From Manual Benchmarking to Continuous Performance Validation in CI

Pioneering Daily Performance Regression Testing at Redis

Dan Eidelman – Performance Team Lead, Redis

What is the role of the performance engineer in the organisation?

Playing Defense

- Verifying performance stability
- Detecting regressions
- Validating new features

Playing Offense

- Optimizing bottlenecks
- Scaling systems
- Researching new methods
- Performing cutting edge POCs

Performance engineers balance two competing missions: pushing the limits and ensuring limits reached are kept.

Focusing on Defence

Where We Were?

Process Challenges

- Manual end-to-end benchmarks ran only before major releases.
- Complex/expansive environment constraints.
- Manual logging led to common errors and inconsistencies.

The Results

- Performance issues discovered late, often hard-to-fix architectural flaws.
- Testing focused on E2E scenarios rather than broad coverage.
- Process caused significant stress and delayed releases.

Traditional manual benchmarking processes were reactive, slow, and prone to high-stakes failure late in the cycle.

Enter Automation

Learning from QA Success

Functional QA automation solved similar challenges years ago by transitioning from manual to systematic processes.

Validation Revolution

Daily automated validation replaces manual testing with a structured approach:

- Tests as code: agile & traceable
- Enables daily quality status
- Standardized quality tools

Democratized Testing

Automation empowers the entire team - anyone can run tests and analyze results, removing bottlenecks and fostering collective ownership of performance.

Borrowing from functional testing's success to scale performance validation.

Why Automation Matters for Performance

Scalable Cloud Environments

The move to cloud environments made it feasible to scale automation for performance testing too.

- Cloud based systems can be supplemented by hardware for realism.

Continuous Integration Principles

Extending CI automation principles to performance regression testing at scale.

Elastic Infrastructure

Infrastructure can spin up multiple environments within minutes, those can be any conceivable size or configuration.

Automated Data Analysis

Performance metrics (throughput, latency, CPU/memory) are collected and analyzed automatically.

The goal: make performance validation as routine as unit testing.

High-Cadence Performance Testing

Automation enables a high-cadence feedback cycle — performance is verified continuously, not periodically.

- Multiple small, fast, and accurate performance “unit-tests.”
- Run automatically after each PR merge — detect regressions within hours, not weeks.
- Developers can trigger performance tests pre-merge for validation.
- Each test targets a focused aspect: data-path, API, or resource efficiency.
- Immediate pass/fail feedback with detailed metrics in dashboards.

Advantages

Larger Coverage, Done Faster

Comprehensive Coverage

Multiple “small tests”

Different performance dimensions:

- Duration
- Latency
- Throughput
- Resource utilization

Simplified Test Logic

setup → action → analysis

Velocity

Early Alerts on issues

Developer self-service

Challenges

The price of automation: scaling complexity.

Infrastructure

Cloud noise

Still Large environments

Long runtimes

New skill set

Data Management

Growing test volume

Result interpretation

Communication

Translating results for developers

Bottom line is needed

What do we do at Redis?

Building a continuous, automated performance validation engine

Infrastructure

Robust Automation

Prometheus Metrics

SQL Results DB

AI Skills

Visual Dashboards

Stability

Guaranteed Longevity

Reduced Cloud Noise

Deterministic Results

Test Coverage

Massive Test Volume

Performance Isolation

Continuous Execution

Reporting

Insightful Simplified Analysis

Fast Issue Detection

Developer Alerts and Next steps

`CI Pipeline → Test Orchestrator → Prometheus → SQL DB → Dashboard → Developer`

Performance Testing Infrastructure

Leveraging existing assets for continuous validation

- Reused existing QA automation infrastructure
    - Strategy: Priority of reuse over rewrite ensures long-term maintainability and full compatibility.
- Integrated with Prometheus and Grafana for real-time observability and live metrics.
- Utilized an SQL DB for long-term storage of processed statistics to enable comprehensive trend analysis.
- Deployed multiple presentation methods to maximize clarity and communication of performance data.

Stability and Longevity

Mitigating cloud noise for accurate regression detection

The Challenge

Cloud stability is a major concern:

- Tests naturally fluctuate up to 20%
- Regressions cannot be accurately detected

Noise Reduction

Use metal instances
Minimize network traffic
Disable Turbo & HT
Core affinity & isolation
Long runs for leak detection
Auto rerun on failure

Impact

Noise reduced to **3%** (often near **1%**).

