ZFS - Perfect Filesystem for Containers

Damian Wojsław
Technical Support Engineer, Canonical
Zettabyte Filesystem

Copy-on-write filesystem for Sun Solaris, now Oracle Solaris
OpenZFS - the implementation in illumos, *BSD, Linux
checksumming
compression
deduplication
extended ACLS
snapshots
clones
RAID and Volume Manager
pool growth
zvols
remote replication
remote replication
snapshots clones
extended ACLS
pool growth
RAID and Volume Manager

root vdev - pool

lvdev

pvdev pvdev pvdev

pvdev pvdev pvdev
<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
<th>MOUNTPOINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>406K</td>
<td>108G19K</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>rpool</td>
<td>47.3G</td>
<td>383G96K</td>
<td></td>
<td>/</td>
</tr>
<tr>
<td>rpool/ROOT</td>
<td>3.53G</td>
<td>383G96K</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>rpool/ROOT/ubuntu</td>
<td>3.53G</td>
<td>383G</td>
<td>3.39G</td>
<td>/</td>
</tr>
<tr>
<td>rpool/home</td>
<td>26.5G</td>
<td>383G96K</td>
<td></td>
<td>/home</td>
</tr>
<tr>
<td>rpool/home/root</td>
<td>216K</td>
<td>383G</td>
<td>216K</td>
<td>/root</td>
</tr>
<tr>
<td>rpool/home/trochej</td>
<td>26.5G</td>
<td>383G</td>
<td>22.8G</td>
<td>/home/trochej</td>
</tr>
<tr>
<td>rpool/home/trochej/Music</td>
<td>1015M</td>
<td>383G</td>
<td>1015M</td>
<td>/home/trochej/Music</td>
</tr>
<tr>
<td>rpool/home/trochej/Videos</td>
<td>2.70G</td>
<td>383G</td>
<td>2.70G</td>
<td>/home/trochej/Videos</td>
</tr>
<tr>
<td>rpool/swap</td>
<td>17.0G</td>
<td>400G</td>
<td>12.5M</td>
<td></td>
</tr>
<tr>
<td>rpool/var</td>
<td>191M</td>
<td>383G</td>
<td>96K</td>
<td>/var</td>
</tr>
<tr>
<td>rpool/var/cache</td>
<td>150M</td>
<td>383G</td>
<td>150M</td>
<td>legacy</td>
</tr>
<tr>
<td>rpool/var/log</td>
<td>40.5M</td>
<td>383G</td>
<td>40.5M</td>
<td>legacy</td>
</tr>
<tr>
<td>rpool/var/spool</td>
<td>116K</td>
<td>383G</td>
<td>116K</td>
<td>legacy</td>
</tr>
<tr>
<td>rpool/var/tmp</td>
<td>168K</td>
<td>383G</td>
<td>168K</td>
<td>legacy</td>
</tr>
</tbody>
</table>
Space allocation works differently than in traditional filesystems and volume managers.

Filesystems are free to allocate blocks from the whole pool - no preallocated filesystem space.
Common pool layouts - RAIDZ, RAIDZ-2

One vdev
Common pool layouts - RAIDZ, RAIDZ-2

Multiple vdevs
Common pool layouts - mirror

Multiple vdevs
Here be dragons

- Space utilization reporting: zfs -o space,
- Pool health
- Deduplication
ZFS as root filesystem

1. Cheap and quick clones of running system for experimentation
2. As above, but for preparing a backup
3. “Golden image” copy of system
LXD is Canonical’s container hypervisor

- Ultra fast “vm-lite” guests (bare metal speed)
- Any distribution of Linux - e.g. Ubuntu, CentOS
- Starts in less than 1 second
- 1*x density of KVM or ESX for idle workloads

LXD REST API

nova-lxd lxc cli other restful apps
LXD and ZFS

1. ZFS as a storage backend to LXD
2. Snapshot, restore
3. Compression on the fly
4. Deduplication (don’t forget to monitor closely)
5. Simplified backup
Demo
References

http://open-zfs.org/
http://github.com/openzfs
http://github.com/zfsonlinux
http://linuxcontainers.org/
http://ubuntu.com/lxd
http://github.com/lxc
Freenode: #openzfs, #ubuntu, #zfsonlinux, #lxcontainers