

Modern Developments with NFS in Linux

Jeff Layton <jlayton@kernel.org>



LINUX

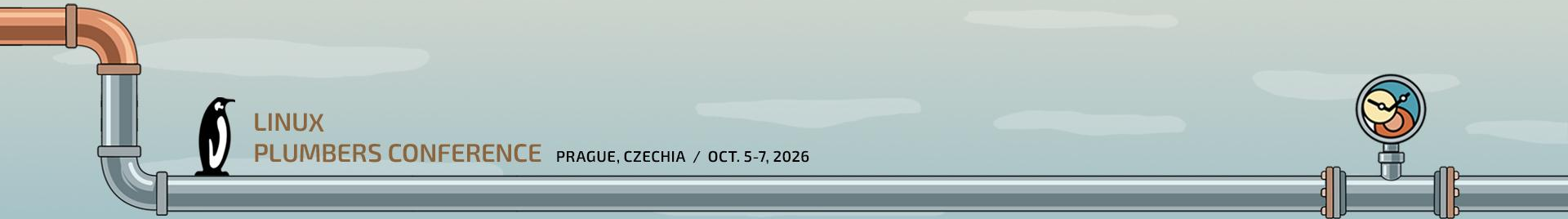
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

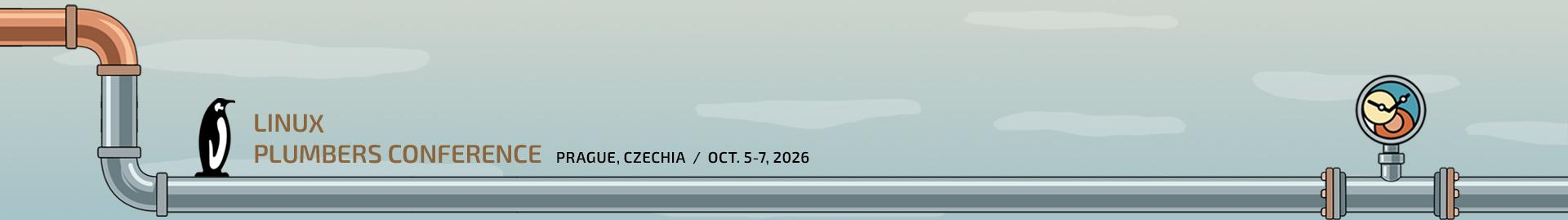


Syllabus

- Scalability and Performance
- New Features and Protocol Work
- Control Plane Enhancements
- Tooling, Testing & Development Community

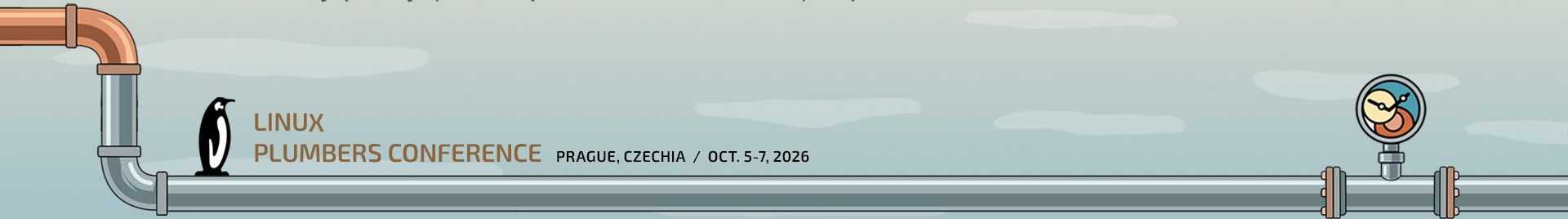


Scalability and Performance



LOCALIO: Same-Host Fast Path

- **Purpose:** Reduce NFS I/O overhead in some common deployment scenarios
- **Primary target:** Containerized workloads where NFSD and client run on the same node
- **Contributor:** Mike Snitzer, Hammerspace
- **In operation:**
 - Simple auxiliary protocol enables a Linux NFS client to detect reliably that it is on the same physical host as NFSD
 - When co-located, READ/WRITE/COMMIT operations bypass XDR, RPC, and network entirely
 - Security policy (i.e., export access controls) is preserved



