AriaOS Apple Silicon Stress Validation

Empirical Behavior Under Sustained Fault Injection and Concurrency

14+ Day Max-Chaos Run

Author: Joseph C. McGinty Jr. **Organization**: ResilientMind AI LLC **UEI**: NW3SNPP7QWF4
CAGE: 14JQ9 **Version**: v1.0 **Date**: January 4, 2026 **Distribution**: Research and evaluation use

---

Abstract

This white paper documents 14+ days of sustained Max-Chaos stress testing on Apple M-Series silicon with 36GB unified memory. The validation cycle tested governance continuity, audit integrity, and recovery behavior under concurrent agent execution, fault injection, memory pressure, and latency perturbation.

Observed metrics include p99 latency stabilization at 123.8ms, CPU utilization at 32.5%, RAM utilization at 44.3%, and error rate at 0.72% under sustained 100 requests-per-second load with up to 50 concurrent agents. Event logs show transient warnings (latency elevation, memory pressure) with recovery to stable operation. No cascading failures observed.

Results were marginally different from prior G8 laptop baseline testing under identical Max-Chaos configuration.

This phase is designed to surface failure modes and recovery limits, not to guarantee system success. The validation demonstrates governance behavior under sustained load but does not constitute exhaustive safety certification.

Problem Statement

Software-only validation reaches an informational ceiling when testing distributed execution and governance systems. Simulated failure injection operates within operating system abstractions that mask physical constraints. Additional software simulation ceases to produce materially new signal about recovery behavior, governance enforcement under resource pressure, or audit integrity across actual system degradation.

Physical hardware testing is required to observe:

- Memory pressure behavior under actual unified memory architecture constraints
- Thermal effects on execution timing and resource availability over extended duration
- Storage write completion patterns during sustained logging under load
- Recovery characteristics after resource exhaustion on specific silicon architectures
- Governance decision integrity when physical compute and memory limits are approached

This validation tests AriaOS governance continuity and recovery behavior on Apple Silicon under sustained stress to identify failure modes and recovery limits that software simulation cannot reproduce.

System Context

AriaOS is an offline-first execution and governance layer designed for environments where network connectivity, power stability, and centralized coordination cannot be assumed. The system maintains decision integrity and constraint enforcement during degraded conditions through distributed supervision, persistent state management, and explicit audit trails.

Core capabilities under test:

- **Governance enforcement**: Constraint evaluation and policy compliance under resource pressure
- **Audit continuity**: Decision logging and trace integrity during sustained load
- **Recovery behavior**: Stabilization after transient resource exhaustion or latency spikes
- **Deterministic execution**: Reproducible behavior under concurrent agent execution

The system does not rely on probabilistic AI models for governance decisions. Governance is implemented through executable constraint predicates that evaluate to pass or fail. Failures halt execution rather than proceeding with degraded constraints.

This validation focuses on empirical observation of governance and recovery behavior, not performance optimization or feature development.

Test Environment

Parameter	Value
Platform	Apple M-Series SoC
Unified Memory	36GB
Operating System	macOS 26.1 (25B78)
Execution Mode	Local-only (no network dependency)
Test Duration	14+ days continuous
Scenario Profile	Max Chaos (full stack, sustained concurrency)
Configuration	Full subsystem stack enabled

Test infrastructure: Single-node deployment with local state management, audit logging, and governance constraint enforcement. No external dependencies. No cloud connectivity.

Methodology

Max-Chaos Profile

The Max-Chaos scenario applies sustained stress across multiple dimensions:

- **Concurrent agent execution**: Up to 50 agents executing simultaneously
- **Request load**: 100 requests per second sustained

- **Fault injection**: Simulated degradation including latency perturbation, resource constraint, and operational anomalies
- **Memory pressure**: Sustained allocation patterns approaching platform limits
- **Governance load**: Continuous constraint evaluation and audit logging

Observability

The validation captured:

