Migrating from Linux to FreeBSD as a backend data store

Dwayne Hart
dwayne.hart@gmail.com
dwayne@mun.ca

Background

Roughly two years ago I was fortunate enough to rejoin the CS/Math & Stats systems group. The Math & Stats department contained three research and one departmental file servers.

The departmental file server was being backed up by two remote servers. While the research file servers had their content copied to at least one remote system containing equal or less disk space. Providing a 24 hour backup window.

Research/Backup File Server Specs

Manufacturer: IBM

Model: x3500

CPU: Xeon 5350 @ 2.0GHz (2 cores)

Memory: 4GB

RAID card: IBM ServeRAID 8k

Drives: 8 x 500GB Seagate SATA II

Hard Drive Layout

Software RAID configuration: the OS partitions were installed on two RAID-1 volumes and the data was stored on two RAID-5 volumes.

```
prompt ~ # cat /proc/mdstat
Personalities : [linear] [raid0] [raid1] [raid10] [raid6] [raid5] [raid4] [multipath]
md0 : active raid1 sdb1[1] sda1[0]
    14648128 blocks [2/2] [UU]
md1 : active raid1 sdb6[1] sda6[0]
    15624896 blocks [2/2] [UU]
md2 : active raid6 sdg1[4] sdh1[5] sdc1[0] sdd1[6] sdf1[3] sde1[2]
    153466880 blocks super 1.2 level 6, 512k chunk, algorithm 2 [6/6] [UUUUUU]
md3 : active raid6 sdc2[2] sdh2[7] sdf2[5] sdg2[6] sda2[0] sde2[4] sdd2[9] sdb2[8]
    2695305216 blocks super 1.2 level 6, 512k chunk, algorithm 2 [8/8] [UUUUUUUU]
```

OS Layout

Both servers were setup to run a 32-bit Gentoo Linux variant with the following partition information.

```
prompt ~# df -h -t ext3
Filesystem            Size  Used Avail Use% Mounted on
/dev/md0               14G  9.7G  3.4G  75% /
/dev/md1               15G  1.6G   13G  12% /var
/dev/md2              145G  132G  5.5G  97%
/mnt/research/fs1
/dev/md3              2.5T  1.1T  1.3T  46%
/mnt/research/fs2
```

Issues?

Due to the physical layout of the software RAID instances, if a drive failed, it was was easy to physically replace and to schedule a maintenance window.

However, when the drive was replaced. We first had to configure the drive via the RAID card utility to be a JBOD, reboot and insert the new drive into the RAID volume. The system became unresponsive when users tried to query or engage the file server to launch new software instances.

Moving away from 32-bit to 64-bit systems. We’re not able to change an OS sysctl flag, reboot the system to magically migrate the entire system from one mode to another.

We’d need to boot the system as a diskless Linux client. Perform a bare metal OS install, ensure the data volumes were maintained and bring the system back online on its own disks.

Working Solution

Instead of going with a traditional disk full system. I elected to go with a diskless FreeBSD implementation which would allow me to leverage all the installed disks as one large ZFSv2 data volume.

I then utilized istgt (installed from ports) to offer a large disk to one of the group’s 64-bit Gentoo Linux compute nodes. At which point I could then leverage autofs and IPSec via racoon (ie. secure NFS implementation).

The driving force behind using diskless FreeBSD was to allow the research group to be able to expand their data store through the acquisition of additional storage gear.

The new server(s) could then be booted as diskless clients and its disks configured as a ZFS data volume; leveraging istgt an additional disk(s) would then be offered to the 64-bit Gentoo Linux file server/compute node.

Note that the 500GB disks have been replaced with 1TB disks.

FreeBSD Server

The FreeBSD server is setup to run an ISC-DHCP instance, TFTP and NFS. Please note that this has been discussed previously at BSDCan 2013 in a talk titled “Tales from the North: System Administration of a Geographically Disperse Network”.

I would like to point out that the primary FreeBSD server located in Math & Stats department is currently running FreeBSD 9.2 amd64 in a Citrix Xenserver instance with a single virtual disk. Whereas the server located in Computer Science is running on a Dell 745 OptiPlex workstation with a GMIRROR configured OS disk store.

By utilizing a directory structure similar to /diskless/fbsd92-amd64. We’re able to install additional versions of FreeBSD that we wish to run.

Once we update the ISC-DHCP configuration file for the given client. Specifically the following lines;

```
filename "pxeboot-9.3-amd64";
option root-path "192.168.2.7:/diskless/fbsd93-amd64";
```

restart the service and reboot the client machine. When the client machine comes back on-line it will be running the patched or new version of the OS.

FreeBSD Server (Math & Stats) Specs

Manufacturer: Citrix Xenserver 6.2

Model: Xen Client

CPU: Intel Xeon E3-1240 v3 @ 3.4GHz (2 cores)

Memory: 1GB

Drives: 500GB UFS2+J

FreeBSD Server (CS) Specs

Manufacturer: Dell OptiPlex

Model: 755

CPU: Intel Core 2 Duo E6550 @ 2.0GHz (2 cores)

