[SoftNAS KB]: Features of the ZFS filesystem

What is ZFS and Why Should I Use it?

Endless scalability

Well, it's not *technically* endless, but it's a 128-bit file system that's capable of managing zettabytes (one billion terabytes) of data. No matter how much hard drive space you have, ZFS will be suitable for managing it.

Snapshots and Clones

One of the other powerful features of ZFS is the ability to take snapshots, which allow you to preserve "point in time" versions of the filesystem.

Here's an example of a snapshot using the recursive (-r) flag...it will also create a snapshot of each sub-dataset using the same snapshot name.

```
----------

# zfs snapshot -r mypool/usr/ports@firstsnapshot # zfs list -t all -o name,used,refer,written NAME USED REFER
WRITTEN mypool 10.5M 44.9K 44.9K mypool/usr 10.1M 44.9K 44.9K mypool/usr/ports 10.1M 10.0M 0 mypool/usr
/ports@firstsnapshot 0 10.0M 10.0M mypool/usr/ports/distfiles 43.4K 43.4K 0 mypool/usr/ports
/distfiles@firstsnapshot 0 43.4K 43.4K

--------
```

As you can see, the snapshots initially take no additional space, as they only contain the data that already exists in the dataset the snapshot belongs to. ZFS snapshots are read-only, but they can be accessed via a hidden ".zfs" directory. This allows you to easily restore a single file that was accidentally modified or deleted:

```
--------


# ls -lh /mypool/usr/ports/.zfs/snapshot/firstsnapshot/  total 10247 -rw-r--r-- 1 root wheel 10M Nov 29 15:01
somefile
---------
```

ZFS has the "rollback" operation, which reverts all changes written since the snapshot:

Sending and Receiving Snapshots

ZFS lets you send snapshots of your pool or dataset and output it to a file. You can also pipe it to other commands. This can be used to send datasets over the internet, **using SSH**, and receive them on a remote host.

This is a local backup.

To restore the backup from the file, we would do something like:

```
# zfs receive -v mypool/myfiles < /mnt/filesystem-backup
```

To copy that snapshot to a remote server.

```
# zfs send mypool/myfiles@backup | ssh you@remoteserver zfs receive -v otherpool/myfiles
```

Here's an example of an incremental backup:

```
# zfs send -i mypool/myfiles@backup mypool/myfiles@laterbackup |
\ ssh you@remoteserver zfs receive -v otherpool/myfiles
```

Data Integrity

One of the major selling features of ZFS is the safety it provides. All data and metadata written are checksummed to ensure that the data has not become corrupted over time. Every time data is read from a ZFS pool, the checksum is calculated and compared to the checksum that was calculated when the data was originally written. If they do not match, it means the data has become corrupted on the storage. If you have a redundant ZFS pool, the corruption will be automatically repaired and noted in the status screen.

```
# zpool scrub mypool
# zpool status mypool

pool: mypool
state: ONLINE
scan: scrub in progress since Fri Nov 29 15:30:59 2013
      13.8M scanned out of 13.9M at 4.60M/s, 0h0m to go
0 repaired, 99.45% done
config:

NAME                      STATE     READ WRITE CKSUM
mypool                    ONLINE       0     0     0
 raidz1-0                 ONLINE       0     0     0
    /tutorial/file1       ONLINE       0     0     0
    /tutorial/file2       ONLINE       0     0     0
    /tutorial/file3       ONLINE       0     0     0
    /tutorial/file4       ONLINE       0     0     0

errors: No known data errors
```

Here we are simulating the failure of a disk:

```
# rm /tutorial/file3
# zpool scrub mypool
# zpool status mypool

pool: mypool
state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://illumos.org/msg/ZFS-8000-2Q
scan: scrub repaired 0 in 0h0m with 0 errors on Fri Nov 29 15:36:45 2013
config:

NAME                      STATE     READ WRITE CKSUM
mypool                    DEGRADED     0     0     0
  raidz1-0                DEGRADED     0     0     0
    /tutorial/file1       ONLINE       0     0     0
    /tutorial/file2       ONLINE       0     0     0
    4747444483933995 70   UNAVAIL      0     0     0  was /tutorial/file3
    /tutorial/file4       ONLINE       0     0     0

errors: No known data errors
```

Here we replace the failed disk with a new one:

```
# truncate -s 2G /tutorial/file3 # zpool replace mypool 474744448393399570 /tutorial/file3 # zpool status mypool
pool: mypool state: ONLINE status: One or more devices is currently being resilvered. The pool will continue to
function, possibly in a degraded state. action: Wait for the resilver to complete. scan: resilver in progress
since Fri Nov 29 15:38:42 2013 13.8M scanned out of 14.1M at 3.46M/s, 0h0m to go 3.45M resilvered, 98.17% done
config: NAME STATE READ WRITE CKSUM mypool ONLINE 0 0 0 raidz1-0 ONLINE 0 0 0 /tutorial/file1 ONLINE 0 0 0
/tutorial/file2 ONLINE 0 0 0 replacing-2 UNAVAIL 0 0 0 474744448393399570 UNAVAIL 0 0 0 was /tutorial/file3/old
/tutorial/file3 ONLINE 0 0 0 (resilvering) /tutorial/file4 ONLINE 0 0 0 errors: No known data errors


---------------------


Once the resilver is finished the pool is back to normal:


# zpool status mypool ---------------------- pool: mypool state: ONLINE scan: resilvered 3.45M in 0h0m with 0
errors on Fri Nov 29 15:38:46 2013 config: NAME STATE READ WRITE CKSUM mypool ONLINE 0 0 0 raidz1-0 ONLINE 0 0 0
/tutorial/file1 ONLINE 0 0 0 /tutorial/file2 ONLINE 0 0 0 /tutorial/file3 ONLINE 0 0 0 /tutorial/file4 ONLINE 0 0
0 errors: No known data errors
----------------------
```

Drive pooling

The creators of ZFS want you to think of it as being similar to the way your computer uses RAM. When you need more memory in your computer, you put in another stick and you're done. Similarly with ZFS, when you need more hard drive space, you put in another hard drive and you're done. No need to spend time partitioning, formatting, initializing, or doing anything else to your disks – when you need a bigger storage "pool," just add disks.

RAID

ZFS is capable of many different RAID levels, all while delivering performance that's comparable to that of hardware RAID controllers. This allows you to save money, make setup easier, and have access to superior RAID levels that ZFS has improved upon.