NAME

zfsd - ZFS fault management daemon

SYNOPSIS

zfsd [**-d**]

DESCRIPTION

zfsd attempts to resolve ZFS faults that the kernel can't resolve by itself. It listens to devctl(4) events, which are how the kernel notifies userland of events such as I/O errors and disk removals. **zfsd** attempts to resolve these faults by activating or deactivating hot spares and onlining offline vdevs.

The following options are available:

-d Run in the foreground instead of daemonizing.

System administrators never interact with **zfsd** directly. Instead, they control its behavior indirectly through zpool configuration. There are two ways to influence **zfsd**: assigning hotspares and setting pool properties. Currently, only the *autoreplace* property has any effect. See zpool(8) for details.

zfsd will attempt to resolve the following types of fault:

device removal
When a leaf vdev disappears, **zfsd** will activate any available hotspare.

device arrival
When a new GEOM device appears, **zfsd** will attempt to read its ZFS label, if any. If it matches a previously removed vdev on an active pool, **zfsd** will online it. Once resilvering completes, any active hotspare will detach automatically.

If the new device has no ZFS label but its physical path matches the physical path of a previously removed vdev on an active pool, and that pool has the autoreplace property set, then **zfsd** will replace the missing vdev with the newly arrived device. Once resilvering completes, any active hotspare will detach automatically.

vdev degrade or fault events
If a vdev becomes degraded or faulted, **zfsd** will activate any available hotspare.

I/O errors
If a leaf vdev generates more than 50 I/O errors in a 60 second period, then **zfsd** will mark that vdev as *FAULTED*. ZFS will no longer issue any I/Os to it. **zfsd** will activate a hotspare if one is

available.

Checksum errors

If a leaf vdev generates more than 50 checksum errors in a 60 second period, then **zfsd** will mark that vdev as *DEGRADED*. ZFS will still use it, but zfsd will activate a spare anyway.

Spare addition

If the system administrator adds a hotspare to a pool that is already degraded, **zfsd** will activate the spare.

Resilver complete

zfsd will detach any hotspare once a permanent replacement finishes resilvering.

Physical path change

If the physical path of an existing disk changes, **zfsd** will attempt to replace any missing disk with the same physical path, if its pool’s autoreplace property is set.

zfsd will log interesting events and its actions to syslog with facility *daemon* and identity [zfsd].

FILES

/var/db/zfsd/cases

When **zfsd** exits, it serializes any unresolved casefiles here, then reads them back in when next it starts up.

SEE ALSO

devctl(4), zpool(8)

HISTORY

zfsd first appeared in FreeBSD 11.0.

AUTHORS

zfsd was originally written by Justin Gibbs <*gibbs@FreeBSD.org*> and
Alan Somers <*asomers@FreeBSD.org*>

TODO

In the future, **zfsd** should be able to resume a pool that became suspended due to device removals, if enough missing devices have returned.