hardware assisted software RAID controllers, which use a standard device controller chip combined with proprietary firmware drivers to present the operating system with a RAID device. Linux has tools for a full software RAID implementation, which gives additional flexibility over hardware and hardware assisted implementations.

The size of a RAID device is limited by the smallest constituent partition within it. Note that while you can create a RAID device using partitions of the same disk, that would defeat the purpose of RAID.

Key Ideas

RAID: the practice of combining multiple disk partitions into a single logical disk.

RAID levels: ways of distributing data across a RAID disk device to optimize for data redundancy or read / write performance.

Striping: Purposely storing logically sequential data on separate disks to increase throughput by taking advantage of concurrent read and write operations.

Mirroring: Storing copies of the same data at the same location on multiple disks. Provides data redundancy in the case of disk failure, and an improved performance in read operations, whereas write performance will be roughly equal to the write performance to a single disk.

Parity: A mathematical way of providing redundancy without mirroring data. Cheaper redundancy, as less disks are required.

RAID 0: minimum 2 partitions, data is distributed evenly between the disks (striped), capacity of the disks are added, no redundancy. This level gives improved read / write performance, and use useful for storing application data when performance is more important than fault tolerance. The storage capacity of this RAID array is the sum of the individual capacities of the disks involved.

RAID 1: minimum 2 disks, even number required. Data is mirrored between the disks. The RAID device has the capacity of the smallest constituent partition. Useful for operating system partitions, or other data where the ability to recover is the most important consideration.

RAID 5: minimum 3 disks. Data is striped, parity information is distributed across the disks. The RAID device has the capacity of the total number of disks involved minus one, as a disk's worth of parity information is distributed across the remaining capacity. Useful for serving files. However, this configuration suffer from slow rebuild times and possible rebuild errors.

RAID 1+0: Data is striped across mirrored drives. The RAID device has the capacity of all the mirrored sets combined (i.e. 4 1TB disks = 2 mirrored 1TB RAID devices, total capacity is 2TB). Useful for database servers and other applications that rely on fast read / write operations.

mdadm: The mdadm utility is used in Linux to combine multiple partitions into RAID devices.

Init system: A running Linux operating system is made up of a large number of applications and services. The init system manages the order in which those services are started at boot time. Most operating systems are standardising on the SystemD init system. SysVinit is another popular init system.

Example Scenario

Create a mirrored raid device using two partitions, and make sure the operating system is configured to start the mdadm service automatically on boot. You will need at least two partitions prepared to complete this exercise.

Now Do It

1. Make sure the mdadm utility is installed using your operating system's package manager package manager.
2. Use the mdadm command to create a mirrored RAID device using two partitions.
3. Check the status of your newly created RAID device using the mdadm command.
4. The mdadm utility has a separate monitoring service called mdmonitor. Configuring the mdmonitor service to monitor the array you created requires information provided in a scan of the device.
5. Add the output of the mdadm --detail --scan command to the mdadm.conf file, often located at at /etc/mdadm.conf or /etc/mdadm/mdadm.conf. If the configuration file doesn't exist, create it, and additionally insert an email address to send alerts to ("MAILADDR root", without quotation marks, is a good default).
6. Configure the operating system to start the monitoring service at boot time.

If you remember nothing else...

RAID devices are about making tradeoffs between performance and redundancy that are appropriate to your use case. The bigger the RAID device, the longer it takes to rebuild in case of disk failure. Using a RAID device is not the same thing as having a comprehensive backup strategy.

Answer Key

1. # yum install -y mdadm
    or
    # apt-get install mdadm

2. # mdadm --create --verbose /dev/md0 --level=mirror --raid-devices=2 /dev/sdb1 /dev/sdc1
mdadm: size set to 96128K
    mdadm: Defaulting to version 1.2 metadata
    mdadm: array /dev/md0 started.
    Note that if the partitions you are using contain file systems, you are prompted to confirm your intention to use them in a RAID device.

3. # mdadm --detail /dev/md0
    /dev/md0:
    Version : 1.2
    ...

Number	Major	Minor	RaidDevice	State	
0	8	17	0	active sync	/dev/sdb1
1	8	33	1	active sync	/dev/sdc1

4. # mdadm --detail --scan
    ARRAY /dev/md/0 metadata=1.2 name=localhost.localdomain:0 UUID=5fd97b87:18209d2d:ac07e389:4fb93cfe

5. mdadm --detail --scan >> /etc/mdadm/mdadm.conf

6. Operating systems using SystemD:
    # systemctl start mdmonitor (or alternatively, mdmonitor.service)
    # systemctl enable mdmonitor
    Operating systems using SysVinit:
    # service mdmonitor start
    # chkconfig mdmonitor on

LINUX FOUNDATION

Get Certified!

Get more information at http://training.linuxfoundation.org/certification/lfcs