

# LOCI - Execution-Aware Signal Before Tests

Catch software bugs, performance regressions, and power spikes before tests, logs, or production.

LOCI analyzes compiled executables immediately after build and delivers real, execution-aware signal - without running the software.

## What LOCI does

After your code is built, LOCI analyzes the compiled binary and predicts how it will behave on real CPUs and GPUs - before any test or workload runs.

**This is not log analysis.**

**This is not code summarization.**

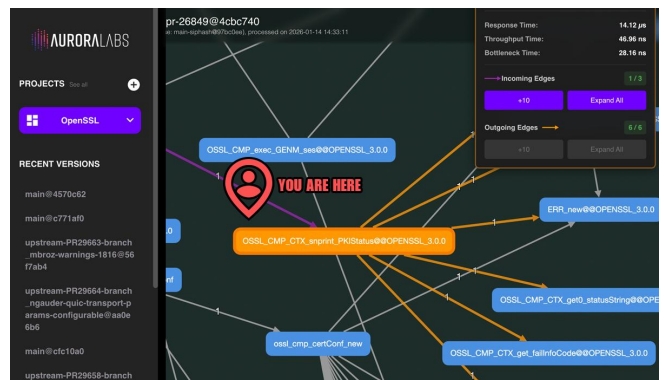
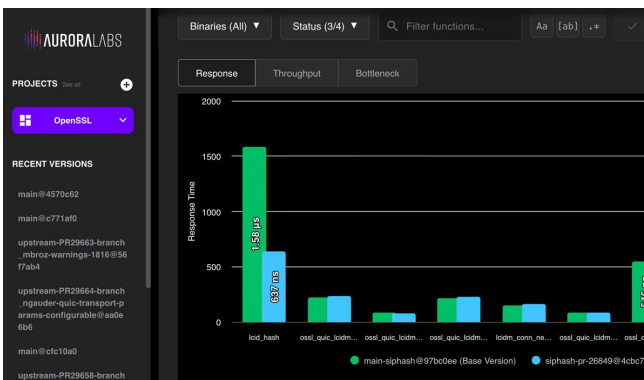
**This is software execution reasoning.**

[Book a 15 min Product Walkthrough >>](#)

## What LOCI flags early

LOCI surfaces actionable issues while they are still cheap to fix:

- Software bugs
- Performance regressions
- Power spikes & inefficiencies
- High-risk execution paths
- Performance improvements



All signals appear post build, pre-run from your IDE or CI.

## Where LOCI fits

**Code → Build → LOCI Signal → Test → Merge**

Runs automatically post-build, pre-run.

- No instrumentation
- No runtime overhead
- No new pipelines

## Why LOCI is different from other LLMs

Unlike LLMs, which generate unconstrained text and may hallucinate, LOCI predicts bounded execution-time values on real measurements, eliminating the possibility of fabricated outputs.

## Proof, not promises - See LOCI in action

Thousands of PRs, LOCI operates on real code, in the open source projects:

- [Active on GitHub](#)>
- Inspectable, PR-level results
- Applied to production-grade projects such as [OpenSSL](#)

## [Book a 15-minute product walkthrough](#)

with the LOCI team. We will review your execution path risks and show how they surface pre-run in your IDE or CI.

## [Watch the video:](#)

LOCI - Execution-Aware Signal Before Tests

