

TECHNICAL PERFORMANCE REPORT

MIG Benchmark v1

Runtime Governance Performance & Reliability Study

What this benchmark asks

Can MIG evaluate and enforce AI-agent governance decisions reliably, preserve decision semantics under load, and do so with practical runtime overhead on modest infrastructure?

1,600

measured HTTP requests

0

HTTP failures

26.10 req/s

peak tested throughput

25-way

peak tested concurrency

House of Galatine Technologies Pvt Ltd

October 2026 | MIG NEXT 4.2.0 | Internal Benchmark v1

Scope note

Results reflect the documented test environment and workloads. This is an internal engineering benchmark, not an independent penetration test, certification, production SLA, or proof of unlimited scalability.

1. Executive Summary

Measured evidence from MIG NEXT under co-resident deployment conditions.

MIG is an execution-governance and runtime-security layer for AI agents. It evaluates consequential actions before execution, combines tenant policy with runtime context, and separates the security verdict from the deployment mode that is currently enforced.

~240-263	400/400	~852 MiB	1.8 GiB
ms	concurrency semantics correct	peak NEXT RSS	host RAM available post- test
median full governance paths			

Across the benchmark, MIG processed 1,600 measured HTTP requests without an HTTP failure. Governance semantic checks were explicitly verified on validation and concurrency workloads. At concurrency 25, throughput reached 26.10 requests per second while all 100 requests in that workload returned HTTP 200 and preserved the expected governance outcome.

The benchmark also verified the core Shadow-to-Enforce contract. The same no-policy action was evaluated DENY in both modes: Shadow returned an effective ALLOW while marking that it would block; Enforce returned an effective DENY and prevented execution. The test tenant was restored to Shadow after the experiment.

Primary engineering conclusion

Under the tested conditions, MIG preserved the expected governance semantics while throughput increased with concurrency. Median full-governance latency remained in the hundreds of milliseconds, memory stayed approximately flat across the concurrency range, and the service recovered to a healthy post-load state.

2. Why This Test Exists

For an AI governance layer, raw model inference speed is not the primary question. MIG sits at the boundary between an autonomous system and consequential execution. The benchmark therefore focused on whether governance remains correct, measurable, and operational as traffic increases.

- Does MIG produce the correct governance decision for policy-matched and no-policy actions?
- Can a customer observe a would-block decision in Shadow Mode without interrupting the workflow?
- Can the same action be blocked when Enforce Mode is enabled?
- Does governance remain semantically stable when requests arrive concurrently?
- How much latency appears in authentication, runtime reads, validation, and event ingestion?

- Does CPU or memory behavior become unstable as concurrency increases?
- Does the system return to a healthy readiness state after load?

AI Agent -> MIG Governance -> Policy + Runtime + Context -> ALLOW / DENY / APPROVAL -> Execution Boundary

3. Environment & Methodology

Component	Condition
MIG build	MIG NEXT 4.2.0
Compute	2 vCPU
Memory	3.8 GiB RAM
Graph database	Local Neo4j
Control plane	Remote Supabase
Co-resident service	MIG LIVE also running on the host
Benchmark client	Executed locally on the same droplet
IDE overhead	VS Code Remote/Pylance disabled during benchmark

The benchmark was intentionally not run on an oversized isolated lab server. The measurements therefore include realistic co-resident resource contention and remote control-plane latency. Sequential tests used persistent HTTP clients; concurrency tests used asynchronous clients with concurrency levels 1, 5, 10, and 25.

Workload	Requests	Purpose
Authentication	200	API-key authentication/control-plane baseline
Runtime agent read	200	Authenticated runtime inventory retrieval
Shadow no-policy validation	200	Would-block governance with non-interrupting Shadow behavior
Matched-policy ALLOW	200	Direct policy match through full validation pipeline
Event ingestion	200	Tenant-scoped telemetry/event recording

Concurrent governance	400	100 requests each at concurrency 1, 5, 10, and 25
Enforce no-policy validation	200	Blocking behavior under ENFORCE

