Hewlett Packard Enterprise

TSC Update & DAOS Community Roadmap

Johann Lombardi
Senior Distinguished Technologist, HPE
Chair of Technical Steering Committee, DAOS Foundation

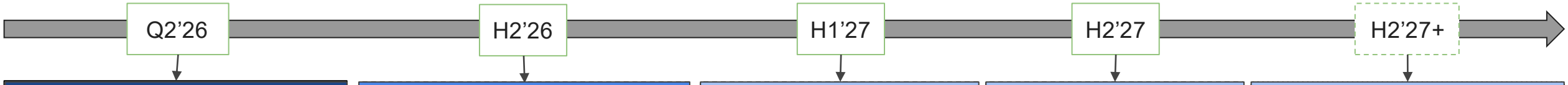
May 21, 2026
Virtual DAOS User Group

Technical Steering Committee (TSC)

- Meet once a week
  - Rotating schedule to accommodate all geos
- Define roadmap & review release tickets/blockers
- Discuss technical topics

DAOS Community Roadmap

Color coding schema:
- Committed (or released) release/features
- In-planning release/features
- Future possible release/features



DAOS 2.8
(DAOS Foundation Release)

OS Packages:
- Leap 15.6/7
- RHEL/Rocky/Alma 9.5/9.6
- New packaging

Networking:
- DOCA-OFED support

Features:
- Catastrophic recovery tool improvement
- Better defaults (e.g. RF3, cell size, …)
- Rebuild bug fixing and perf improvement
- Explicit rebuild control
- Improved allocator for non-PMem phase 1
- PyTorch data loader

Tech preview:
- Non-PMem support phase 2
- Rolling upgrade preparation

UX Improvements:
- Reintegration of all pools
- daos pool listing

DAOS 3.0
(DAOS Foundation Release)

OS Packages:
- Leap 15.7 (x86_64)
- RHEL/Rocky/Alma 9.x (x86_64)
- Ubuntu 24.04 client (x86_64/ARM64)

Features:
- Non-PMem support phase 2
- Catastrophic recovery improvement final
- Protocol changes for multi-provider
- Hard links support in libdfs/dfuse

Tech preview:
- Incremental rebuild/reintegration

DAOS 3.2
(DAOS Foundation Release)

Features:
- SSD hotplug & LED without VMD
- Mount POSIX snapshots RO
- Incremental rebuild/reintegration
- Network multipath

Tech preview:
- SSD encryption support via SED

DAOS 3.4
(DAOS Foundation Release)

Features:
- Rolling upgrade
- Fault domains beyond servers
- Multi-provider support

Tech preview:
- QLC support

Future Releases
(DAOS Foundation Release)

Features:
- Pool resizing
- Inline compression
- Inline encryption
- Inline deduplication
- Middleware consistency checker
- Progressive layout
- Pipeline API
- SQL support with predicate pushdown
- Distributed transactions
- Pool/container freeze
- Tiered container / HSM
- Support for multiple DAOS systems
- multi-NIC support per engine/process
- Container parking/serialization
- WORM containers
- flock support
- per-pool agent certificate

Git Branch Status

- release/2.6 = upcoming 2.6.5 version (HPE)
- master = future 3.0 release (DAOS Foundation)
  - Test builds using 2.9.x numbering
- release/2.8 = upcoming 2.8.0 version (DAOS Foundation)

Better Defaults (2.8)

- Lessons learnt from the Aurora deployment
  - More reliability by default, can be relaxed via explicit settings
- Pool's property changes
  - RF3 as the default on the pool
    - If enough nodes/fault domains available, downgraded to the number of nodes/fault domains otherwise
  - Reserved space for rebuilds (space_rb) increased from 0% to 5%
  - Default EC Cell Size (ec_cell_sz) increased from 64KiB to 128KiB
- Container's property changes
  - Checksum enabled by default
    - crc64 checksum type
    - Including server-side checksum verification on write
- More to come …
  - Use POSIX container type by default on container creation (no need for the -t POSIX anymore)
  - Change default rebuild throttling to reduce impact on I/Os
  - Enable checksum scrubber by default? (once stabilized)

