NAME

zinject - ZFS Fault Injector

DESCRIPTION

zinject creates artificial problems in a ZFS pool by simulating data corruption or device failures. This program is dangerous.

SYNOPSIS

zinject List injection records.

zinject -b *objset*:*object*:*level*:*start*:*end*
[**-f** *frequency*]
-amu
[pool]

Force an error into the pool at a bookmark.

zinject -c *id*|**all**

Cancel injection records.

zinject -d *vdev*
-A degrade|**fault**
pool

Force a vdev into the DEGRADED or FAULTED state.

zinject -d *vdev*
-D *latency*:*lanes*
pool

Add an artificial delay to I/O requests on a particular device, such that the requests take a minimum of *latency* milliseconds to complete. Each delay has an associated number of *lanes* which defines the number of concurrent I/O requests that can be processed.

For example, with a single lane delay of 10 ms (**-D** *10*:*1*), the device will only be able to service a single I/O request at a time with each request taking 10 ms to complete. So, if only a single request is submitted every 10 ms, the average latency will be 10 ms; but if more than one request is submitted every 10 ms, the average latency will be more than 10 ms.

Similarly, if a delay of 10 ms is specified to have two lanes (**-D** *10*:*2*), then the device will be able to service two requests at a time, each with a minimum latency of 10 ms. So, if two requests are submitted every 10 ms, then the average latency will be 10 ms; but if more than two requests are submitted every 10 ms, the average latency will be more than 10 ms.

Also note, these delays are additive. So two invocations of **-D** *10*:*1* are roughly equivalent to a single invocation of **-D** *10*:*2*. This also means, that one can specify multiple lanes with differing target latencies. For example, an invocation of **-D** *10*:*1* followed by **-D** *25*:*2* will create 3 lanes on the device: one lane with a latency of 10 ms and two lanes with a 25 ms latency.

zinject -d *vdev*
[**-e** *device_error*]
[**-L** *label_error*]
[**-T** *failure*]
[**-f** *frequency*]
[**-F**]
pool

Force a vdev error.

zinject -I
[**-s** *seconds*|**-g** *txgs*]
pool

Simulate a hardware failure that fails to honor a cache flush.

zinject -p *function*
pool

Panic inside the specified function.

zinject -t data
-C *dvas*
[**-e** *device_error*]
[**-f** *frequency*]
[**-l** *level*]
[**-r** *range*]
[**-amq**]
path

Force an error into the contents of a file.

zinject -t dnode
-C *dvas*
[**-e** *device_error*]
[**-f** *frequency*]
[**-l** *level*]
[**-amq**]

path
Force an error into the metadnode for a file or directory.

zinject -t *mos_type*
-C *dvas*
[**-e** *device_error*]
[**-f** *frequency*]
[**-l** *level*]
[**-r** *range*]
[**-amqu**]
pool
Force an error into the MOS of a pool.

OPTIONS

-a Flush the ARC before injection.

-b *objset*:*object*:*level*:*start*:*end*
Force an error into the pool at this bookmark tuple. Each number is in hexadecimal, and only one block can be specified.

-C *dvas* Inject the given error only into specific DVAs. The mask should be specified as a list of 0-indexed DVAs separated by commas (e.g. *0,2*). This option is not applicable to logical data errors such as **decompress** and **decrypt**.

-d *vdev* A vdev specified by path or GUID.

-e *device_error*
Specify

checksum	for an ECKSUM error,
decompress	for a data decompression error,
decrypt	for a data decryption error,
corrupt	to flip a bit in the data after a read,
dtl	for an ECHILD error,
io	for an EIO error where reopening the device will succeed, or
nxio	for an ENXIO error where reopening the device will fail.

For EIO and ENXIO, the "failed" reads or writes still occur. The probe simply sets the error value reported by the I/O pipeline so it appears the read or write failed. Decryption errors only currently work with file data.

-f *frequency*
Only inject errors a fraction of the time. Expressed as a real number percentage between **0.0001** and **100**.

-F Fail faster. Do fewer checks.

-f *txgs* Run for this many transaction groups before reporting failure.

-h Print the usage message.

-l *level* Inject an error at a particular block level. The default is **0**.

-L *label_error*
Set the label error region to one of **nvlist**, **pad1**, **pad2**, or **uber**.

-m Automatically remount the underlying filesystem.

-q Quiet mode. Only print the handler number added.

-r *range*
Inject an error over a particular logical range of an object, which will be translated to the appropriate blkid range according to the object's properties.

-s *seconds*
Run for this many seconds before reporting failure.

-T *failure*
Set the failure type to one of **all**, **claim**, **free**, **read**, or **write**.

-t *mos_type*
Set this to

mos	for any data in the MOS,
mosdir	for an object directory,
config	for the pool configuration,
bpobj	for the block pointer list,
spacemap	for the space map,
metaslab	for the metaslab, or
errlog	for the persistent error log.

-u Unload the pool after injection.

ENVIRONMENT VARIABLES

ZFS_HOSTID
: Run **zinject** in debug mode.

SEE ALSO

zfs(8), zpool(8)