Funded by the European Union
NextGenerationEU
GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
Plan de Recuperación, Transformación y Resiliencia
CDTI INNOVACIÓN
openchip
Enabling the Digital World

Enabling High-Performance Storage for RISC-V: Porting the Lustre Parallel File System Client

RISC-V Summit Europe 2026

Dave Cremins & Abdul Halim

09/06/2026

Agenda

- RISC-V Storage Challenge
- Lustre
- Porting Lustre Client to RISC-V
  - Patches
  - Testing
  - Upstream
  - CI

Openchip

- Officially established in Barcelona in 2021
- A European Semiconductor company with offices in Spain, Ireland, Poland, France, Germany & Italy – see funding below
- Specialize in energy-efficient SoC solutions based on the RISC-V architecture
- Target domains: HPC & AI
- Aligned with EU Chips Act
- Goal: Advance European Digital Sovereignty

About Me

- Principal Software Engineer
- 20 years of XP (high level - low level)
- Current focus: Architecture, K8s Orchestration, PCIe Card Communication, AI Inference & Confidential Compute
- Interests: Fantasy books, Games, Movies & Cycling

Grant from the Program DARE SGA 1 co-financed by CDTI and the Horizon Europe Research and Innovation Framework Programme of the European Union:

Funded by the European Union NextGenerationEU

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

Plan de Recuperación, Transformación y Resiliencia

CDTI INNOVACIÓN

The Challenge

RISC-V is Ready for HPC... But Where's the Storage?

- Accelerated adoption of RISC-V in HPC
  - European Processor Initiative (EPI)
  - EPAC accelerators
  - SiFive HPC cores
- **Critical Gap:** No RISC-V support for highly utilized HPC filesystem
- HPC clusters need high-performance shared storage
- RISC-V nodes are isolated from existing infrastructure without this support
- How do we connect into existing Lustre infrastructure?

"Lustre support identified as critical missing component for RISC-V HPC adoption"
— RISC-V HPC Gap Analysis, 2024

Lustre

- Open-source, POSIX-compatible parallel file system for HPC & AI
- Designed for massive scalability, high performance, and high-availability
- Maintained by Whamcloud community

NEC LXFS-z Storage Appliance is built on Lustre
Source: NEC LXFS-z Storage Appliance