NFSv4 Asynchronous COPY Re-enabled

- **Purpose:** Avoid moving large files over the network unnecessarily
 - Long-running COPY operations signal completion via callback
- **Primary target:** Bulk copies of large files within or between servers
- **Contributor:** Olga Kornievskaja, Red Hat
- **Attack surfaces addressed:**
 - Clients could start an unlimited number of async COPY operations
- **Client and server bugs addressed:**
 - Completion notification was not reliable during network partition
 - NFSv4.2 protocol issues
 - Clarifications and errata compiled in draft-cel-nfsv4-copy-implementation-experience



LINUX

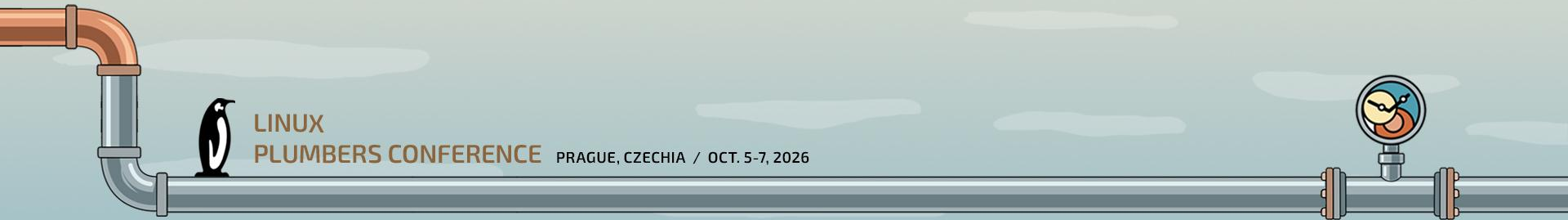
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



I/O Cache Controls: Reduce Pagecache Usage

- **Purpose:** Avoid thrashing, memory reclaim, and some data copy operations
- **Primary target:** Resident set larger than server RAM
- **Contributor:** Mike Snitzer, Hammerspace
- **In operation:**
 - Debugfs switch selects buffered, DONTCACHE, or direct I/O (no admin interface yet)
 - Falls back to buffered if the FS lacks the newer cache modes
 - NB: Use kernel v7.2 or newer if using DONTCACHE for writes for improved writeback



Which mode should you use?

Situation	Best Mode
Seq. bulk writes	dontcache or direct
Random Writes	direct
Many Concurrent Readers	dontcache or direct
Writes + cache missing reads (AI workload)	direct
Writes + readers that fit in server RAM	mixed
Never	dontcache on reads



LINUX

PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Increased Implementation Limits

- **Purpose:** Make NFSv4 COMPOUNDS more capable to reduce network roundtrips
- **Primary target:** NFS clients on WANs
- **Contributor:** Chuck Lever, Oracle
- **In operation:**
 - Server threads now carry up to 8MB of buffer pages
 - Advertised I/O block size has been increased from 1MB to 4MB
 - NFSv4.1 CREATE_SESSION ca_maxrequestsize and ca_maxresponsesize have been increased as well
 - Maximum operations per NFSv4 COMPOUND has been increased to 200



