NAME

nda - NVMe Direct Access device driver

SYNOPSIS

device nvme
device scbus

DESCRIPTION

The **nda** driver provides support for direct access devices, implementing the NVMe command protocol, that are attached to the system through a host adapter supported by the CAM subsystem.

SYSCTL VARIABLES

The following variables are available as both sysctl(8) variables and loader(8) tunables:

hw.nvme.use_nvd
The nvme(4) driver will create **nda** device nodes for block storage when set to 0. Create nvd(4) device nodes for block storage when set to 1. See nvd(4) when set to 1.

kern.cam.nda.nvd_compat
When set to 1, nvd(4) aliases will be created for all **nda** devices, including partitions and other geom(4) providers that take their names from the disk’s name. nvd(4) devices will not, however, be reported in the *kern.disks* sysctl(8).

kern.cam.nda.sort_io_queue
This variable determines whether the software queued entries are sorted in LBA order or not. Sorting is almost always a waste of time. The default is to not sort.

kern.cam.nda.enable_biospeedup
This variable determines if the **nda** devices participate in the speedup protocol. When the device participates in the speedup, then when the upper layers send a *BIO_SPEEDUP*, all current *BIO_DELETE* requests not yet sent to the hardware are completed successfully immediate without sending them to the hardware. Used in low disk space scenarios when the filesystem encounters a critical shortage and needs blocks immediately. Since trims have maximum benefit when the LBA is unused for a long time, skipping the trim when space is needed for immediate writes results in little to no excess wear. When participation is disabled, *BIO_SPEEDUP* requests are ignored.

kern.cam.nda.max_trim
The maximum number of LBA ranges to be collected together for each DSM trims send to the hardware. Defaults to 256, which is the maximum number of ranges the protocol supports. Sometimes poor trim performance can be mitigated by limiting the number of ranges sent to the

device. This value must be between 1 and 256 inclusive.

The following report per-device settings, and are read-only unless otherwise indicated. Replace *N* with the device unit number.

kern.cam.nda.N.rotating
This variable reports whether the storage volume is spinning or flash. Its value is hard coded to 0 indicating flash.

kern.cam.nda.N.unmapped_io
This variable reports whether the **nda** driver accepts unmapped I/O for this unit.

kern.cam.nda.N.flags
This variable reports the current flags.

OPEN
The device is open.

DIRTY
Set when a write to the drive is scheduled. Cleared after flush commands.

SCTX_INIT
Internal flag set after sysctl(8) nodes have been created.

kern.cam.nda.N.sort_io_queue
Same as the *kern.cam.nda.sort_io_queue* tunable.

kern.cam.nda.N.trim_ticks
Writable. When greater than zero, hold trims for up to this many ticks before sending to the drive. Sometimes waiting a little bit to collect more trims to send at one time improves trim performance. When 0, no delaying of trims are done.

kern.cam.nda.N.trim_goal
Writable. When delaying a bit to collect multiple trims, send the accumulated DSM TRIM to the drive.

kern.cam.nda.N.trim_lbas
Total number of LBAs that have been trimmed.

kern.cam.nda.N.trim_ranges

Total number of LBA ranges that have been trimmed.

kern.cam.nda.N.trim_count
Total number of trims sent to the hardware.

kern.cam.nda.N.deletes
Total number of *BIO_DELETE* requests queued to the device.

NAMESPACE MAPPING

Each nvme(4) drive has one or more namespaces associated with it. One instance of the **nda** driver will be created for each of the namespaces on the drive. All the **nda** nodes for a nvme(4) device are at target 0. However, the namespace ID maps to the CAM lun, as reported in kernel messages and in the *devlist* sub command of camcontrol(8).

Namespaces are managed with the *ns* sub command of nvmecontrol(8). Not all drives support namespace management, but all drives support at least one namespace. Device nodes for **nda** will be created and destroyed dynamically as namespaces are activated or detached.

FILES

*/dev/nda** NVMe storage device nodes

SEE ALSO

cam(4), geom(4), nvd(4), nvme(4), gpart(8)

HISTORY

The **nda** driver first appeared in FreeBSD 12.0.

AUTHORS

Warner Losh <*imp@FreeBSD.org*>