XI'AN · HYPERSCALE FOR RADIX

WEEK 26 · 22 JUN 2026
22–28 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 9

Previously: the simulator and production test suites had just been brought into alignment on epochs and follower pools, and the first end-to-end reshape tests were coming online.

This was a consolidation week. flightofthefox folded the dozens of test scripts scattered across the simulator and the production code into a single shared 'scenarios' framework — one set of stories that runs in both places. On top of that, the machinery that runs shard splits and merges got a new conductor, and a quiet but sweeping cleanup tidied the internal folder layout.

96	+8.0k −6.6k	218	34+
+23% vs last wk	+1% vs last wk		
COMMITS	LINES CHANGED	FILES TOUCHED	DAY COMMIT STREAK

COMMITS / DAY 

peak 26/day · 7/7 days active

WHAT FLIGHTOFTHEFOX BUILT

One test suite, two worlds

flightofthefox migrated the simulator tests, the production end-to-end tests, and the fault tests into one shared scenarios framework. The exact same reshape, straddler, and cross-shard stories now run on both harnesses — one source of truth for 'correct.'

'see "The Crash Lab" on hyperscale.rs

A new conductor for splits and merges

flightofthefox introduced a reshape orchestrator that coordinates a split or merge — who seats the new shard, who watches the old one, who keeps things in sync. Ad-hoc logic replaced with a clean duty machine shared by both runners.

'see "The Will" on hyperscale.rs

Start at one, grow on demand

The genesis startup was simplified: every network now boots from a single shard and grows to its target size under the same code path. Fewer magic knobs, and the simulator and production now share one genesis ceremony.

'see "The Census" on hyperscale.rs

THE BOTTOM LINE

STANDOUT CHANGE

The simulator and production code now run the exact same test scenarios end-to-end — the closest the project has come to one source of truth for what 'correct' means.

WHAT'S NEXT

Likely next: harden the remaining fault scenarios in the new framework, then move into the lottery-style validator shuffling that M1 still has to deliver.

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 Crash Lab — *Replay every bug byte-for-byte.* For the first time, the simulator and the live code run the exact same test scenarios — every bug found in one is automatically checked against the other, which is the foundation flightofthefox needs before a public testnet.

hyperscale.rs

JARGON, DECODED

Scenario — A test written once in a shared format that runs on both the simulator and the real production code.

Orchestrator — A coordinator that walks the network through the ordered steps of a complex operation, like splitting a shard in two.

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



WHERE THE WORK LANDED

node · The System
sweeping refactor: new orchestrator, folder tidy

simulation · The Crash Lab
tests migrated into shared scenarios

scenarios · The Crash Lab
new portable scenario crate scaffolded

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 9. 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.