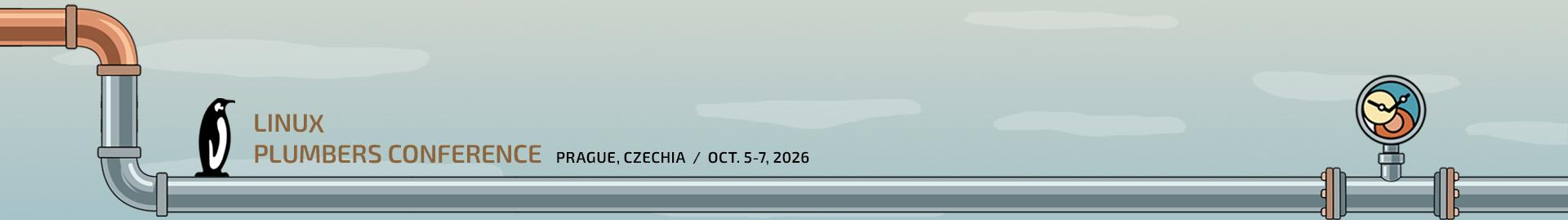
LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Dynamic Thread Pool Sizing

- **Purpose:** Fewer idle resources to handle larger workload bursts
- **Primary target:** Servers with bursty or highly variable workloads
- **Contributor:** Jeff Layton, Meta; Neil Brown, hobbyist
- **In operation:**
 - thread pool target counts set via 'nfsdctl threads --min-threads'
 - Light workloads stay near minimum thread count
 - Heavier workloads can create new threads on demand
 - As workload decreases, threads exit



Global nfsd_mutex breakup

- **Purpose:** better scalability and container isolation
- **Primary target:** containerized NFS server workloads
- **Contributor:** Jeff Layton, Meta
- **In operation:**
 - Most per-net data structures moved under new per-net mutex
 - nfsd's filecache moved under new global mutex
 - global_nfsd_mutex still protects few remaining global data structures



LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



New Features and Protocol Work



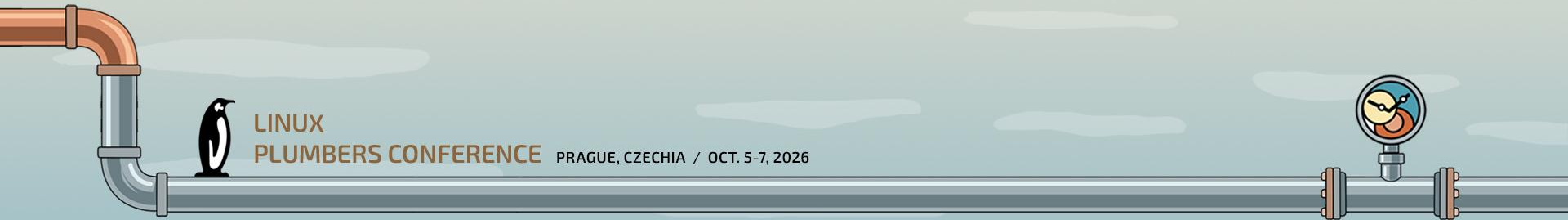
LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



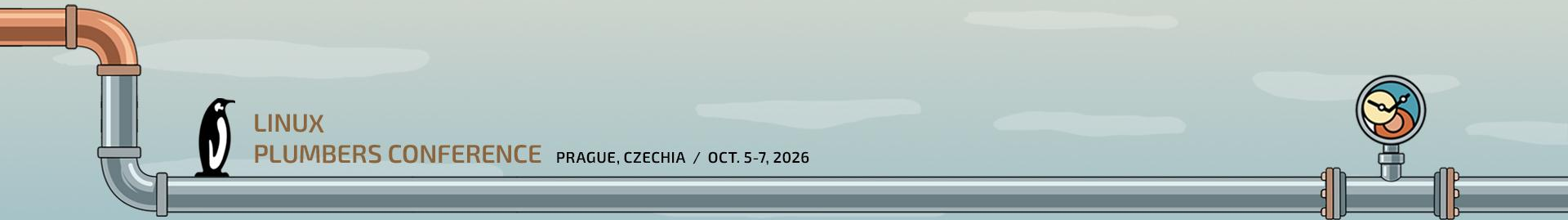
Multigrain Timestamps

- **Purpose:** Reliable cache-coherency using inode m/ctime
- **Primary Target:** Servers serving local Linux filesystems
- **Contributor:** Jeff Layton, Meta
- **In Operation:**
 - Track when the m/ctime of an inode has been queried via `->getattr`.
 - Ensure that any change to the inode results in an observable change to the m/ctime.
 - Implemented for btrfs, ext4, xfs and tmpfs (more on the way)