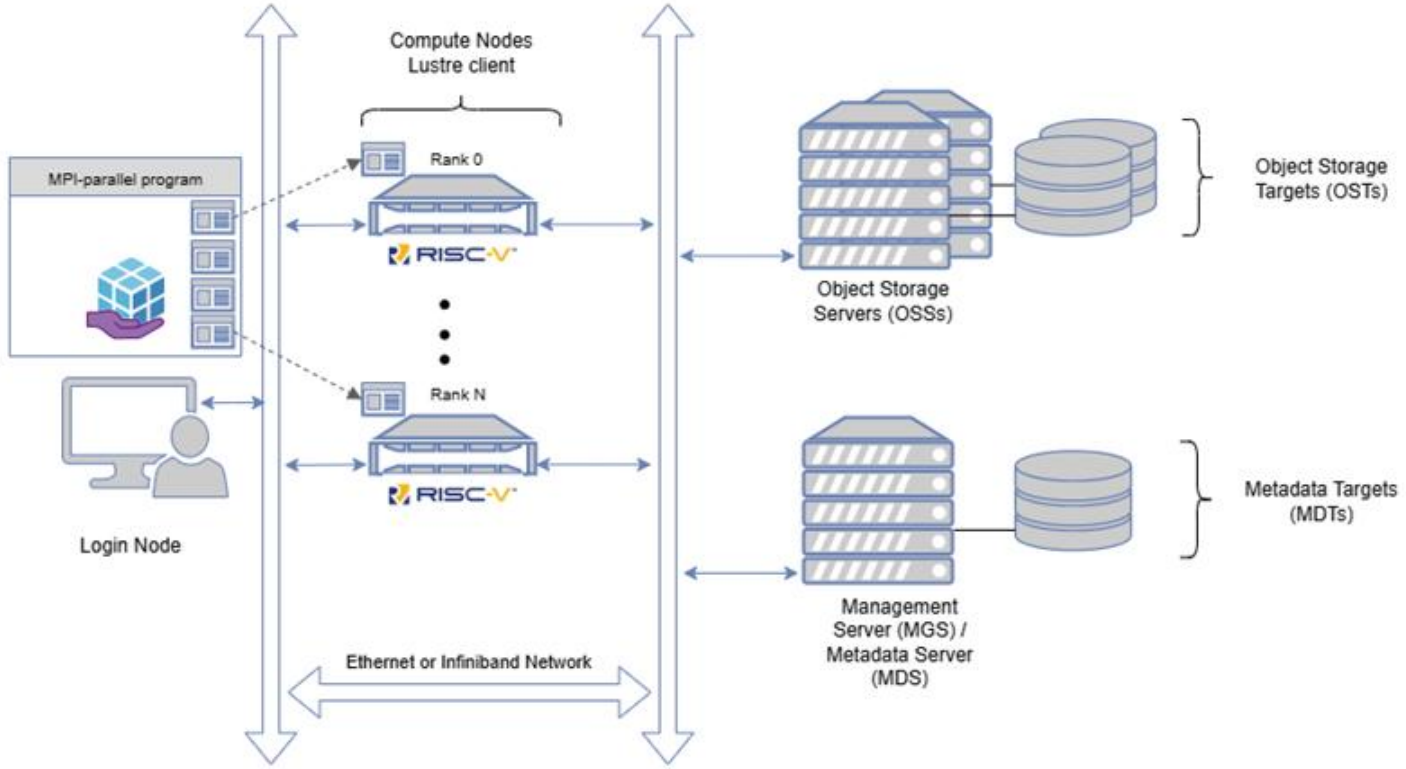
Use Cases

- **WRF** → NetCDF/HDF5 grids & checkpoints → many ranks writing shared outputs → pNetCDF on Lustre delivers 10–20× faster than serial; stripe for large files.
- **CFD** → mesh/snapshots → large sequential I/O + many files → large stripe size, distributed metadata for shared dirs.
- **AI/ML + HPC** → checkpoints & features → bursty write/read → tiered Lustre (NVMe + HDD) under one namespace

Lustre: The Backbone of HPC Storage

Lustre core architecture:

- **Clients** ⟶ access via LNet (TCP/RDMA)
- **MDS/MDT** ⟶ metadata (namespace, attributes, locks)
- **OSS/OSTs** ⟶ object storage servers/targets for file data
- **MGS** ⟶ cluster config; often collocated with MDS
- **Backends**: ZFS/ext4 on JBOD/JBOF, NVMe, RAID; HA via Pacemaker/Corosync



Lustre file system components in a basic HPC cluster

Approach for Porting Lustre Client to RISC-V

Architecture Support

Architecture	Status
X86	✅ Full Support
PowerPC	✅ Full Support
ARM	✅ Full Support
s390x	✅ Full Support
RISC-V	❌ Not Supported

1 Identify Architecture Gaps

- Build system detection
- Arch-specific code paths
- Kernel API dependencies

2 Minimal Invasive Changes

- Add RISC-V alongside existing architectures
- Follow ARM64 patterns where applicable
- Preserve backward compatibility

3 Separate Concerns

- RISC-V patches (architecture-specific)
- Kernel 6.16 patches (arch-agnostic)
- Keep patches independent & reviewable

Component	Description	Status
llite	VFS interface layer	✅
MDC	Metadata client	✅
OSC	Object storage client	✅
LOV/LMV	Logical volumes	✅
LNET	Network layer	✅
ksocklnd	TCP/IP transport	✅
libcfs	Core libraries	✅
Total	16 kernel modules	✅

Not In Scope (Future Work):

- Server components (MDS, OSS)
- ldiskfs/ZFS backends
- RDMA transports (o2iblnd)
- Kerberos/GSS authentication

Result: Clean patch series fit for upstream

The Patch Series Overview

#	Commit	Description	Files	Changes	Purpose
1	a880b1f	SUBARCH mapping	2	+25/-4	Adds riscv64 to build system architecture detection using cleaner case statements instead of sed patterns
2	7ca6e2b	RPM filtering	1	+1/-1	Enables proper RPM package dependency resolution for RISC-V kernel version suffixes
3	acb63f4	DKMS ext4 source	1	+1/-1	Replaces hardcoded x86_64 path with $(arch) variable for ext4 debuginfo source
4	af31322	Erasure coding	1	+2/-1	Enables 64-bit optimized EC path for RISC-V (little-endian, 64-bit compatible)
5	929bfd1	Crash debug tools	1	+1	Adds RISC-V64 to crash module builder script for kernel debugging support
6	7f98f76	prefetch include	1	+1	Includes linux/prefetch.h as fallback for kernels where arch-specific prefetchw() is unavailable
7	339096f	SUBARCH cleanup	1	-10	Removes duplicate SUBARCH mapping now consolidated in lustre-build.m4