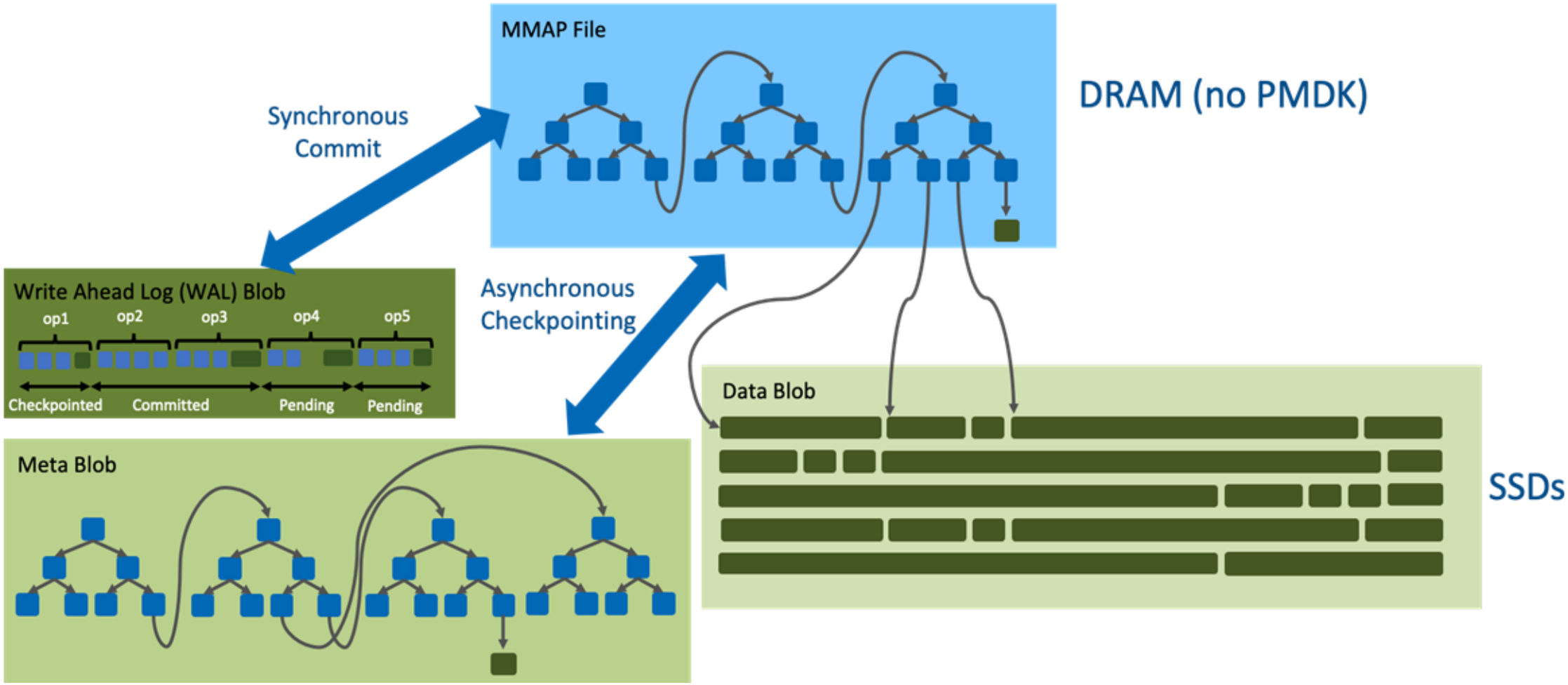
New Packaging (2.8)

- Generate all DAOS packages and dependencies as part of the DAOS build
  - Via docker
  - Some packages have been renamed with the daos- prefix
- Can generate packages for various distribution via docker
  - E.g. for ubuntu packages
    docker build . -f ./utils/docker/Dockerfile.ubuntu -t daos --build-arg DAOS_KEEP_SRC=yes --build-arg DEPS_JOBS=10

Self-healing and Explicit Rebuild Control (2.8)

- Self-Healing Policy Controls (self_heal properties)
  - Introduces the self_heal system property (to work with the existing self_heal property) for granular control over auto-recovery behaviors
  - Allows administrators to explicitly enable and disable self-healing policies at the system and per-pool levels
  - Supports "exclude-only" policies, allowing the system to immediately exclude failed targets from I/O without automatically triggering a resource-intensive rebuild
- Explicit / Interactive Rebuild Control
  - Provides administrators with new commands to manually start, stop, and manage rebuild operations on demand
  - Resource Management: Operators can now stagger overall recovery by manually triggering rebuilds in smaller batches, preventing simultaneous system-wide rebuilds from saturating CPU and network resources
  - Safely handles reclaim behaviors during manual rebuild stops to ensure storage space is efficiently freed up
- Enhanced Rebuild Monitoring & Visibility
  - Exposes a new "Data redundancy" field in the dmg/daos pool query output to easily identify if a pool is in a *normal* or *degraded* state
  - Enables administrators to easily verify that a target exclusion has been successfully completed before manually initiating an interactive rebuild

Non-PMEM Support Overview



Non-PMEM Support (2.8 & 3.0)

- Supported in Version 2.8 (Phase 2):
  - Basic bucket and object eviction functionality
  - Dynamic sizing of the non-evictable zone (expands on demand up to the VOS file size limit)
- Reference Metrics (For context):
  - Bucket / Page Size: 16MB
  - Bucket Load Time: ~4ms
  - Single Bucket Capacity: Ranges from 550MB (at 4k IO) to 137GB (at 1M IO)
- Version 2.8 Limitations to be addressed in 3.0:
  - Evictable Bucket Constraint: Strictly limited to a maximum of one evictable bucket per object shard
  - Large Object Spillover: Data exceeding the single evictable bucket overflows into non-evictable zones
  - Premature ENOSPACE Errors: Spillover causes rapid consumption of non-evictable zones, leading to errors even when evictable space is available
  - Object Fragmentation: Ongoing I/O and aggregation cause fragmentation, leading to performance degradation over time