Memory: 3GB

Drives: 2 x 500GB Seagate SATA II UFS2+J GMIRROR

FreeBSD Server/Client Config

```
prompt /usr/local/etc/istgt # cat auth.conf
[AuthGroup1]
    Auth    "discover" "XXXXXXXXXXXXX"
[AuthGroup10]
    Auth    "linuxfileserver" "YYYYYYYYYYYY"
prompt /usr/local/etc/istgt # cat istgtcontrol.conf
[Global]
    Comment      "ISTGT control configuration"
    Timeout      60
    AuthMethod   Auto
    Host         localhost
    Port         3261
    Flags        "rw"
    Size         "auto"
```

```
prompt /usr/local/etc/istgt # cat istgt.conf
[Global]
    NodeBase                  "iqn.2012-05.ca.mun.math.fbsd-diskstore"
    DiscoveryAuthMethod      CHAP
    DiscoveryAuthGroup       AuthGroup1
[UnitControl]
    Comment                   "Unit Controller"
    AuthMethod                CHAP Mutual
    AuthGroup                 AuthGroup
    Portal                UC1 127.0.0.1:3261
    Netmask                   127.0.0.1
[PortalGroup1]
    Comment                   "Portal Group 1"
    Portal                DA1 192.168.2.15:3260
[InitiatorGroup1]
    Comment                   "Initiator Group 1"
    InitiatorName             "ALL"
    Netmask                   192.168.2.0/23

[LogicalUnit10]
    Comment                  "Data Disk (iqn.ca.mun.math.fbsd-diskstore:linuxfileserver-
idisk0)"
    TargetName                iqn.2012-05.ca.mun.math.fbsd-diskstore:linuxfileserver-
idisk0
    TargetAlias              "linuxfileserver-idisk0"
    Mapping                  PortalGroup1 InitiatorGroup1
    AuthMethod                Auto
    AuthGroup                 AuthGroup10
    UnitType                 Disk
    QueueDepth                255
    LUN0                    Storage /storage/systems/linuxfileserver.math.mun.ca/idisk0
Auto
```

```
prompt ~# zfs create storage/systems/linuxfileserver.math.mun.ca

prompt ~# truncate -s 5T /storage/systems/linuxfileserver.math.mun.
ca/idisk0

prompt ~# /usr/local/etc/rc.d/istgt onestart
```

The /etc/rc.conf files are not updated to have the istgt service to startup at boot or import the ZFS volume. As I would first need to determine the state of the ZFS file system. This can be done by importing the file system or by running ‘zpool status’.

```
prompt ~# zpool import storage
prompt ~# zpool status
  pool: storage
 state: ONLINE
  scan: resilvered 118G in 2h54m with 0 errors on Tue Jan  7 14:30:22 2014
config:

     NAME        STATE     READ WRITE CKSUM
     storage     ONLINE       0     0     0
       raidz2-0  ONLINE       0     0     0
         aacd0   ONLINE       0     0     0
         aacd1   ONLINE       0     0     0
         aacd2   ONLINE       0     0     0
         aacd3   ONLINE       0     0     0
         aacd4   ONLINE       0     0     0
         aacd5   ONLINE       0     0     0
         aacd6   ONLINE       0     0     0
         aacd7   ONLINE       0     0     0

errors: No known data errors
```

Please note that the client FreeBSD node located in our off-site location. Is setup differently, in that its local ZFSv1 volume is used to store the backup information copied over from the linux file server through the use of rsync.

Once the data has been sync'd over. A ZFS snapshot is taken and a thirty two (32) day backup window is maintained (if possible) by utilizing a cron script.

Linux File Server Config

```
prompt /etc/iscsi # cat initiatorname.iscsi
InitiatorName=iqn.2012-05.ca.mun.math.linuxfileserver

prompt /etc/iscsi # cat iscsid.conf
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = discover
discovery.sendtargets.auth.password = XXXXXXXXXXX
node.session.auth.authmethod = CHAP
node.session.auth.username = linuxfileserver
node.session.auth.password = YYYYYYYYYYY
node.startup = manual

prompt ~ # /etc/init.d/iscsi start
prompt ~ # iscsiadm -m discovery --type sendtargets --portal 192.168.2.15:3260

prompt ~ # iscsiadm -m node --targetname iqn.2012-05.ca.mun.math.fbsd-diskstore:linuxfileserver-idisks --portal 192.168.2.15:3260 --login
```

Once you’ve logged in successfully to the iSCSI FreeBSD disk store service. You can either look into /etc/partitions or read the output from the kernel buffer ring using the command ‘dmesg’.

```
prompt ~ # mount /dev/sdc /users/math/research/fs1

prompt ~ # mount -o bind /users/math/research/fs1 /nfs4/users/math/research/fs1

prompt ~ # /etc/init.d/nfs restart
```

```
linuxfileserver prompt ~# df -h -t ext4
Filesystem       Size  Used Avail Use% Mounted on
/dev/md0          28G   25G  2.0G  93% /
/dev/md1         4.5G  443M  3.8G  11% /var
/dev/sde         5.0T  1.6T  3.3T  33% /users/math/research/fs1

linuxclient prompt ~# cd /autofs/research/fs1 ; df -h -t nfs4
Filesystem                               Size  Used Avail Use% Mounted on
192.168.2.7:/users/math/research/fs1    5.0T  1.6T  3.3T  33%
/autofs/research/fs1
```

Future Work

Migrate the diskless FreeBSD information away from using a traditional file system to one based on a memory backed file system.

```
mdconfig -a -t vnode -f /home/images/FBSD92-amd64 -u 100
mount -ro /dev/md100 /diskless/FBSD92-amf64
```

Move towards a more fault tolerant/redundent configuration. Create an iSCSI targets on each FreeBSD data store and have Linux create a RAID-1 mirror array.

Look at the possibility of abstracting out the FreeBSD client config files and directories from /diskless/fbsd<version number>-amd64/conf so that it would be possible to ensure that the files are consistent when possible across all FreeBSD versions.

Procure one or two more storage nodes to install in Computer Science in order to expand Math & Stats’ offsite backups.

Questions?

Thank you all!

From the conference attendees, BSDCan, event support staff and the BSD ecosystem.