# DAOS System Administration

## System Monitoring

System monitoring and telemetry data will be provided as part of the control plane and will be documented in a future revision.

## Storage Operations

### Per-Storage-Server Space Utilization

To query SCM and NVMe storage space usage and show how much space is available to create new DAOS pools with, run the following command:

bash-4.2$ dmg storage query usage  
Hosts SCM-Total SCM-Free SCM-Used NVMe-Total NVMe-Free NVMe-Used  
----- --------- -------- -------- ---------- --------- ---------  
wolf-71 6.4 TB 2.0 TB 68 % 1.5 TB 1.1 TB 27 %  
wolf-72 6.4 TB 2.0 TB 68 % 1.5 TB 1.1 TB 27 %

The command output shows online DAOS storage utilization, only including storage statistics for devices that have been formatted by DAOS control-plane and assigned to a currently running rank of the DAOS system. This represents the storage that can host DAOS pools.

Note that the table values are per-host (storage server) and SCM/NVMe capacity pool component values specified in [dmg pool create](https://daos-stack.github.io/admin/pool_operations/#pool-creationdestroy) are per rank. If multiple ranks (I/O processes) have been configured per host in the server configuration file [daos\_server.yml](https://github.com/daos-stack/daos/blob/master/utils/config/daos_server.yml) then the values supplied to dmg pool create should be a maximum of the SCM/NVMe free space divided by the number of ranks per host.

For example if 2.0 TB SCM and 10.0 TB NVMe free space is reported by dmg storage query usage and the server configuration file used to start the system specifies 2 I/O processes (2 “server” sections), the maximum pool size that can be specified is approximately dmg pool create -s 1T -n 5T (may need to specify slightly below the maximum to take account of negligible metadata overhead).

### NVMe SSD Health Monitoring

Useful admin dmg commands to query NVMe SSD health:

* Query Per-Server Metadata:
  + dmg storage query (list-devices|list-pools)
  + dmg storage scan --nvme-meta shows mapping of metadata to NVMe controllers

The NVMe storage query list-devices and list-pools commands query the persistently stored SMD device and pool tables respectively. The device table maps the internal device UUID to attached VOS target IDs. The rank number of the server where the device is located is also listed, along with the current device state. The current device states are the following: - NORMAL: a fully, functional device in-use by DAOS - EVICTED: the device is no longer in-use by DAOS - UNPLUGGED: the device is currently unplugged from the system (may or not be evicted) - NEW: the device is plugged and available, and not currently in-use by DAOS

The transport address is also listed for the device. This is either the PCIe address for normal NVMe SSDs, or the BDF format address of the backing NVMe SSDs behind a VMD (Volume Management Device) address. In the example below, the last two listed devices are both VMD devices with transport addresses in the BDF format behind the VMD address 0000:5d:05.5.

The pool table maps the DAOS pool UUID to attached VOS target IDs and will list all of the server ranks that the pool is distributed on. With the additional verbose flag, the mapping of SPDK blob IDs to VOS target IDs will also be displayed.

$ dmg -l boro-11,boro-13 storage query list-devices  
-------  
boro-11  
-------  
 Devices  
 UUID:5bd91603-d3c7-4fb7-9a71-76bc25690c19 [TrAddr:0000:8a:00.0]  
 Targets:[0 2] Rank:0 State:NORMAL  
 UUID:80c9f1be-84b9-4318-a1be-c416c96ca48b [TrAddr:0000:8b:00.0]  
 Targets:[1 3] Rank:0 State:NORMAL  
 UUID:051b77e4-1524-4662-9f32-f8e4d2542c2d [TrAddr:0000:8c:00.0]  
 Targets:[] Rank:0 State:NEW  
 UUID:81905b24-be44-4106-8ff9-03002e9dd86a [TrAddr:5d0505:01:00.0]  
 Targets:[0 2] Rank:1 State:EVICTED  
 UUID:2ccb8afb-5d32-454e-86e3-762ec5dca7be [TrAddr:5d0505:03:00.0]  
 Targets:[1 3] Rank:1 State:NORMAL

$ dmg -l boro-11,boro-13 storage query list-pools  
-------  
boro-11  
-------  
 Pools  
 UUID:08d6839b-c71a-4af6-901c-28e141b2b429  
 Rank:0 Targets:[0 1 2 3]  
 Rank:1 Targets:[0 1 2 3]  
  