4. Sequential Performance

Workload	N	p50	p95	p99	Max	Throughput
Authentication	200	209.29 ms	285.48 ms	501.21 ms	532.94 ms	4.42/s
Runtime agent read	200	220.97 ms	263.62 ms	510.20 ms	528.90 ms	4.28/s
Shadow no-policy validation	200	263.12 ms	388.19 ms	557.26 ms	659.36 ms	3.53/s
Matched-policy ALLOW	200	244.58 ms	329.65 ms	594.47 ms	866.93 ms	3.77/s
Event ingestion	200	226.44 ms	302.49 ms	806.60 ms	1401.26 ms	3.99/s
ENFORCE no-policy validation	200	240.48 ms	287.08 ms	497.77 ms	544.79 ms	4.00/s

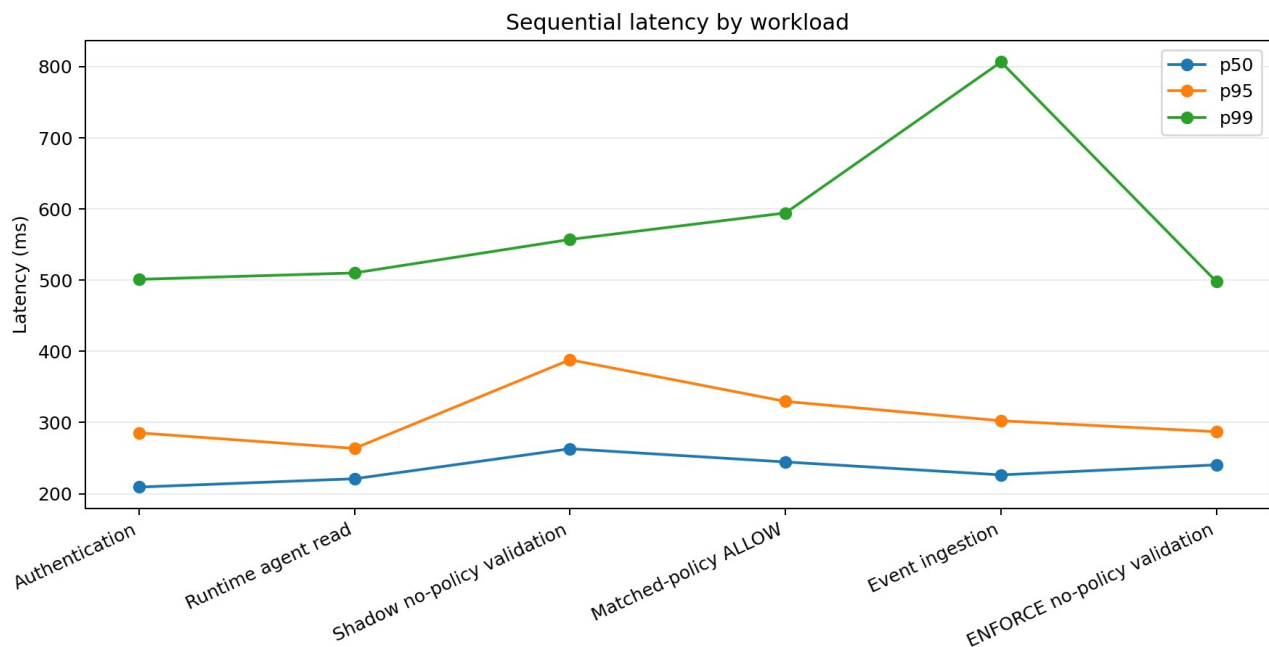


Figure 1. Sequential latency percentiles across the six measured workload types.

Interpretation

Representative full-governance median latency in this deployment was approximately 240-263 ms. Authentication alone measured a 209.29 ms median, indicating that remote authentication/control-plane access is a material part of end-to-end latency. Because these are independent runs, subtracting the values would not be a valid estimate of exact internal governance cost.

Tail latency remains visible. The largest sequential outlier was 1,401.26 ms on event ingestion, and the matched-policy ALLOW path reached an 866.93 ms maximum. These values are retained in the report rather than excluded as inconvenient outliers.

5. Shadow vs Enforce: Same Action, Different Deployment Mode

The benchmark used the same Tenant B agent, active session, read_file action, and no-policy condition in both modes. The security engine therefore evaluated DENY in both cases. Only the deployment mode changed the effective outcome.