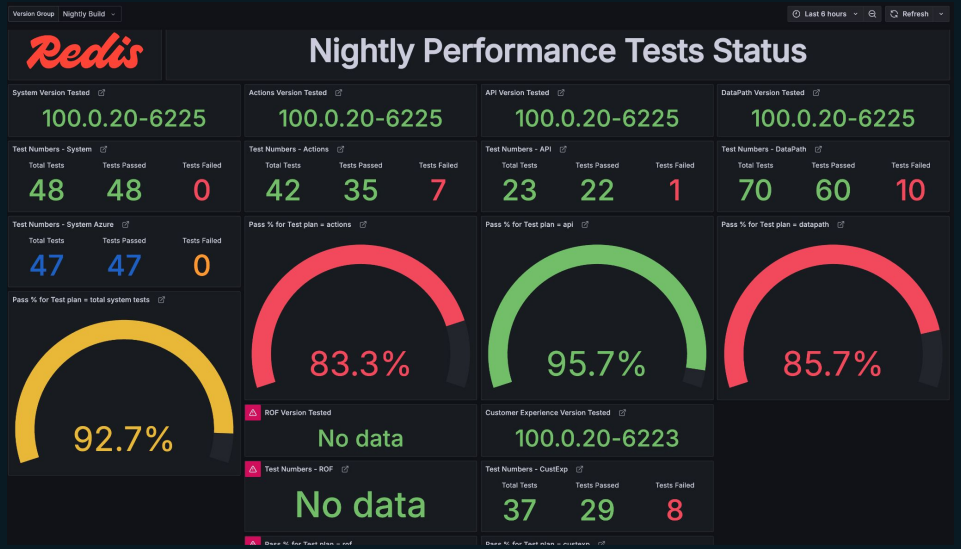
Significant improvement in regression detection thresholds.

`Cloud Noise → Hardware Optimization → Process Isolation → Deterministic Results`

Scaling Test Operations

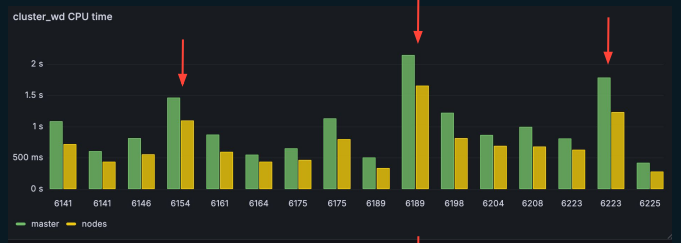
Managing growth and data complexity

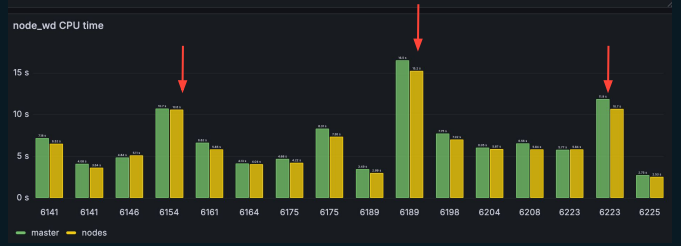
- Exponential growth in test volume
- Generation of massive, multi-dimensional metric datasets
- Active pruning of older tests based on risk/ROI assessment
- **Opportunity:** Large data volumes enable advanced AI-driven insights



AI Assisted Correlation

Process Level





Second Level

Smarter Reporting and Developer Empowerment

From detection to understanding

- **AI Code Correlation:** Links regressions directly to likely commits/PRs.
- **Enriched Performance Context**: Developers receive actionable details, not just a pass/fail.
- **Low-Impact eBPF Profiling**: Enriches test data with actionable analysis
- **Visual Easy to Read Reports**: Highlights performance changes versus baselines.
- **Goal**: Every developer ads a new AI based skill: “Performance Engineer”

Report Example

🔍 Cluster Memory Regression Analysis

Generated on 2025-11-04 19:22:37

Summary

Target Version	Baseline	Measurements Analyzed	Process Regressions	Process Improvements	New Processes
100.0.20-6146	8.0.0-19	3	42	4	18

Memory Changes by Process and Node

Top memory changes (regressions in red, improvements in green). Baseline and target totals shown in subtitle.

⚠️ service_6

Max Growth: +964.8% | **Total Diff:** +545.6 MB across 6 instance(s)

▼ Show details

Measurement	Node	Baseline (MB)	Target (MB)	Diff (MB)	Growth %
test_large_traffic	3	13.5	143.4	+130.0	+964.8%
test_large_traffic	5	13.9	141.6	+127.8	+921.2%
test_large_traffic	2	13.9	142.0	+128.1	+920.3%
test_large_traffic	4	14.3	145.0	+130.6	+911.2%
test_medium	1	16.0	32.9	+16.9	+105.3%
test_large_traffic	1	14.4	26.7	+12.3	+85.3%

Summary

- From manual, release-based benchmarks → daily CI performance validation.
- Unified infrastructure, metrics, and reporting pipeline.
- AI-assisted analysis to scale insights.

Redis transformed performance testing into a continuous engineering discipline.

Q&A

Contact:

LinkedIn

linkedin.com/in/daneidelman/

Email

Dan.Eidelman@Redis.com