

NAME

gvirstor - control utility for virtual data storage devices

SYNOPSIS

gvirstor label [-hv] [-s *virsize*] [-m *chunksize*] *name prov* ...

gvirstor stop [-fv] *name* ...

gvirstor destroy [-fv] *name* ...

gvirstor add [-vh] *name prov* ...

gvirstor remove [-v] *name prov* ...

gvirstor clear [-v] *prov* ...

gvirstor dump *prov* ...

gvirstor list

gvirstor status

gvirstor load

gvirstor unload

DESCRIPTION

The **gvirstor** utility is used for setting up a virtual storage device of arbitrary large size (for example, several TB), consisting of an arbitrary number of physical storage devices with the total size which is equal to or smaller than the virtual size. Data for the virtual devices will be allocated from physical devices on demand. The idea behind **gvirstor** is similar to the concept of Virtual Memory in operating systems, effectively allowing users to overcommit on storage (free file system space). The concept is also known as "thin provisioning" in virtualization environments, only here it is implemented on the level of physical storage devices.

The first argument to **gvirstor** indicates an action to be performed:

label Set up a virtual device from the given components with the specified *name*. Metadata is stored in the last sector of every component. Argument **-s** *virsize* is the size of new virtual device, with default being set to 2 TiB (2097152 MiB). Argument **-m** *chunksize* is the chunk size, with default being set to 4 MiB (4096 KiB). The default arguments are thus "**-s 2097152 -m 4096**".

stop Turn off an existing virtual device with the given *name*. This command does not touch on-disk metadata. As with other GEOM classes, stopped geoms cannot be started manually.

destroy Same as **stop**.

add Adds new components to existing virtual device with the given *name*. The specified virstor device must exist and be active (i.e. module loaded, device present in */dev*). This action can be safely performed while the virstor device is in use ("hot" operation).

remove Removes components from existing virtual device with the given *name*. Only unallocated providers can be removed.

clear Clear metadata on the given providers.

dump Dump metadata stored on the given providers.

list See geom(8).

status See geom(8).

load See geom(8).

unload See geom(8).

Additional options:

-f
Force the removal of the specified virtual device.

-h
Hardcode providers' names in metadata.

-v
Be more verbose.

EXAMPLES

The following example shows how to create a virtual device of default size (2 TiB), of default chunk (extent) size (4 MiB), with two physical devices for backing storage.

```
gvirstor label -v mydata /dev/ada4 /dev/ada6  
newfs /dev/virstor/mydata
```

From now on, the virtual device will be available via the */dev/virstor/mydata* device entry. To add a new physical device / component to an active virstor device:

```
gvirstor add mydata ada8
```

This will add physical storage of *ada8* to */dev/virstor/mydata* device.

To see the device status information (including how much physical storage is still available for the virtual device), use:

```
gvirstor list
```

All standard `geom(8)` subcommands (e.g. **status**, **help**) are also supported.

SYSCTL VARIABLES

gvirstor has several `sysctl(8)` tunable variables.

```
int kern.geom.virstor.debug
```

This `sysctl` controls verbosity of the kernel module, in the range 1 to 15. Messages that are marked with higher verbosity levels than this are suppressed. Default value is 5 and it is not recommended to set this tunable to less than 2, because level 1 messages are error events, and level 2 messages are system warnings.

```
int kern.geom.virstor.chunk_watermark
```

Value in this `sysctl` sets warning watermark level for physical chunk usage on a single component. The warning is issued when a `virstor` component has less than this many free chunks (default 100).

```
int kern.geom.virstor.component_watermark
```

Value in this `sysctl` sets warning watermark level for component usage. The warning is issued when there are less than this many unallocated components (default is 1).

All these `sysctls` are also available as `loader(8)` tunables.

DIAGNOSTICS

The **gvirstor** utility exits 0 on success, and >0 if an error occurs.

gvirstor kernel module issues log messages with prefixes in standardized format, which is useful for log message filtering and dispatching. Each message line begins with

```
GEOM_VIRSTOR[%d]:
```

The number (%d) is message verbosity / importance level, in the range 1 to 15. If a message filtering, dispatching or operator alert system is used, it is recommended that messages with levels 1 and 2 be taken seriously (for example, to catch out-of-space conditions as set by watermark) `sysctls`.

SEE ALSO

geom(4), fstab(5), geom(8), glabel(8), newfs(8)

HISTORY

The **gvirstor** utility first appeared in FreeBSD 7.0.

AUTHORS

Ivan Voras <*ivoras@FreeBSD.org*>

Sponsored by Google Summer of Code 2006.

BUGS

Commands **add** and **remove** contain unavoidable critical sections which may make the virstor device unusable if a power failure (or other disruptive event) happens during their execution. It is recommended to run them when the system is quiescent.

ASSUMPTIONS AND INTERACTION WITH FILE SYSTEMS

There are several assumptions that **gvirstor** has in its operation: that the size of the virtual storage device will not change once it is set, and that the sizes of individual physical storage components will always remain constant during their existence. For alternative ways to implement virtual or resizable file systems see `zfs(1M)`, `gconcat(8)` and `growfs(8)`.

Note that **gvirstor** has nontrivial interaction with file systems which initialize a large number of on-disk structures during `newfs`. If such file systems attempt to spread their structures across the drive media (like UFS/UFS2 does), their efforts will be effectively foiled by sequential allocation of chunks in **gvirstor** and all their structures will be physically allocated at the start of the first virstor component. This could have a significant impact on file system performance (which can in some rare cases be even positive).