Main code change 

openchip
Enabling the Digital World

RISC-V Architecture Characteristics

Key Questions from Lustre Maintainers:

Characteristic	RISC-V	x86_64	Impact
PAGE_SIZE	4 KiB	4 KiB	✅ Compatible
Endianness	Little	Little	✅ Compatible
Memory Ordering	Weak (RVWMO)	Strong (TSO)	⚠️ Like ARM64

Memory Ordering Notes:

- RISC-V uses RVWMO (Weak Memory Ordering)
- Like ARM64's relaxed model
- Lustre already handles this for ARM64
- Existing barriers/atomics work correctly
- References: arch/riscv/vm-layout.rst, RISC-V ISA Manual, Chapter 14

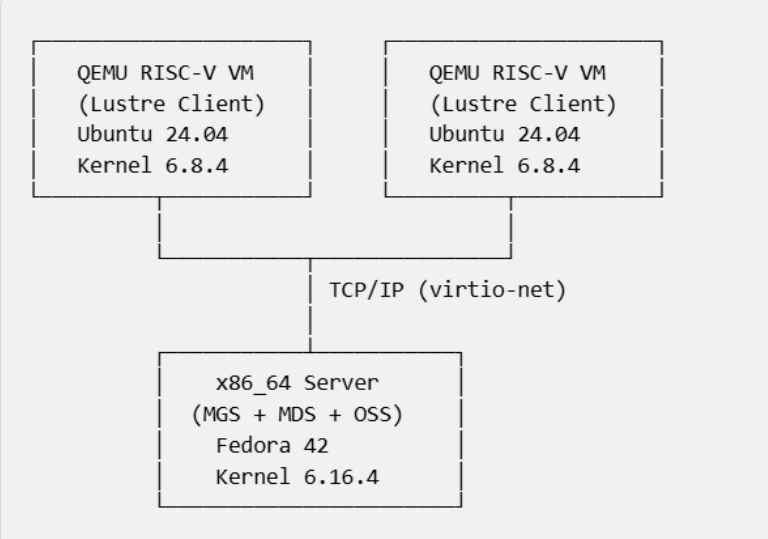
Challenges:

- Very stringent kernel version dependency & support matrix
- RISC-V specific dependencies & ecosystem (OS Distros)
- RISC-V Infrastructure (or lack thereof)

Testing

Why QEMU?

- No RISC-V hardware required
- Reproducible environment
- Easy CI/CD integration
- Sufficient for functional testing



Test Category	Test	Status
Module Loading	Load 16 kernel modules	✅ Pass
	Verify no errors in dmesg	✅ Pass
Mount Operations	Single client mount	✅ Pass
	Multi-client mount	✅ Pass
Basic I/O	File create/delete	✅ Pass
	Read/write operations	✅ Pass
	Directory operations	✅ Pass
Benchmarks	FIO sequential	✅ Pass
	FIO random	✅ Pass
	IOR parallel I/O	✅ Pass

What We Learned from Review

Review Feedback Made Our Patches Better

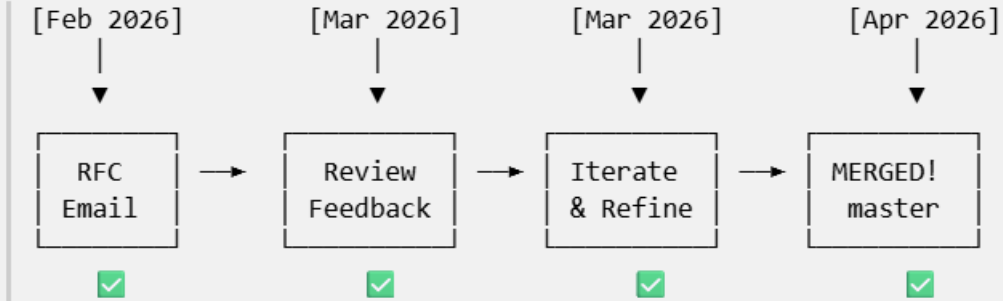
Changes from Review:

Original Proposal	After Review
8 patches	7 patches (2 not needed)
sed-based SUBARCH	case statement (cleaner)
#ifdef prefetchw guard	include header (simpler)
Hypervisor detection	Not needed for RISC-V
Strict errors disable	Not needed

Added During Review:

- SUBARCH cleanup patch (remove redundant code)
- Better commit messages
- Test-Parameters annotations

Upstream Journey



Commits in Lustre master:

- `0cffcc01b7` - SUBARCH mapping
- `98e07a36c6` - RPM filtering
- `419d9b6682` - DKMS arch-agnostic
- `f1719e943e` - Erasure coding
- `29563c1628` - Crash debug tools
- `404d45aa3e` - prefetch include
- `6dc92106e5` - SUBARCH cleanup

Available in: Lustre 2.17.51+ (master branch)

Lesson: Upstream review catches unnecessary complexity!