- **Event logs** with severity classification (INFO, WARN, CRITICAL)
- **Latency percentiles** (p50, p90, p99) sampled continuously
- **Resource utilization** (CPU, RAM) tracked over full duration
- **Error rate** calculated as percentage of failed operations
- **Active system state** (concurrent requests, active agents, elapsed time)

Validation Objectives

This testing was designed to identify:

- Governance constraint enforcement behavior under sustained load
- Audit log integrity during extended operation
- Recovery patterns after transient resource pressure
- Failure modes under concurrent execution stress
- Behavioral limits of the Apple Silicon platform under this workload

---

Results

Summary Metrics

Based on observed dashboard state during 14+ day continuous operation:

Metric	Observed Value
P99 Latency	123.8 ms
CPU Usage	32.5%
RAM Usage	44.3%
Error Rate	0.72%
Active Requests	49 (at observation time)
Active Agents	31 (at observation time)
Request Rate	100 requests/second (configured)
Max Concurrent Agents	50 (configured)

Latency Behavior

Latency metrics over the 14+ day window showed:

- **P99 latency**: Stabilized at 123.8ms with visible variation in the 100-150ms range based on dashboard chart
- **P90 latency**: Remained below p99 ceiling (specific value not recorded in screenshot)
- **P50 latency**: Remained below p90 ceiling (specific value not recorded in screenshot)

- **Pattern**: Three latency percentiles track together with periodic spikes visible in chart, indicating transient load variation followed by stabilization

No sustained latency degradation observed over the 14+ day window. Spikes were transient and recovered to baseline.

Resource Utilization

- **CPU utilization**: Stable at 32.5% during observation, indicating headroom for additional load
- **RAM utilization**: Stable at 44.3% of 36GB (approximately 16GB active usage), indicating operation within platform capacity
- **Resource trend**: Dashboard charts show stable CPU and RAM utilization over extended window with no monotonic increase (no memory leak pattern observed)

Error Rate

- **Observed error rate**: 0.72% over 14+ day operation
- **Error pattern**: Event log shows transient WARN events (stress warnings, latency elevation, memory pressure) with recovery to INFO state
- **No CRITICAL failures** visible in event log sample
- **Recovery behavior**: System continued operation after WARN events without intervention

Figures

Figure 1: Test Configuration and Active State

Test Results Summary
Test Results Summary

Configuration panel showing Max-Chaos scenario with 100 req/sec, 50 max concurrent agents, and active system state showing 49 requests and 31 agents during sustained 14+ day operation. Demonstrates system operates within configured parameters under extended stress.

Figure 2: Live Metrics and Event Log (14+ Day Window)

Live Metrics Dashboard
Live Metrics Dashboard

Latency percentiles (p50/p90/p99) and resource utilization (CPU/RAM) over 14+ day continuous operation. Event log shows transient WARN events with recovery. No cascading failures. Demonstrates governance continuity and recovery behavior under sustained load.

Notable Events and Recovery Behavior

Event log analysis from visible sample:

WARN Events (transient, recovered): - `[STRESS_WARN] Response latency elevated` - Multiple instances, followed by return to baseline - `[STRESS_WARN] Memory pressure detected` - Followed by continued operation without failure

INFO Events (nominal operation): - `[STRESS_INFO] Health check passed` - Regular health verification - `[STRESS_INFO] Cache refreshed` - Periodic maintenance operations

Recovery Pattern:

The system exhibited stable recovery from transient stress conditions. WARN events did not escalate to CRITICAL failures. Latency spikes resolved without manual intervention. Memory pressure warnings occurred without process termination or constraint violation.

This indicates that governance constraints, audit logging, and state management continued functioning during resource pressure. The system did not enter degraded states requiring human intervention.

No observed cascading failures: Transient warnings remained isolated. No evidence of failure propagation across subsystems.

Comparative Note: G8 Baseline

Results under Max-Chaos on Apple M-Series were marginally different from prior G8 laptop baseline testing. Both platforms demonstrated stable operation under identical Max-Chaos configuration over extended duration.

Observed differences were within expected variance for different silicon architectures and memory subsystems. No significant behavioral divergence detected.