Hard links Support (3.0)

- While soft links are supported, hard links haven't
  - Allowed us to store inode attributes into parent directory entry
  - Remove an indirection for better performance
- Why adding support for it now?
  - Some python package manager heavily uses hard links
  - Some backup utilities
- Use a separate inode table, but just for hard links
  - Don't impact performance on the most common case when creating regular files
  - Use a global inode table to store inodes for files with hard links
  - Layout change done on the fly when a new link is added, using distributed transactions for consistency
- See https://daosio.atlassian.net/wiki/spaces/DC/pages/12942114817/Support+for+hardlinks+in+DAOS+filesystem

Incremental Reintegration (3.2)

- When an engine is excluded and then reintegrated, the whole object history is recreated
  - Local data is deleted
  - No real option if local SSDs were impacted (e.g. reformatted or replaced)
  - Slow process by definition since N:1 communication pattern
- Why not rely on epoch to reapply all missed updates since the exclusion?
  - Accelerate reintegration
  - More resilience to spurious exclusion (e.g. unstable networks)
  - Ideally instantaneous reintegration if no updates were made while the node was excluded
- Challenges
  - Object punch + aggregation
    - history completely deleted
    - how to know the object should be deleted on the reintegrated node?
  - When to give up and fall back to full reintegration?
  - Testing

Network Multipath Requirements (3.2)

- Configuration:
  - 1 server with multiple NICs running one or multiple engine(s)
  - 1 compute node with multiple NICs running one or multiple process(es)
- Each engine and client process should be able to take a list of NICs, all running under the same provider
- The fact that multiple routes might be used under the hood should be invisible to libdaos/engine
  - RPC time out and resent can be experienced
- 2 use cases switched via environment variables
- Use case 1: Resilience
  - All traffic is sent through one primary NIC
  - Alternative route used when primary route is offline/timing out
  - Fail back to primary route should be automatic
- Use case 2: Resilience + Performance
  - Traffic is round-robin across all the NICs
  - When one route is offline/timing out, use an alternative route
  - Offline routes that become active again should be re-enabled automatically
- An engine that has at least one working route should (use case 1 and 2):
  - NOT be evicted by SWIM
  - Be eventually reachable by any nodes in the DAOS system (servers and clients)
- Use case 2 should get aggregated bandwidth equal to the sum of all the configured NICs

Multipath Design Consideration (3.2)

- CART to expose a single context, and multiplex it under the hood over multiple Mercury contexts
  - libdaos/engine to progress only one CART context
- Fine to:
  - Bind RPC to a NIC at RPC creation time
  - Resent through an alternative route require RPC timeout & resent for libdaos/engine
- SWIM
  - Ping via all the routes in parallel
  - If got a reply via one route, node is considered as alive
- Route liveness map
  - Maintain in rank-to-endpoint mapping table information about route status
- Piggy-backing on RPC
  - Server to client/server: reporting working/failing server endpoints based on traffic received by the server

Other Features for Consideration from Enakta (not in roadmap yet)

- Go API bindings
  - Interception libraries not functional with Go application (no libc)
  - Can be helpful to integrate with services like Versity S3 Gateway
- Python fsspec integration
  - Filesystem interface python
  - Provide a python native integration for libdfs
- HSM/Tiering integration
  - Handled externally with minimal changes to DAOS
  - Need to be evaluated
- New kernel interface
  - Potential replacement for dfuse
  - Using its own mechanism for kernel/userspace communication, not relying on dfuse
  - To be evaluated and benchmarked against dfuse

Notable Community Efforts

- Evaluation of running the DAOS client stack on a DPU (HPE Labs)
  - https://arxiv.org/pdf/2509.13997
- Splitting the control plane into two processes (Alibaba)
  - Separating the raft engine from the RPC processing
  - Improve resilience and upgrade
  - Discussed at the TSC
- Per-pool certificate for multi-tenancy (HPE)
  - Restrict what pools are visible to a client node
  - https://daosio.atlassian.net/wiki/spaces/~7120209e5689b1088c45d3a4206c5d3adf73a7/pages/12946833410/Client+Node+Authentication+Design
- Support multiple authentication type (HPE)
  - https://daosio.atlassian.net/wiki/spaces/DC/pages/12796362760/Multiple+Authentication+Types

Resources

- Foundation website: https://daos.io/
- Github: https://github.com/daos-stack/dao
- Online doc: https://docs.daos.io
- Mailing list & slack: https://daos.groups.io
- YouTube channel: http://video.daos.io
- DAOS User Group on Nov 16, 2025: https://daos.io/event/9th-daos-user-group-dug-at-sc25