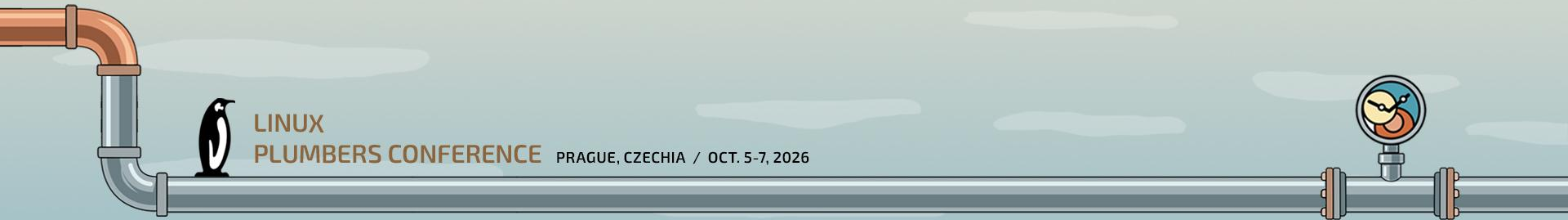
NFSv4.2 Attribute Delegation

- **Purpose:** Reduce network round-trips needed to meet POSIX compliance requirements
- **Primary target:** Common POSIX applications that stat(2) files with dirty data
- **Contributors:** Jeff Layton, Meta
- **In operation:**
 - NFSv4 OPEN returns a new delegation type
 - Client is empowered to manage file size and mtime attributes without consulting server
 - Based on RFC 9754



NFSv4 Write Delegations

- **Purpose:** Enable clients to cache aggressively in more situations
- **Primary target:** Single-client write-intensive workloads
- **Contributor:** Dai Ngo, Oracle
- **In operation:**
 - Write delegation extended to O_WRONLY opens
 - Write delegation extended to LAYOUTGET



NFSv4.1 Directory Delegations

- **Purpose:** Enable clients to cache directory entries more aggressively
- **Primary target:** Single-client directory metadata-intensive workloads
- **Contributor:** Jeff Layton, Meta / Anna Schumacher, Hammerspace
- **In operation:**
 - GET_DIR_DELEGATION implemented
 - CB_NOTIFY callbacks when directory contents change
 - Server supports this fully (except for CHILD_ATTR notifications)
 - Client has support for GET_DIR_DELEGATION. CB_NOTIFY support is in progress.



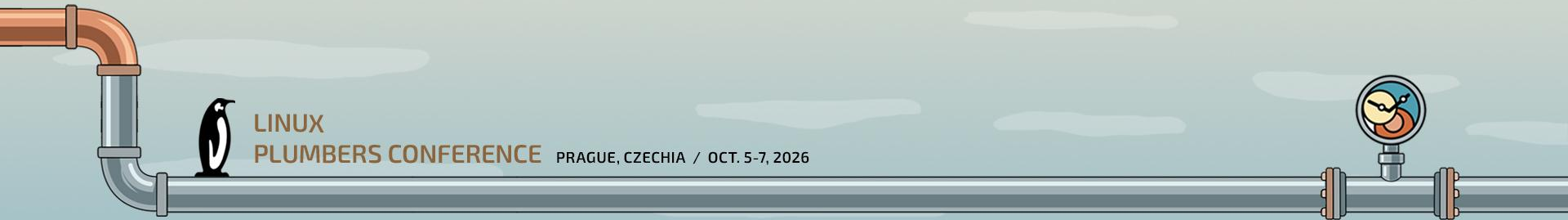
LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



