NAME

ccd - Concatenated Disk driver

SYNOPSIS

device ccd

DESCRIPTION

The **ccd** driver provides the capability of combining one or more disks/partitions into one virtual disk.

This document assumes that you are familiar with how to generate kernels, how to properly configure disks and devices in a kernel configuration file, and how to partition disks.

In order to compile in support for the **ccd**, you must add a line similar to the following to your kernel configuration file:

```
device   ccd                # concatenated disk devices
```

As of the FreeBSD 3.0 release, you do not need to configure your kernel with **ccd** but may instead use it as a kernel loadable module. Simply running ccdconfig(8) will load the module into the kernel.

A **ccd** may be either serially concatenated or interleaved. To serially concatenate the partitions, specify the interleave factor of 0. Note that mirroring may not be used with an interleave factor of 0.

There is a run-time utility that is used for configuring **ccd**s. See ccdconfig(8) for more information.

The Interleave Factor

If a **ccd** is interleaved correctly, a "striping" effect is achieved, which can increase sequential read/write performance. The interleave factor is expressed in units of DEV_BSIZE (usually 512 bytes). For large writes, the optimum interleave factor is typically the size of a track, while for large reads, it is about a quarter of a track. (Note that this changes greatly depending on the number and speed of disks.) For instance, with eight 7,200 RPM drives on two Fast-Wide SCSI buses, this translates to about 128 for writes and 32 for reads. A larger interleave tends to work better when the disk is taking a multitasking load by localizing the file I/O from any given process onto a single disk. You lose sequential performance when you do this, but sequential performance is not usually an issue with a multitasking load.

An interleave factor must be specified when using a mirroring configuration, even when you have only two disks (i.e., the layout winds up being the same no matter what the interleave factor). The interleave factor will determine how I/O is broken up, however, and a value 128 or greater is recommended.

ccd has an option for a parity disk, but does not currently implement it.

The best performance is achieved if all component disks have the same geometry and size. Optimum striping cannot occur with different disk types.

For random-access oriented workloads, such as news servers, a larger interleave factor (e.g., 65,536) is more desirable. Note that there is not much **ccd** can do to speed up applications that are seek-time limited. Larger interleave factors will at least reduce the chance of having to seek two disk-heads to read one directory or a file.

Disk Mirroring

You can configure the **ccd** to "mirror" any even number of disks. See ccdconfig(8) for how to specify the necessary flags. For example, if you have a **ccd** configuration specifying four disks, the first two disks will be mirrored with the second two disks. A write will be run to both sides of the mirror. A read will be run to either side of the mirror depending on what the driver believes to be most optimal. If the read fails, the driver will automatically attempt to read the same sector from the other side of the mirror. Currently **ccd** uses a dual seek zone model to optimize reads for a multi-tasking load rather than a sequential load.

In an event of a disk failure, you can use dd(1) to recover the failed disk.

Note that a one-disk **ccd** is not the same as the original partition. In particular, this means if you have a file system on a two-disk mirrored **ccd** and one of the disks fail, you cannot mount and use the remaining partition as itself; you have to configure it as a one-disk **ccd**. You cannot replace a disk in a mirrored **ccd** partition without first backing up the partition, then replacing the disk, then restoring the partition.

Linux Compatibility

The Linux compatibility mode does not try to read the label that Linux' md(4) driver leaves on the raw devices. You will have to give the order of devices and the interleave factor on your own. When in Linux compatibility mode, **ccd** will convert the interleave factor from Linux terminology. That means you give the same interleave factor that you gave as chunk size in Linux.

If you have a Linux md(4) device in "legacy" mode, do not use the CCDF_LINUX flag in ccdconfig(8). Use the CCDF_NO_OFFSET flag instead. In that case you have to convert the interleave factor on your own, usually it is Linux' chunk size multiplied by two.

Using a Linux RAID this way is potentially dangerous and can destroy the data in there. Since FreeBSD does not read the label used by Linux, changes in Linux might invalidate the compatibility layer.

However, using this is reasonably safe if you test the compatibility before mounting a RAID read-write for the first time. Just using ccdconfig(8) without mounting does not write anything to the Linux RAID. Then you do a **fsck.ext2fs** (*ports/sysutils/e2fsprogs*) on the **ccd** device using the **-n** flag. You can mount the file system read-only to check files in there. If all this works, it is unlikely that there is a problem with **ccd**. Keep in mind that even when the Linux compatibility mode in **ccd** is working correctly, bugs in FreeBSD's **ex2fs** implementation would still destroy your data.

WARNINGS

If just one (or more) of the disks in a **ccd** fails, the entire file system will be lost unless you are mirroring the disks.

If one of the disks in a mirror is lost, you should still be able to back up your data. If a write error occurs, however, data read from that sector may be non-deterministic. It may return the data prior to the write or it may return the data that was written. When a write error occurs, you should recover and regenerate the data as soon as possible.

Changing the interleave or other parameters for a **ccd** disk usually destroys whatever data previously existed on that disk.

FILES

*/dev/ccd** **ccd** device special files

SEE ALSO

dd(1), ccdconfig(8), config(8), disklabel(8), fsck(8), gvinum(8), mount(8), newfs(8)

HISTORY

The concatenated disk driver was originally written at the University of Utah.