

# Hot Upgrading a RAID6 array

St. Louis Unix User's Group  
12 May 2021

Lee Lammert  
Omnitec Corporation

# Problem – space low

- Currently 12T RAID6 Array
  - 6x2T
  - LVM
  - Current partition structure 6x2T:

Device	Size	Type	
/dev/sda1	509M	Microsoft basic data	<i>boot</i>
/dev/sda2	2G	Linux RAID	<i>SWAP</i>
/dev/sda3	100G	Linux RAID	<i>System</i>
/dev/sda4	1.8T	Linux RAID	<i>Data (/u)</i>

- Six 1.8T, RAID6 -> 4x1.8T usable -> 7.1T usable
- 7.1T usable, 5.5T used, upgrade required

# Replacing drives in a softRAID

- A softraid disk can be swapped as follows:
  - Fail the disk
  - Remove disk
  - Replace disk
  - Recreate partition table (copy from another disk)
  - Add new drives to array(s)
- Simple, eh?

# Project Tasks

- Save the current backup jobs to mitigate risk when upgrading the current LVM
- *NOTE: It would be possible to use a 10T external drive, but PCIe is faster than USB3, and much faster than USB2*
- Use two new 4T drives and LVM to create 8T of temp storage
- Move all backup jobs there
- Replace each 2T drive with a 4T drive
- Move backup jobs back to new array
- Add extra 4T drives to the main array

# Risks

- Original 6x2T array was created 10+ years ago with LVM to preserve expansion option
- **Bad idea! Why?**
- LVM is a single point of failure. Rebuilding the underlying array disk-by-disk is fairly safe, but what happens if the LVM config has a problem?
- Need to address that risk

# Mitigate the LVM risk

- Backups are DR for clients, and a LVM is a **single point of failiure**, so must minimize risk of **any** data loss
- Solution:
  - There are two blank slots in the chassis, ..
  - Add 2x 4T drives as LVM to hold DR backups during the process!
  - When finished, reformat those 4T drives and add to original array
- Start with a new two port PCIe controller
  - **Be careful, it will jumble the drive IDs!**

# Create the transfer 8T Transfer LVM

- New drives: /dev/sde, /dev/sdf
- pvcreate /dev/sde
- pvcreate /dev/sdf
- vgcreate vg-temp /dev/sde /dev/sdf
- lvcreate vg-temp -n lv-Temp
- mkfs -t xfs /dev/vg-temp/lv-Temp

# Move the backup jobs

- Stop one backup job temporarily (comment out in crontab)
- `rsync -avh <old array> <new array>`
  - ^ **Required to move hard links**
- Update rsnapshot configuration with new location
- Restart that backup
- Repeat until complete (~two weeks for ours)
- No backup job misses more than one or possibly two incrementals



# Moving a backup job demo

- Disable in crontab
- Move the data
- Update .conf
- Restart job
- Repeat

# Replace the 2T drives

- Remove a partition from md\_  
mdadm --manage /dev/md0 --fail /dev/sdg2
- Check with mdstat  
cat /proc/mdstat
- Swap the drive, replace the partition table
- Add a partition to md\_  
mmdadm --add /dev/md\_ /dev/sd\_\_
- Repeat for ***all*** partitions in an array
- Repeat for all six 2T drives

# Final tasks

- Rebuild boot sector (/dev/sda)
- Copy entire partition 1 (including boot sector) from sda to sdb (and maybe sdc) in case of a disk failure that would lose the grub2 boot loader
- It is possible to rebuild a bootable disk by starting with a thumb drive, chroot'ing and mounting the existing system, and building a new grub boot on a different drive
- Reboot with a thumb drive to resize the md arrays (must be unmounted for xfs)

# Free the 2x4T new drives

- Unmount the temporary logical volume
- Remove the fstab entry
- Disable the logical volume  
    `lvchange -an /dev/vg-temp/lv-Temp`
- Verify with `lvscan`  
    `lvscan`
- Remove the logical volume  
    `lvremove /dev/vol_grp/log_grp1`

# Add 2x4T to new 15T array

- Recreate partition table by copying from another drive  
Open with fdisk and write to create backup MBR
- Add partitions to array(s)  
`mdadm --add /dev/md_ /dev/sd__`
- Grow the RAID array with the new partitions  
`mdadm --grow --raid-devices=_ /dev/md_`
- Next, grow the file system to match  
`mdadm --grow --raid-devices=8 /dev/md1`  
`mdadm --grow --raid-devices=8 /dev/md2`
- Finally, grow the filesystem  
`xfs_growfs __ (mount point)`

Happy Rebuilding!

Thanks!!

Lee Lammert  
*lvl@omnitec.net*

**OMNITEC** *Corporation*