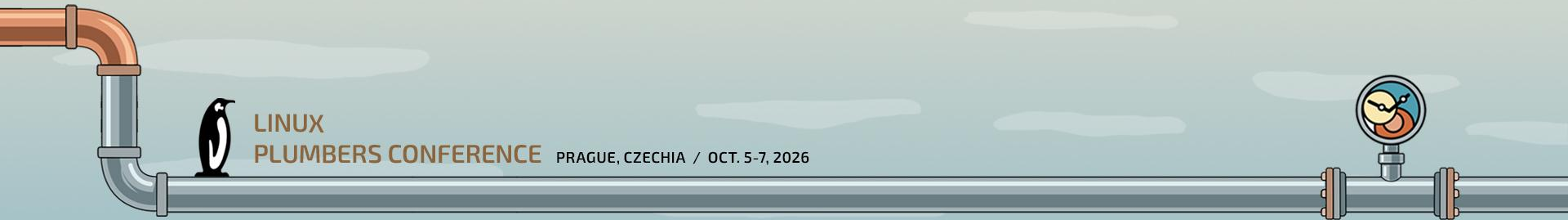
NFSv4 Forechannel: Dynamic Session Resizing

- **Purpose:** Better control of client's server-side resource consumption
- **Primary target:** Resource-constrained servers
- **Contributor:** Neil Brown, hobbyist
- **In operation:**
 - CREATE_SESSION starts with a small number of slots
 - Slot count increases and decreases with workload and server memory availability



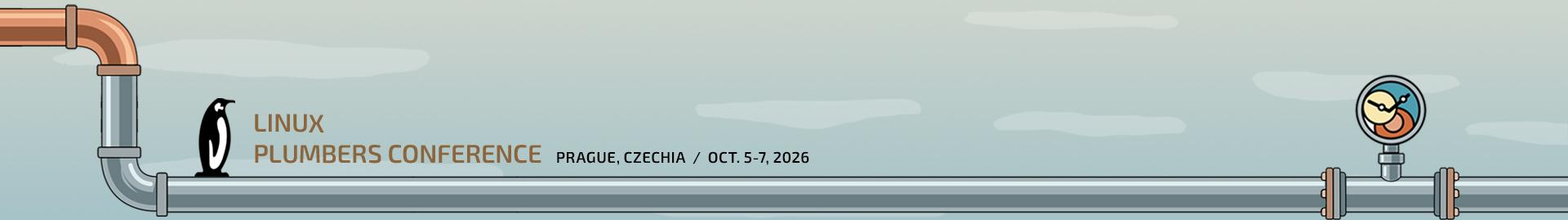
NFSv4 Backchannel Session Scalability

- **Purpose:** Increase NFSv4 callback throughput with less inter-client contention
- **Primary target:** Workloads that trigger significant callback traffic
- **Contributor:** Jeff Layton, Meta
- **In operation:**
 - Instead of a single workqueue, one workqueue per client
 - Instead of one slot, 32 slots
 - Delegation and layout recall, CB_NOTIFY, and CB_OFFLOAD now handled in parallel



POSIX ACL Extension for NFSv4.2

- **Purpose:** Give clients direct access to POSIX ACLs on POSIX-only NFSv4 servers
- **Primary target:** Servers that do not implement NFSv4 ACLs natively
- **Contributor:** Rick Macklem, hobbyist
- **In operation:**
 - New fattr4 attributes enable SETATTR/GETATTR of POSIX ACLs on files



Signed NFS Filehandles

- **Purpose:** Deter filehandle guessing attacks
- **Primary target:** Admins who need to strengthen restrictive POSIX directory permissions
- **Contributor:** Benjamin Coddington, Hammerspace
- **In operation:**
 - NFSD administrator sets a private signing key
 - When the `sign_fh` option is present on an export:
 - Egress filehandles for that export include a signature (siphash-2-4)
 - NFSD verifies the signature on ingress filehandles for that export