$ dmg -l boro-11,boro-13 storage query list-pools --verbose  
-------  
boro-11  
-------  
 Pools  
 UUID:08d6839b-c71a-4af6-901c-28e141b2b429  
 Rank:0 Targets:[0 1 2 3] Blobs:[4294967404 4294967405 4294967407 4294967406]  
 Rank:1 Targets:[0 1 2 3] Blobs:[4294967410 4294967411 4294967413 4294967412]

* Query Storage Device Health Data:
  + dmg storage query (device-health|target-health)
  + dmg storage scan --nvme-health shows NVMe controller health stats

The NVMe storage query device-health and target-health commands query the device health data, including NVMe SSD health stats and in-memory I/O error and checksum error counters. The server rank and device state are also listed. The device health data can either be queried by device UUID (device-health command) or by VOS target ID along with the server rank (target-health command). The same device health information is displayed with both command options.

$ dmg -l boro-11 storage query device-health  
 --uuid=5bd91603-d3c7-4fb7-9a71-76bc25690c19  
or  
$ dmg -l boro-11 storage query target-health  
 --rank=0 --tgtid=0  
-------  
boro-11  
-------  
 Devices  
 UUID:5bd91603-d3c7-4fb7-9a71-76bc25690c19 [TrAddr:0000:8a:00.0]  
 Targets:[0 1 2 3] Rank:0 State:NORMAL  
 Health Stats:  
 Temperature:289K(15C)  
 Controller Busy Time:0s  
 Power Cycles:0  
 Power On Duration:0s  
 Unsafe Shutdowns:0  
 Media Errors:0  
 Read Errors:0  
 Write Errors:0  
 Unmap Errors:0  
 Checksum Errors:0  
 Error Log Entries:0  
 Critical Warnings:  
 Temperature: OK  
 Available Spare: OK  
 Device Reliability: OK  
 Read Only: OK  
 Volatile Memory Backup: OK

### NVMe SSD Eviction and Hotplug

* Manually Evict an NVMe SSD: dmg storage set nvme-faulty

To manually evict an NVMe SSD (auto eviction will be supported in a future release), the device state needs to be set to “FAULTY” by running the following command:

$ dmg -l boro-11 storage set nvme-faulty --uuid=5bd91603-d3c7-4fb7-9a71-76bc25690c19  
-------  
boro-11  
-------  
 Devices  
 UUID:5bd91603-d3c7-4fb7-9a71-76bc25690c19 Targets:[] Rank:1 State:FAULTY

The device state will transition from “NORMAL” to “FAULTY” (shown above), which will trigger the faulty device reaction (all targets on the SSD will be rebuilt and the SSD will remain evicted until device replacement occurs).

**Full NVMe hot plug capability will be available and supported in DAOS 2.0 release. Use is currently intended for testing only and is not supported for production.**

* Replace an Evicted SSD with a New Device: dmg storage replace nvme

To replace an NVMe SSD with an evicted device and reintegrate it into use with DAOS, run the following command:

$ dmg -l boro-11 storage replace nvme --old-uuid=5bd91603-d3c7-4fb7-9a71-76bc25690c19 --new-uuid=80c9f1be-84b9-4318-a1be-c416c96ca48b  
-------  
boro-11  
-------  
 Devices  
 UUID:80c9f1be-84b9-4318-a1be-c416c96ca48b Targets:[] Rank:1 State:NORMAL

The old, now replaced device will remain in an “EVICTED” state until it is unplugged. The new device will transition from a “NEW” state to a “NORMAL” state (shown above).

* Reuse a FAULTY Device: dmg storage replace nvme

In order to reuse a device that was previously set as FAULTY and evicted from the DAOS system, an admin can run the following command (setting the old device UUID to be the new device UUID):

$ dmg -l boro-11 storage replace nvme --old-uuid=5bd91603-d3c7-4fb7-9a71-76bc25690c19 --new-uuid=5bd91603-d3c7-4fb7-9a71-76bc25690c19  
-------  
boro-11  
-------  
 Devices  
 UUID:5bd91603-d3c7-4fb7-9a71-76bc25690c19 Targets:[] Rank:1 State:NORMAL

The FAULTY device will transition from an “EVICTED” state back to a “NORMAL” state, and will again be available for use with DAOS. The use case of this command will mainly be for testing, or for accidental device eviction.