Signal	SHADOW	ENFORCE
Evaluated decision	DENY	DENY
Effective decision	ALLOW	DENY
Would block	true	true
Can execute	true	false
Requests	200	200
HTTP failures	0	0
p50	263.12 ms	240.48 ms
p95	388.19 ms	287.08 ms
p99	557.26 ms	497.77 ms

SAME REQUEST -> MIG evaluates DENY**SHADOW**

Effective ALLOW
would_block=true
can_execute=true

Deployment mode

Same tenant
Same agent
Same session
Same action

ENFORCE

Effective DENY
would_block=true
can_execute=false

The latency difference between Shadow and Enforce should not be interpreted as proof that one mode is inherently faster. They were separate 200-request samples subject to external timing variation.

6. Concurrency & Throughput Scaling

Conc.	N	p50	p95	p99	Throughput	Peak CPU	Peak RSS
1	100	240.34 ms	359.97 ms	560.21 ms	3.80/s	34.60%	846.89 MiB
5	100	380.97 ms	874.65 ms	1094.57 ms	11.20/s	44.82%	848.09 MiB
10	100	519.05 ms	804.33 ms	849.27 ms	18.01/s	54.88%	849.60 MiB
25	100	811.62 ms	1298.39 ms	1472.67 ms	26.10/s	94.65%	852.27 MiB

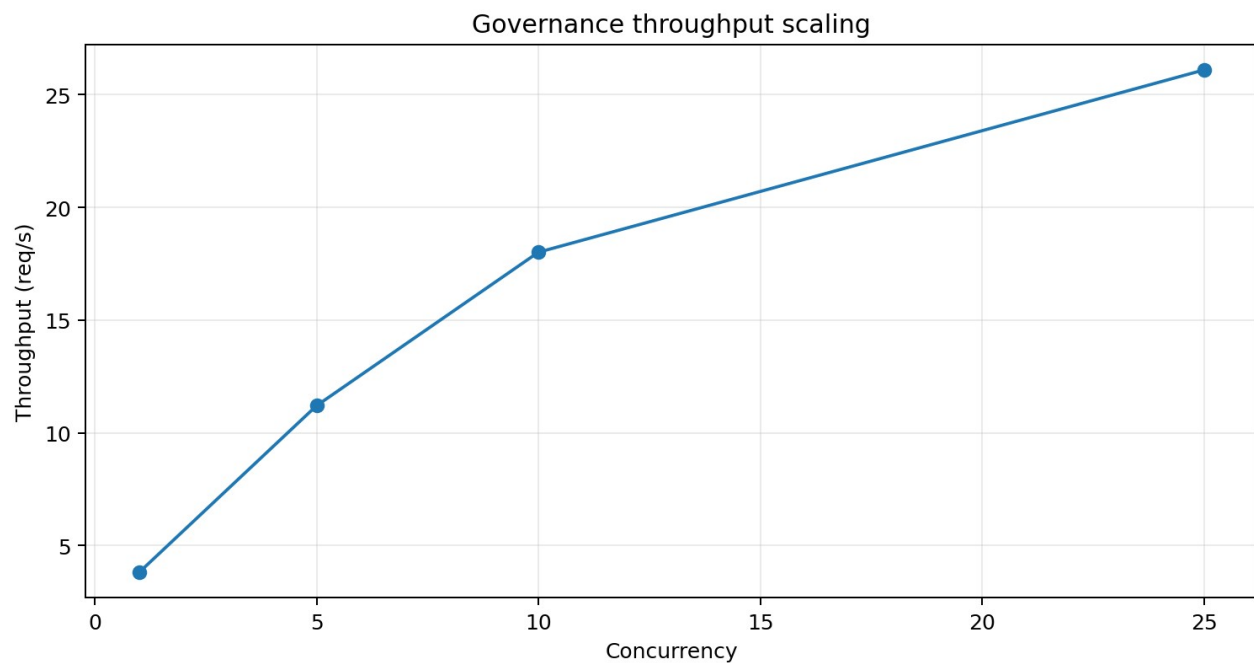


Figure 2. Throughput increased from 3.80 req/s at concurrency 1 to 26.10 req/s at concurrency 25.

Concurrency result

All 400 concurrency requests returned HTTP 200 and passed the expected governance semantic checks. Observed throughput increased by approximately 6.9x from concurrency 1 to 25. Latency also increased as demand rose, exposing the expected throughput/queueing tradeoff rather than suggesting unlimited scaling.

