ATOL: Filesystems and Their Management

Marek Grác
xgrac@fi.muni.cz

Red Hat Czech s.r.o. / Faculty of Informatics, Masaryk University

Advanced Topics of Linux Administration

Creating partiotions

- *fdisk* ($<$ 1.5 TB), *cfdisk*, *parted* – view and manage partition tables
- List partition tables from command line
- *partprobe* – inform the OS of partition table changes
- *cat /proc/partitions*

Making Filesystems

- *mkfs*
- *mkfs.ext2*, *mkfs.ext3*, *mkfs.msdos*
- Specific filesystem utilities can be called directly
    - *mke2fs [options] device*

Filesystem Labels

- Alternate way to refer to devices
- Device independent
  - *e2label devfile [fslabel]*
  - *mount [options] LABEL=fslabel mountpoint*
- *blkid* – used to see labels and filesystems type of all devices

Mount Points and /etc/fstab

- Configuration of the filesystem hierarchy
- Used by *mount*, *fsck* and other programs
- Maintains the hierarchy between system reboots
- May use filesystem volume labels in the device field
- The *mount -a* command cen be used to mount all filesystems listed in `/etc/fstab`

Unmounting Filesystems

- *umount [options] device|mountpoint*
- You cannot unmount a filesystem that is in use
  - Use *fuser* to check and/or kill processes
- Use the *remount* option to change a mounted filesystem's options atomically
  - *mount -o remount,ro /data*

Handling Swap Files and Partitions

- Swap space is a suppplement to system RAM
- Basic setup involves:
  - Create a swap partition or file
  - Write special signature using *mkswap*
  - Add appropriate entries to `/etc/fstab`
  - Activate swap space with *swapon -a*

Filesystems in Linux

- Local disk file systems
  - ext2, ext3, ext4
  - reiserFS
  - XFS
- Shared disk file systems (SAN vs NAS, cluster)
  - GFS, GFS2
  - GPFS
  - Lustre

Slide by Pavol Babinčák

Extended file system (ext2, ext3)

- Designed for Linux
- ext2
  - Very stable
  - Through faultcan hurt filesystem
  - Repair is easy but quite slow
  - Inode size <= 128 for Windows driver
- ext3 = ext2 + journaling
  - Backwards compatibility with ext2
  - Repair is fast (?) but some metadata operations are slow
  - Immutable files and append-only files

Slide by Pavol Babinčák

ReiserFS

- ReiserFS3 in vanilla Linux kernel. Reiser4 not ready for enterprise.
- Reiser3
  - Good for small files
  - Not so stable
  - Less users, less support
- Reiser4
  - Plugin driven filesystems
  - Transactions

Slide by Pavol Babinčák

XFS

- One of the first journaling fs under UNIX (kernel 2.4.X)
  - Good for large files, big directories, big filesystems
  - Slow and problematic repair
  - Creation/Deletion of directory entries are slow
  - Quota can be set on per directory base
- Features in XFS
  - Delayed allocation for reducing fragmentation
  - Native backup/restore utilities able to make fs dump without unmounting

Slide by Pavol Babinčák

GFS

- GFS2 is available in vanilla kernel since 2.6.19
  - Cluster filesystem
  - All nodes are equal, running are controlling access to shared resources
  - Failure cluster member affects only other members using shared resources
- Features in GFS2:
  - Direct I/O support allows databases to achieve high performance
  - Dynamic multi-path routing around failed components in SAN

Slide by Pavol Babinčák

GPFS

- Proprietary, generally bundled with IBM hardware
- Used on very large clusters (up to 2000 nodes)
- High performance and grids
- Features in GPFS:
  - SQL based syntax policies for file placement and management
  - Shared disk or network block IO configuration
  - Offer clustered NFS (HA)
  - Snapshot by copy-on-write

Slide by Pavol Babinčák

Lustre

- Not part of vanilla kernel, only patches
- Architecture:
  - Uses modified ext3 as storage fs
  - Single metadata target
  - typicaly 2-8 object storage servers
  - clients accessing data
- Features in Lustre:
  - Support for HA, recovery, transparent reboots
  - Data blocks stripped across objects (bandwidth agregation, not limited by size of target object)

Slide by Pavol Babinčák

Software RAID Configuration

- Create and define RAID devices using *mdadm*
  - *mdadm -C /dev/md0 -a yes -l 1 -n 2 -x 1 elements*
- Format each RAID device with a filesystem
  - *mke2fs -k /dev/md0*
- Test the RAID devices
- allows to check the status of your RAID devices
  - *mdadm –detail /dev/md0*

Software RAID Testing and Recovery

- Simulating disk failures
  - *mdadm /dev/md0 -f /dev/sda1*
- Recovering from a software RAID disk failure
  - replace the failed hard drive and power on
  - reconstruct partitions on the replacement drive
  - *mdadm /dev/md0 -a /dev/sda1*
- *mdadm*, `/proc/mdstat` and syslog messages

What is Logical Volume Manager?

- A layer of abstraction that allows easy manipulation of volumes. Including resizing of filesystems.
- Allow reorganization of filesystems across multiple physical devices
  - Devices are designated as Physical Volumes (PV)
  - One or more PV are used to create a Volume Group (VG)
  - PV are defined with Physical Extents of a fixed size
  - Logical Volumes (LV) are created on PV and are composed of Physical Extents
  - Filesystems may be created on Logical Volumes

Creating Logical Volumes

- Create physical volumes
  - *pvcreate /dev/sda3*
- Assign physical volumes to volume groups
  - *vgcreate vg0 /dev/sda3*
- Create logical volumes from volume groups
  - *lvcreate -L 256M -n data vg0*
  - *mke2fs -j /dev/vg0/data*

Resizing Logical Volumes

- Growing Volumes
  - *lvextend* can grow logical volumes
  - *resize2fs* can grow EXT3 filesystems online
  - *vgextend* adds new physical volumes to an existing volume group
- Shrinking Volumes
  - Filesystem have to be reduced first
  - Requires a filesystem check and cannot be performed online
  - *lvreduce* can then reduce volume
- Volume Groups can be reduced with:
  - *pvmove /dev/sda3*
  - *vgreduce vg0 /dev/sda3*

Lab: Installation

- Goals:
  - Deploy LVM on the software RAID device
  - Create a group with two partitions such that new partition could be added, and the filesystem could be extended

Lab: Prepare a paper

- Themes:
  - Compare software and hardware RAID
  - Compare new filesystems in Linux (ext4, zfs, reiser4, . . .)
- Format:
  - Short presentation (15–20 minutes; 5-7 slides)
  - Paper containing comparision (1000 words)