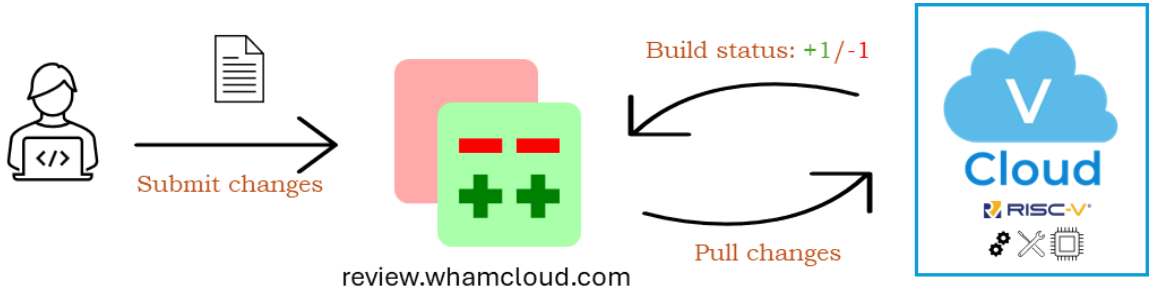
RISC-V CI/CD Infrastructure

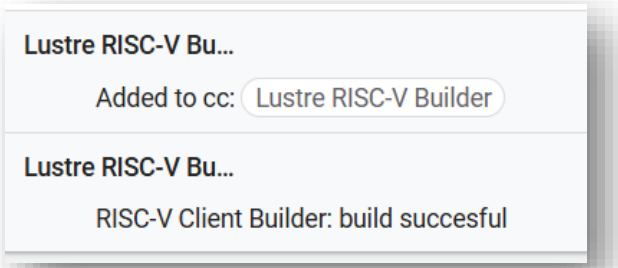
Continuous RISC-V Verification on Gerrit

The Challenge:

- Patches merged... but how to prevent future regressions?
- Need automated RISC-V build verification for all new patches
- No existing RISC-V CI infrastructure in Lustre project

The Solution:





- All open patches on Gerrit verified for RISC-V build compatibility
- Automated testing - no manual intervention needed
- Regression prevention - catch RISC-V issues before merge
- First-class architecture - RISC-V treated same as x86/ARM

Impact:
"Any patch going forward is guaranteed to build on RISC-V"

Summary

1 Lustre Now Works on RISC-V - MERGED! 🎉

- 7 patches, 22 insertions, 8 deletions
- All 16 client modules functional
- Available in Lustre 2.17.51+

2 RISC-V Ecosystem is Mature

- Complex kernel subsystem ported with minimal changes
- ARM64-like architecture characteristics

3 Upstream Process Works

- RFC → Review → Refinement → Merged in ~3 months
- Community feedback improved our patches

4 RISC-V HPC Gap Closing

- Critical storage infrastructure now available
- Every future Lustre release includes RISC-V support

"RISC-V is ready for HPC storage - and now Lustre proves it!"

Acknowledgments - RISE & Cloud-V

RISE Project (https://riseproject.dev/)
- RISC-V Software Ecosystem initiative
- Enabling open-source software on RISC-V
- Supporting critical HPC infrastructure porting

Cloud-V (https://cloud-v.co/)
- Providing RISC-V hardware resources
- Gerrit builder running on Cloud-V RISC-V board
- Enabling continuous integration for Lustre

Community Impact:
- Lustre maintainers can confidently accept RISC-V-affecting patches
- Contributors get immediate feedback on RISC-V compatibility
- RISC-V support is sustainable long-term

openchip
Enabling the Digital World

Thank You

Resources:

- Lustre Git: https://git.whamcloud.com/fs/lustre-release.git (master branch)
- JIRA Ticket: https://jira.whamcloud.com/browse/LU-19929
- Mailing List: http://lists.lustre.org/pipermail/lustre-devel-lustre.org/2026-February/011349.html
- CDAC Paper: https://dl.acm.org/doi/10.1145/3784828.3785406