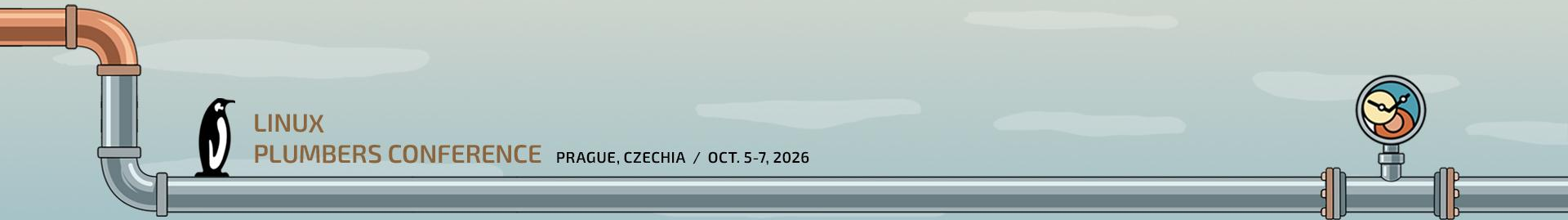
LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Case-Folding: Reporting Export Settings

- **Purpose:** Report export's case-folding settings
- **Primary target:** Non-POSIX NFS clients (Windows)
- **Contributor:** Chuck Lever, Oracle
- **In operation:**
 - NFSv3 reports case-folding settings for an entire FSID
 - NFSv4 reports case-folding settings for the queried object's parent directory



NFS Re-export: NFSD as a Caching Gateway

- **Purpose:** Turn NFSD into a caching gateway — re-export + FS-Cache
- **Primary target:** Read-heavy fan-out — rendering, HPC, EDA
- **In operation:**
 - nfs-kernel-server re-exports an upstream NFS service; cachefilesd caches to RAM + NVMe
 - Data crosses the high-latency link once, then serves a local client fleet at speed
- **Shipping today — cloud and on-prem:**
 - AWS KNFSD File Cache — Apache-2.0, Preview (Jul 2026)
 - Google Cloud knfsd-cache-utils — HPC / burst-to-cloud
 - VFX / HPC sites run it on-prem across their own render farms
- **Origin:** “Adventures in NFS re-exporting” — DNEG / Daire Byrne, 2020
 - Upstream thread: Lever, Myklebust, Layton, Fields, Amazon’s van der Linden



LINUX

PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Control Plane Enhancements



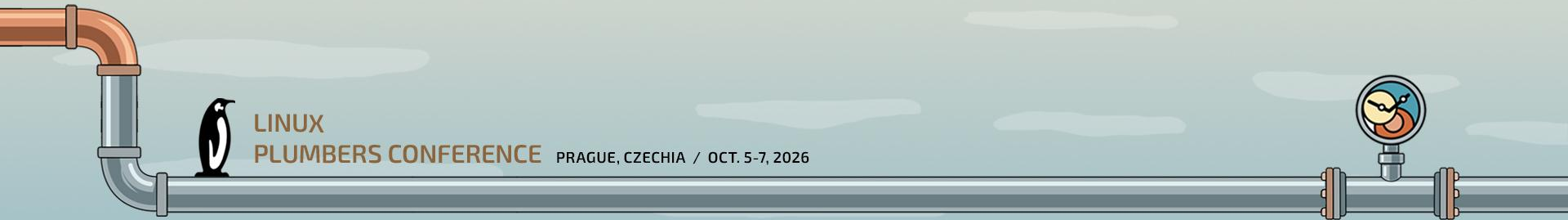
LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