### NVMe SSD Identification

The SSD identification feature is simply a way to quickly and visually locate a device. It requires the use of Intel VMD (Volume Management Device), which needs to be physically available on the hardware as well as enabled in the system BIOS. The feature supports two LED device events: locating a healthy device and locating an evicted device.

* Locate a Healthy SSD: dmg storage identify vmd

To quickly identify an SSD in question, an administrator can run the following command:

$ dmg -l boro-11 storage identify vmd --uuid=6fccb374-413b-441a-bfbe-860099ac5e8d  
  
If a non-VMD device UUID is used with the command, the following error will occur:  
localhost DAOS error (-1010): DER\_NOSYS

The status LED on the VMD device is now set to an “IDENTIFY” state, represented by a quick, 4Hz blinking amber light. The device will quickly blink by default for about 60 seconds and then return to the default “OFF” state.

* Locate an Evicted SSD:

If an NVMe SSD is evicted, the status LED on the VMD device is set to a “FAULT” state, represented by a solidly ON amber light. No additional command apart from the SSD eviction command would be needed, and this would visually indicate that the device needs to be replaced and is no longer in use by DAOS. The LED of the VMD device would remain in this state until replaced by a new device.

## System Operations

The DAOS Control Server acting as the access point records details of DAOS I/O Server instances that join the DAOS system. Once an I/O Engine has joined the DAOS system, it is identified by a unique system “rank”. Multiple ranks can reside on the same host machine, accessible via the same network address.

A DAOS system can be shutdown and restarted to perform maintenance and/or reboot hosts. Pool data and state will be maintained providing no changes are made to the rank’s metadata stored on persistent memory.

Storage reformat can also be performed after system shutdown. Pools will be removed and storage wiped.

System commands will be handled by the DAOS Server listening at the access point address specified as the first entry in the DMG config file “hostlist” parameter. See [daos\_control.yml](https://github.com/daos-stack/daos/blob/master/utils/config/daos_control.yml) for details.

The “access point” address should be the same as that specified in the server config file [daos\_server.yml](https://github.com/daos-stack/daos/blob/master/utils/config/daos_server.yml) specified when starting daos\_server instances.

!!! warning Controlled start/stop/reformat have some known limitations. Whilst individual system instances can be stopped, if a subset is restarted, existing pools will not be automatically integrated with restarted instances.

### Query

The system membership can be queried using the command:

$ dmg system query [--verbose] [--ranks <rankset>|--host-ranks <hostset>]

* <rankset> is a pattern describing rank ranges e.g. 0,5-10,20-100
* <hostset> is a pattern describing host ranges e.g. storagehost[0,5-10],10.8.1.[20-100]
* --verbose flag gives more information on each rank

Output table will provide system rank mappings to host address and instance UUID, in addition to rank state.

### Shutdown

When up and running, the entire system can be shutdown with the command:

$ dmg system stop [--force] [--ranks <rankset>|--host-ranks <hostset>]

* <rankset> is a pattern describing rank ranges e.g. 0,5-10,20-100
* <hostset> is a pattern describing host ranges e.g. storagehost[0,5-10],10.8.1.[20-100]

Output table will indicate action and result.

DAOS Control Servers will continue to operate and listen on the management network.

### Start

To start the system after a controlled shutdown run the command:

$ dmg system start [--ranks <rankset>|--host-ranks <hostset>]

* <rankset> is a pattern describing rank ranges e.g. 0,5-10,20-100
* <hostset> is a pattern describing host ranges e.g. storagehost[0,5-10],10.8.1.[20-100]

Output table will indicate action and result.

DAOS I/O Engines will be started.

### Reformat

To reformat the system after a controlled shutdown run the command:

$ dmg storage format --reformat

* --reformat flag indicates that a reformat operation should be performed disregarding existing filesystems
* if no record of previously running ranks can be found, reformat is performed on hosts in dmg config file hostlist
* if system membership has records of previously running ranks, storage allocated to those ranks will be formatted

Output table will indicate action and result.

DAOS I/O Engines will be started and all DAOS pools will have been removed.

