XI'AN · HYPERSCALE FOR RADIX

WEEK 24 · 8 JUN 2026

8–14 Jun 2026

Hyperscale Weekly

Split the work, not the world

CURRENT MILESTONE

Milestone 1 · Adaptive Sharding

Teaching the network to split and merge its shards on its own

Week 7

Previously: flightofthefox had just closed a fork-safety gap by locking shard consensus into the HotStuff-2 safe-vote rule, and was driving a full epoch transition through the simulator with the new topology schedule.

A wide-ranging week. With the basic building blocks settled, flightofthefox turned to the messy middle: what happens to a cross-shard transaction whose shards are reshaped out from under it? The answer — a fence built on settled waves, a new way for shards to join via snapshot sync, and the first splits and merges actually running end-to-end in the simulator — is the meat of this week's digest.

161
+87% vs last wk
COMMITS

+16k −3.8k
+136% vs last wk
LINES CHANGED

276
FILES TOUCHED

27
DAY COMMIT STREAK

COMMITS / DAY



peak 42/day · 7/7 days active

WHAT FLIGHTOFTHEFOX BUILT

A fence for stranded transactions

When a shard splits or merges, cross-shard transactions can be left hanging. flightofthefox built a settled-waves fence: the surviving side reconstructs what the disappeared shard had committed, and aborts what can't carry over. Nothing is silently lost.

'see "The Will" on hyperscale.rs

New shards can snap into place

A new shard no longer has to replay history from scratch. flightofthefox added a snap-sync protocol and a BoundaryStore that pins each shard's state at every epoch crossing. A joining shard downloads a recent checkpoint and catches up fast.

'see "The Archive" on hyperscale.rs

Splits and merges run for real

The simulator grew end-to-end scenarios: two sibling shards merging into their parent, a shard splitting live with observers tracking it, a validator relocating between shards. Adaptive sharding — the headline promise of M1 — is no longer just theory.

'see "The Crash Lab" on hyperscale.rs

THE BOTTOM LINE

STANDOUT CHANGE

The settled-waves fence is now in place, so cross-shard transactions survive a shard split or merge without getting lost or silently double-spent.

WHAT'S NEXT

Likely next: stress-testing the reshape loop in the simulator with rougher scenarios — committee failures, validator drops mid-reshape, rotation stalls — before turning to the lottery-style validator shuffling also promised in M1.

Written by AI from public commits, the community chat and hyperscale.rs — may contain mistakes.

Written by AI from public commits, the community chat and hyperscale.rs — may contain mistakes.

From the community

What people are saying, and the words behind the work.

THIS WEEK'S CONCEPT

The Will — *A fence for transactions stranded by a reshape.* More than a third of the week's commits are about the settled-waves fence: the machinery that decides, safely and deterministically, what happens to cross-shard transactions when one of their shards disappears at a reshape.

hyperscale.rs

JARGON, DECODED

Straddler — A transaction that touches more than one shard, so it needs both shards to commit it together.

Snap-sync — A way for a new shard to download a recent checkpoint of state instead of replaying every block from the beginning.

Settled wave — A batch of cross-shard transactions that a shard has fully committed and won't reconsider.

Written by AI from public commits, the community chat and hyperscale.rs — may contain mistakes.

ZOOM OUT

The bigger picture

Where the work landed, and the map to read along on hyperscale.rs.

MOMENTUM · LINES CHANGED / WEEK



last 8 weeks · newest right

THIS WEEK'S WORK



98% Rust 2% Config

WHERE THE WORK LANDED

simulation · The Crash Lab
split, merge, and relocate e2e tests

shard · The Will
settled-waves fence machinery

beacon · The Census
per-shard boundary data and QCs

THE FULL PICTURE

The System
The whole sharded design at a glance.

The Journey
One transaction's trip through the network.

The Clock
Consensus-attested time across independent shards.

The Census
The leaderless beacon of validators, stake and shards.

The Overlap
Why two conflicting blocks can never both commit.

The Generals
All-or-nothing cross-shard commits, computed not voted.

The Archive
Every published byte is preserved or provably expired.

The Library
All state in one merkle tree; a shard is a subtree.

The Will
How in-flight transactions settle when a shard dies.

The Lottery
Random, ever-moving committees; proven cheats are jailed.

The Triage
Under overload, urgent traffic never waits for bulk.

The Governor
Validator entry is re-priced every epoch, like a market.

The Crash Lab
The simulator that replays any failure byte-for-byte.

The Proof
Proving safety with maths, not just tests.

The Asterisks
Every design's trade-offs — including Hyperscale's own.

THE ROAD TO MAINNET

M1	M2	M3	M4	M5	M6
Adaptive Sharding	Radix Engine	Gateway & API	Validator GUI	Migration	Live support
~4 mo	~5 mo	~3 mo	~3 mo	~3 mo	12 mo

Now: Milestone 1 · Adaptive Sharding — Week 7. Since day one: 2,324 commits — one developer, fully in public.

LINKS

WEBSITE

hyperscale.rs

CODE (PUBLIC)

github.com/hyperscalers/hyperscale-rs

MILESTONE PROPOSAL (PDF)

hyperscale.rs/xian_proposal.pdf

COMMUNITY (TELEGRAM)

t.me/hyperscale_rs

AUTHOR

flightofthefox

scan for hyperscale.rs

Questions? flightofthefox is always happy to discuss in the community Telegram — t.me/hyperscale_rs.

Written by AI from public commits, the community chat and hyperscale.rs — may contain mistakes.