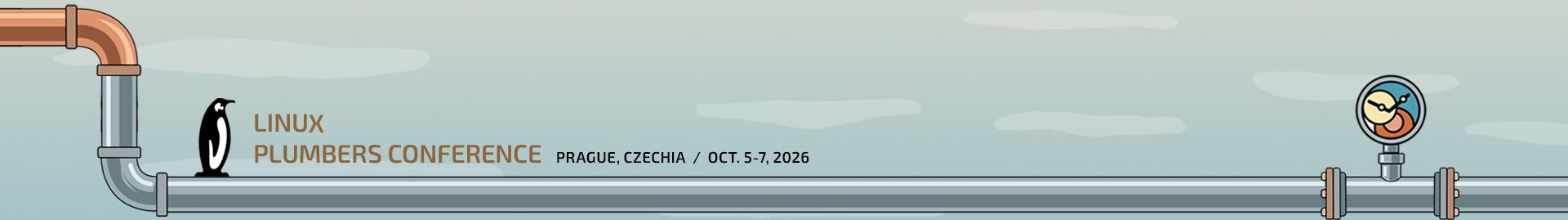
Netlink Administrative API

- **Purpose:** Extensible declarative control plane for NFSD and lockd
- **Contributor:** Jeff Layton, Meta
- **In operation:**
 - New nfsdctl command (carried in nfs-utils) side-by-side with /proc/fs/nfsd
 - New features go only in nfsdctl
 - New selftests to help catch regressions

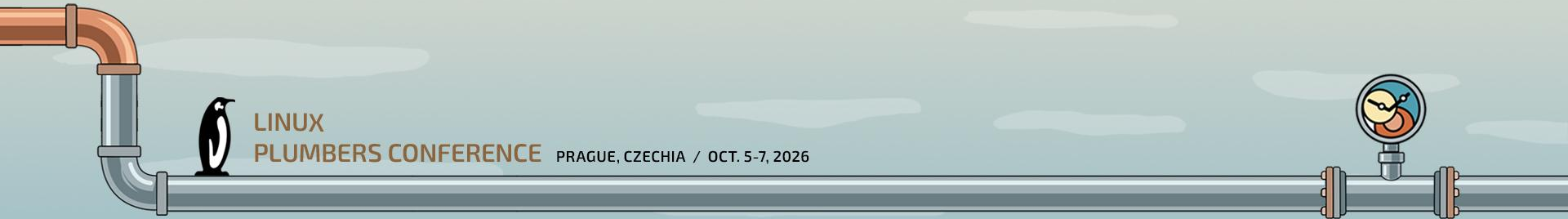


Netlink for mountd Upcalls

- **Purpose:** Extensible upcall/downcall for kernel/mountd requests
- **Contributor:** Jeff Layton, Meta
- **In operation:**
 - Similar to TLS handshake upcall: kernel calls up, mountd responds
 - Increased maximum number and size of export operations
 - More flexible options (e.g., IPv6 addresses with port numbers)



Tooling, Testing and Development Community



xdrngen: XDR Code Generation Toolchain

- **Purpose:** Reduce security vulnerabilities and coding errors in kernel XDR code
- **Primary target:** Overworked NFSD maintainers
- **Contributor:** Chuck Lever, Oracle
- **In operation:**
 - Hand-rolled XDR kernel code is to be replaced with a spec-generated, verifiable implementation in C
 - Later Rust will be used instead of C
 - To date: server side NLM, NFSv2, and some NFSv4.2 code has already been replaced



LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Upstream CI: kdevops-Based Testing

- **Purpose:** Catch regressions before they reach linux-next and torvalds.git
- **Contributors:** Jeff Layton, Meta; Chuck Lever, Oracle
- NFSD upstream CI testing based on kdevops
 - **Three branches:** nfsd-testing, nfsd-next, nfsd-fixes
 - **Five test suites:** xfs, git, ltp, nfstest, pynfs
 - **Next steps:** Light up testing for Kerberos, TLS, and pNFS block
- **Gaps:** administrative interfaces, security policy, implementation limits; NFSv3 and NFSv4.2 specific protocol features
- CI-discovered regressions caught and fixed pre-merge
 - Many behavior issues with write and attribute delegation