### Manual Fresh Start

To reset the DAOS metadata across all hosts, the system must be reformatted. First, ensure all daos\_server processes on all hosts have been stopped, then for each SCM mount specified in the config file (scm\_mount in the servers section) umount and wipe FS signatures.

Example illustration with two IO instances specified in the config file:

* clush -w wolf-[118-121,130-133] umount /mnt/daos1
* clush -w wolf-[118-121,130-133] umount /mnt/daos0
* clush -w wolf-[118-121,130-133] wipefs -a /dev/pmem1
* clush -w wolf-[118-121,130-133] wipefs -a /dev/pmem0
* Then restart DAOS Servers and format.

### Fault Domain Maintenance and Reintegration

Details on how to drain an individual storage node or fault domain (e.g. rack) in preparation for maintenance activity and how to reintegrate it will be provided in a future revision.

### DAOS System Extension

Ability to add new DAOS server instances to a pre-existing DAOS system will be documented in a future revision.

## Fault Management

DAOS relies on massively distributed single-ported storage. Each target is thus effectively a single point of failure. DAOS achieves availability and durability of both data and metadata by providing redundancy across targets in different fault domains.

### Fault Detection & Isolation

DAOS servers are monitored within a DAOS system through a gossip-based protocol called SWIM[[1]](#footnote-40) that provides accurate, efficient, and scalable server fault detection.

Storage attached to each DAOS target is monitored through periodic local health assessment. Whenever a local storage I/O error is returned to the DAOS server, an internal health check procedure will be called automatically. This procedure makes an overall health assessment by analyzing the IO error code and device SMART/Health data. If the result is negative, the target will be marked as faulty, and further I/Os to this target will be rejected and re-routed.

Once detected, the faulty target or servers (effectively a set of targets) must be excluded from each pool membership. This process is triggered either manually by the administrator or automatically (see the next section for more information). Upon exclusion from the pool map, each target starts the collective rebuild process automatically to restore data redundancy. The rebuild process is designed to operate online while servers continue to process incoming I/O operations from applications.

Tools to monitor and manage rebuild are still under development.

### Rebuild Throttling

The rebuild process may consume many resources on each server and can be throttled to reduce the impact on application performance. This current logic relies on CPU cycles on the storage nodes. By default, the rebuild process is configured to consume up to 30% of the CPU cycles, leaving the other 70% for regular I/O operations.

During the rebuild process, the user can set the throttle to guarantee that the rebuild will not use more resources than the user setting. The user can only set the CPU cycle for now. For example, if the user set the throttle to 50, then the rebuild will at most use 50% of the CPU cycle to do the rebuild job. The default rebuild throttle for CPU cycle is 30. This parameter can be changed via the daos\_mgmt\_set\_params() API call and will be eventually available through the management tools.

## Software Upgrade

Interoperability in DAOS is handled via protocol and schema versioning for persistent data structures. Further instructions on how to manage DAOS software upgrades will be provided in a future revision.

### Protocol Interoperability

Limited protocol interoperability is provided by the DAOS storage stack. Version compatibility checks will be performed to verify that:

* All targets in the same pool run the same protocol version.
* Client libraries linked with the application may be up to one protocol version older than the targets.

If a protocol version mismatch is detected among storage targets in the same pool, the entire DAOS system will fail to start up and will report failure to the control API. Similarly, the connection from clients running a protocol version incompatible with the targets will return an error.

### Persistent Schema Compatibility and Update

The schema of persistent data structures may evolve from time to time to fix bugs, add new optimizations, or support new features. To that end, the persistent data structures support schema versioning.

Upgrading the schema version will not be performed automatically and must be initiated by the administrator. A dedicated upgrade tool will be provided to upgrade the schema version to the latest one. All targets in the same pool must have the same schema version. Version checks are performed at system initialization time to enforce this constraint.

To limit the validation matrix, each new DAOS release will be published with a list of supported schema versions. To run with the new DAOS release, administrators will then need to upgrade the DAOS system to one of the supported schema versions. New pool shards will always be formatted with the latest version. This versioning schema only applies to a data structure stored in persistent memory and not to block storage that only stores user data with no metadata.

## Storage Scrubbing

Support for end-to-end data integrity is planned for DAOS v1.2 and background checksum scrubbing for v2.2. Once available, that functionality will be documented here.

1. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1028914 [↑](#footnote-ref-40)