Contact:

Dave Cremins:

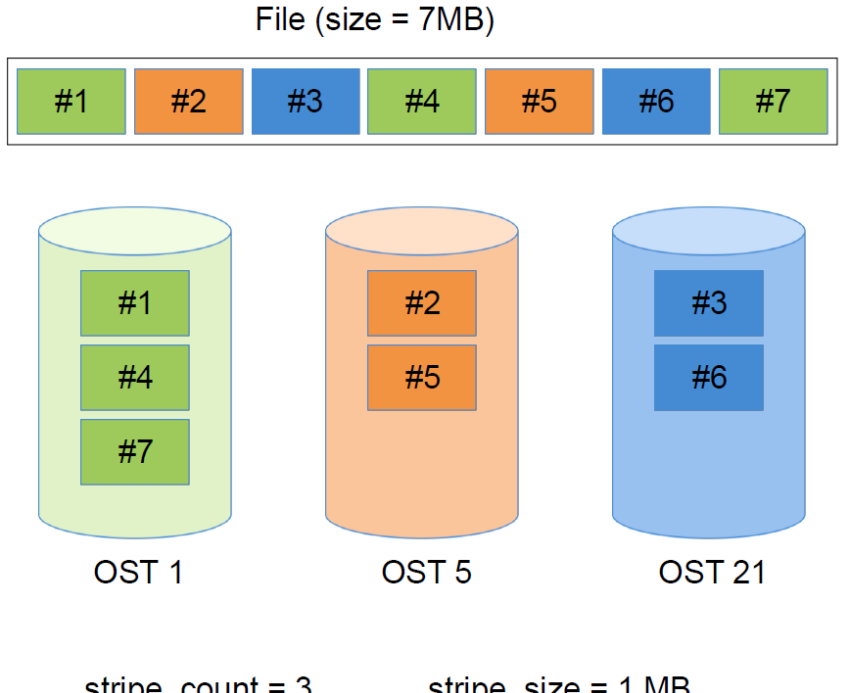
Contact:

Abdul Halim:

BACKUP

Lustre Concept

- Lustre splits a file into stripes (extents) and places them across multiple OSTs
- The two most basic properties of a Lustre file are:
  - *stripe_count* (the number of OSTs to stripe across)
  - *stripe_size* (how much data is written to an OST)
- Clients can read/write to many OSTs in parallel, increasing aggregate bandwidth and concurrency



Example Flow

- When a client opens a file, it sends a request to the MDS server
- MDS server responds to the client with information about how the file is striped (which OSTs are used, stripe size of file, etc.)
- Based on the file offset, client can calculate which OST holds the data
- Client directly contacts appropriate OST to read/write data

Development Environment

RISC-V Client Environment:

Component	Specification
Platform	QEMU virt machine
Architecture	rv64gc
OS	Ubuntu 24.04 RISC-V
Kernel	6.8.4
Lustre	2.17.0
Resources	4 vCPU, 4GB RAM

x86 Server Environment:

Component	Specification
Platform	Native / VM
OS	Fedora 42
Kernel	6.16.4
Lustre	2.17.0
Roles	MGS + MDS + OSS

Network: virtio-net, TCP/IP via socklnd

Related Work - CDAC

CDAC (Centre for Development of Advanced Computing), India:

- Independent Lustre RISC-V porting effort
- Part of indigenous HPC initiatives
- Published at HPCAsia 2026 Workshops

Significance:

- ✅ Validates feasibility across different teams
- ✅ Different approaches, same conclusion: "It works!"

Reference:

Billa et al., "Porting Lustre File System to RISC-V: Challenges and Solutions" HPCAsia 2026 Workshops, ACM
DOI: 10.1145/3784828.3785406

Future Work

Roadmap: What's Next?

Completed (2026): ✅

- ✅ Port Lustre client to RISC-V
- ✅ Upstream patches accepted
- ✅ Available in Lustre 2.17.51+

Near-term (2026-2027):

- ☐ Native RISC-V hardware testing (waiting for hardware)
- ☐ Performance optimization & benchmarking
- ☐ RDMA support (o2iblnd) when RISC-V RDMA matures
- ☐ RISC-V Vector Extension (RVV) optimized erasure coding

Long-term (2027+):

- ☐ Server-side components (MDS, OSS)
- ☐ ldiskfs backend support for RISC-V
- ☐ Integration with EPI/EPAC platforms
- ☐ Production deployments in European HPC

Community Impact: Any new Lustre release now includes RISC-V support automatically!