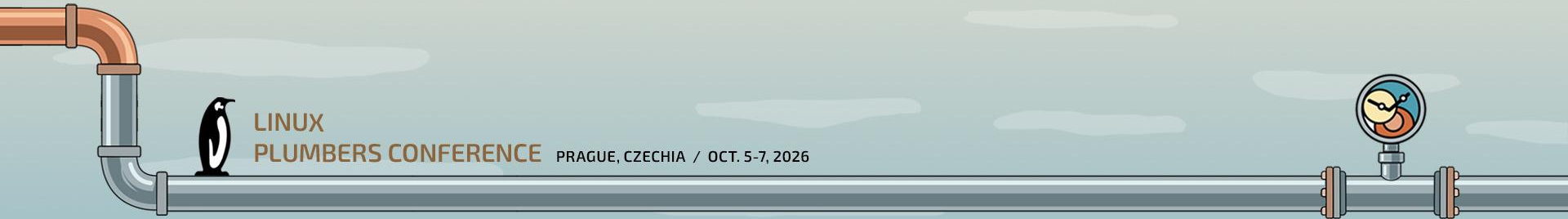
LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Dev Community: Maintainer Profile & Process Maturity

- **Purpose:** Document contribution and review expectations
- **Primary target:** NFSD maintainers and contributors
- **Contributor:** Chuck Lever, Oracle
- **See:** `Documentation/filesystems/nfs/nfsd-maintainer-entry-profile.rst`
 - Defines review expectations, CI requirements, merge criteria



LLM Assistance

Purpose: Better maintainer scalability

Primary target: NFSD maintainers and reviewers

Contributors: Jeff Layton, Meta; Chris Mason, Meta; Chuck Lever, Oracle

Inbound pressure

- Agentic vulnerability detection greatly increases the volume of contributions
 - Submitters include both bots and security researchers without NFS-specific domain knowledge
 - More vulnerability reports, false positives, and independent bug rediscovery
- Higher complexity too, not just higher volume

Our response

- Review & triage: troubleshooting, vulnerability discovery, code review
- Authoring: fix, test, and documentation generation
- Release work: release management, LTS backports



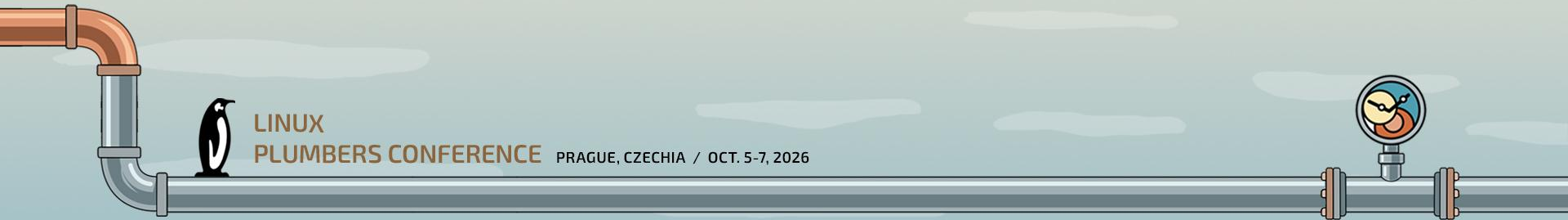
LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



What's Next

- I/O cache control: Maturing from experimental to supported configuration
- pNFS SCSI layout: continued hardening for broad deployment
- xdrgen: Coverage of NFSv3 and NFSv4
- Netlink API: Completing the control plane
- TLS Session Tagging: Certificate-based export access control policy
- Removal of NFSv2 support (once distros ship CONFIG_NFSD_V2=n)
- VFS-layer work to allow for parallel directory-morphing operations



Credits

- **Maintainers:** Jeff Layton, Chuck Lever
- **Reviewers:** Neil Brown, Olga Kornievskaia, Dai Ngo, Tom Talpey, Christoph Hellwig
- **Top contributors:** Al Viro, Mike Snitzer, Sergey Bashirov, Li Lingfeng, Benjamin Coddington, Yang Erkun

