Solid State Drives/NVMe

< Solid State Drives

NVM Express (NVMe) is a specification for accessing SSDs attached through the PCI Express bus. As a logical device interface, NVM Express has been designed from the ground up, capitalizing on the low latency and parallelism of PCI Express SSDs, and mirroring the parallelism of contemporary CPUs, platforms and applications.

Related articles
Solid State Drives

Contents

Installation

The Linux NVMe driver is natively included in the kernel since version 3.3. NVMe devices should show up under `/dev/nvme*`.

Extra userspace NVMe tools can be found in **nvme-cli (https://aur.archlinux.org/packages/nvme-cli/)**[AUR] or **nvme-cli-git (https://aur.archlinux.org/packages/nvme-cli-git/)**[AUR].

See **Solid State Drives** for supported filesystems, maximizing performance, minimizing disk reads/writes, etc.

Performance

Sector size

See **Advanced Format#How to determine if HDD employ a 4k sector**.

Discards

Note: Although **continuous TRIM** is an option (albeit not recommended) for SSDs, **NVMe devices should not be issued discards**.

Discards are disabled by default on typical setups that use **ext4** and **LVM**, but other filesystems might need discards to be disabled explicitly.

Intel, as one device manufacturer, recommends not to enable discards at the filesystem level, but suggests the **periodic TRIM** method, or apply `fstrim` manually.**[1] (https://communities.intel.com/thread/75161?start=0&tstart=0)**

Airflow

NVMe SSDs are known to be affected by high operating temperatures and will throttle performance over certain thresholds.**[2] (http://www.legitreviews.com/samsung-ssd-950-pro-512gb-nvme-pcie-ssd-review_174096/3)**

Testing

Raw device performance tests can be run with **hdparm (https://www.archlinux.org/packages/?name=hdparm)**:

```
# hdparm -Tt --direct /dev/nvme0n1
```

Power Saving APST

NVME Power Saving Patch

Andy Lutomirski has created a patchset which fixes powersaving for NVME devices in linux. The patch has been merged into mainline kernel v4.11.

To test if NVME Power Management is working, install **nvme-cli (https://aur.archlinux.org/packages/nvme-cli/)**^{AUR} if running an older kernel, and run
`# nvme get-feature -f 0x0c -H /dev/nvme[0-9]` .

When APST is enabled the output should contain "Autonomous Power State Transition Enable (APSTE): Enabled" and there should be non-zero entries in the table below indicating the idle time before transitioning into each of the available states.

If ASPT is enabled but no non-zero states appear in the table, the latencies might be too high for any states to be enabled by default. The output of `# nvme id-ctrl /dev/nvme[0-9]` should show the available non-operational power states of the NVME controller. If the total latency of any state (enlat + xlat) is greater than 25000 (25ms) then to enable it you must pass a value at least that high to the `default_ps_max_latency_us` option for the `nvme_core` module in the boot parameters. This should enable ASPT and make the table in `# nvme get-feature` show the entries.

On 4.13, drive errors can occur (ext4 fs) that cause the system to become unusable. This seems to be the result of a power saving state that the drive (Intel NVMe, in my case) cannot use. Adding the boot parameter nvme_core.default_ps_max_latency_us=5500 disables the lowest power saving state.

References

- **Intel Linux NVMe driver reference (http://www.intel.com/content/dam/support/us/en/documents/ssdc/data-center-ssds/Intel_Linux_NVMe_Guide_330602-002.pdf)**

Retrieved from "https://wiki.archlinux.org/index.php?title=Solid_State_Drives/NVMe&oldid=508815"

- This page was last edited on 28 January 2018, at 23:25.
- Content is available under GNU Free Documentation License 1.3 or later unless otherwise noted.