At the highest tested level, p50 latency was 811.62 ms, p95 was 1,298.39 ms, and p99 was 1,472.67 ms. Peak NEXT CPU reached 94.65%. The sampler treats roughly 100% as one fully occupied CPU core; the host had 2 vCPUs, so the result does not indicate complete two-core saturation.

7. Resource Behaviour & Post-Load Recovery

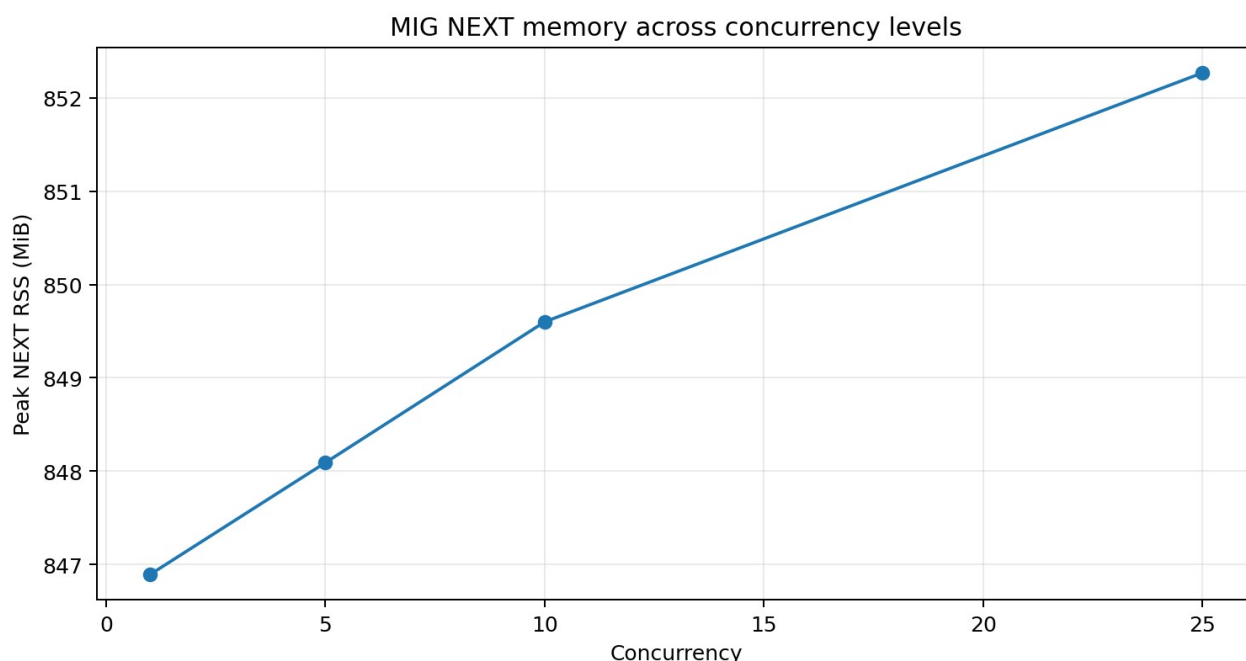


Figure 3. Peak NEXT RSS remained approximately flat across the tested concurrency range.

Peak NEXT RSS increased from approximately 846.89 MiB at concurrency 1 to 852.27 MiB at concurrency 25 - a difference of roughly 5.4 MiB. This is encouraging evidence for the tested duration, but it is not a substitute for a multi-hour or multi-day soak test.

Post-load signal	Observed state
Available host memory	~1.8 GiB
Swap used	~780 KiB
Load average	0.04 / 0.14 / 0.10
MIG NEXT readiness	healthy

Neo4j readiness	healthy
Supabase readiness	healthy
Tenant B mode after test	SHADOW (restored and verified)

Recovery result

The benchmark did not leave MIG in a degraded or incorrect governance state. Readiness was healthy after load, host load settled to low levels, and the temporary ENFORCE change was restored to SHADOW and verified.

8. Reliability & Engineering Findings

1,600/1,600

measured HTTP requests
completed

0

HTTP failures

400/400

concurrent semantics
correct

200/200

ENFORCE semantic checks
correct

Governance correctness stayed stable. The strongest result is not raw speed: validation and concurrency workloads preserved the expected decision semantics throughout the tested conditions.

Normal decision latency is currently in the hundreds of milliseconds. Full-governance p50 values in the sequential tests were approximately 240-263 ms. This can fit many consequential agent/tool actions, but not every ultra-low-latency use case.

Remote control-plane access is a material latency component. Authentication alone measured a 209.29 ms p50, so architecture-level latency work can target control-plane round trips without weakening the governance engine.

Tail latency remains an optimization target. Event ingestion and some validation runs showed visible p99/max tails. These should be profiled rather than hidden.

Throughput scales under concurrency. Observed throughput rose from 3.80 req/s at concurrency 1 to 26.10 req/s at concurrency 25 while the tested governance semantics remained correct.

Memory stayed approximately flat. Peak NEXT RSS changed by only about 5.4 MiB across the concurrency range.

9. What Benchmark v1 Demonstrates - and What It Does Not

Evidence produced	Not established by this test
Tenant-scoped authentication and runtime reads	Independent penetration-test assurance
Consistent policy-matched ALLOW behavior	SOC 2 / ISO 27001 certification
Shadow would-block semantics	A production SLA
Enforce blocking semantics	Unlimited horizontal scalability
Runtime event ingestion	Multi-region or geographically distributed behavior
Governance semantics under 25-way concurrency	Long-duration memory-leak absence
Throughput scaling in the tested environment	Performance at millions of requests
Stable memory during the tested load	External customer workload evidence
Healthy post-load readiness and mode restoration	Proof of security outside the tested scenarios

Recommended external wording: "Internal Benchmark v1 under documented test conditions." This communicates evidence without presenting the benchmark as a certification or independent audit.

10. Investor / Website Reuse

PITCH DECK METRIC BLOCK

1,600	0	26.10 req/s	~852 MiB
measured requests	HTTP failures	peak tested throughput	peak MIG memory

Deck-ready statement: MIG Benchmark v1 measured authentication, runtime reads, policy evaluation, Shadow Mode, Enforce Mode, telemetry ingestion and concurrent governance execution on a 2-vCPU, 3.8-GiB co-resident deployment. Across 1,600 measured requests, no HTTP failures were observed. At 25-way governance concurrency, throughput reached 26.10 req/s while all 100 requests in that workload preserved the expected governance semantics.

WEBSITE COPY

Governance designed for the runtime. MIG sits between autonomous agents and consequential execution. In our October 2026 internal Benchmark v1, MIG processed 1,600 measured requests without an HTTP failure across authentication, runtime telemetry, policy evaluation, Shadow Mode, Enforce Mode and concurrent governance workloads. Full governance

validation showed median latency of approximately 240-263 ms in the tested deployment. At 25 concurrent governance requests, MIG reached 26.10 req/s while the expected decision semantics were preserved for all 100 requests in that workload.

Product message

Shadow when you want visibility. Enforce when you are ready to block. MIG separates what the security engine determines from what the deployment mode currently enforces.

Suggested small print: Internal Benchmark v1 conducted by House of Galatine Technologies Pvt Ltd in October 2026 on a 2-vCPU, 3.8-GiB co-resident deployment with local Neo4j and remote Supabase. Results reflect tested workloads and infrastructure and are not a production SLA or independent security certification.

A. Appendix: Benchmark Evidence Summary

Evidence item	Result
Total measured HTTP requests	1,600
HTTP failures	0
Sequential workloads	6 x 200 requests = 1,200
Concurrency workload	4 x 100 requests = 400
Peak tested concurrency	25
Peak measured throughput	26.10 req/s
Peak NEXT CPU	94.65%
Peak NEXT RSS	852.27 MiB
Post-benchmark readiness	MIG NEXT, Neo4j and Supabase healthy
Post-benchmark tenant mode	Tenant B restored to SHADOW

Artifact reference

Phase 14 benchmark-complete snapshot: mig-next-phase14-benchmark-complete-2026-10-02.tar.gz

SHA-256: 2bc586fcfbf5b562d3b0057aba2aa9d463e15ec35f36ef7ced23b994a3665008

House of Galatine Technologies Pvt Ltd

MIG Benchmark v1 - Runtime Governance Performance & Reliability Study