This suggests AriaOS governance and execution behavior is consistent across hardware platforms within the tested memory range (36GB class).

Limitations

What This Does Not Prove

Not server-class validation: Testing was conducted on laptop-class hardware. Thermal characteristics, I/O throughput, and power management differ from server or embedded platforms.

Not datacenter saturation testing: Single-node deployment does not test multi-node coordination, network partition recovery, or distributed state reconciliation under stress.

Not 128GB unified memory behavior: Testing was limited to 36GB. Governance behavior under larger state surfaces, longer retention windows, and higher concurrent agent counts remains unvalidated.

Not exhaustive safety certification: This validation identifies specific failure modes and recovery behaviors. It does not constitute comprehensive safety validation for mission-critical deployment.

Absence of observed failure is not proof of absence: 14+ days of stable operation demonstrates resilience under tested conditions. It does not prove absence of failure modes under untested conditions or longer durations.

Next Validation Step: Platform Gap

Why Higher Unified Memory Unlocks New Signal

Testing at 36GB has reached a behavioral plateau. The system operates with resource headroom (44.3% RAM utilization) under Max-Chaos load. Additional testing at this memory tier yields diminishing new signal about governance behavior under saturation.

Higher unified memory testing (128GB class) provides new observational opportunities:

Longer-horizon drift observation: Extended state retention windows allow observation of governance decision integrity over days-to-weeks of accumulated state, not hours.

Sustained saturation behavior: Higher concurrent agent counts and larger state surfaces test governance enforcement when memory utilization approaches platform limits, not operating with headroom.

Larger audit trail validation: Audit log integrity under sustained high-throughput logging over weeks tests storage and retrieval behavior at scale not achievable at 36GB.

Governance under approach-to-limits: Observing constraint evaluation and enforcement as physical limits are approached provides different signal than operation with resource headroom.

Platform Requirement

Next validation phase requires:

- Apple Silicon platform with 128GB unified memory
- Sustained Max-Chaos testing over 4 to 6 weeks
- Governance behavior observation as memory utilization approaches saturation
- Audit trail integrity validation at higher log volumes
- Recovery behavior testing after resource exhaustion at higher scale

Platform examples: Apple M4 or M5 MacBook Pro class, Mac Studio class, or equivalent Apple Silicon workstation with 128GB unified memory.

This is not a performance optimization request. This is empirical validation of governance and recovery behavior at a scale not testable on 36GB platforms.

Conclusion

Empirical findings from 14+ day Max-Chaos validation on Apple M-Series (36GB):

- Governance constraints and audit logging continued functioning under sustained concurrent load (100 req/sec, 50 agents)
- P99 latency stabilized at 123.8ms with transient spikes and recovery
- Resource utilization remained stable (32.5% CPU, 44.3% RAM) without monotonic increase over 14+ days
- Error rate of 0.72% with transient WARN events that recovered without escalation
- No cascading failures observed during extended stress period
- Recovery behavior from transient warnings occurred without manual intervention

- Results marginally different from G8 baseline, suggesting consistent behavior across platforms

Platform limitation: Additional testing at 36GB yields diminishing new signal. Governance behavior under sustained saturation, longer-horizon drift, and approach-to-memory-limits remains unobserved.

Next validation step: Higher-memory Apple Silicon validation (128GB unified memory) to observe governance enforcement, audit integrity, and recovery behavior under sustained saturation conditions not achievable at 36GB.

Appendix A: Test Parameters

Parameter	Value
Scenario Profile	Max Chaos
Request Rate	100 requests/second
Max Concurrent Agents	50
Test Duration	14+ days continuous
Target Subsystems	Full stack
Platform	Apple M-Series SoC
Unified Memory	36GB
Execution Mode	Local-only

Appendix B: Glossary

Max Chaos: Stress testing profile applying concurrent agent execution, sustained request load, fault injection, and resource pressure simultaneously.

P99 Latency: 99th percentile latency. 99% of requests complete within this time. Used to measure tail latency behavior under stress.

Agent Churn: Rate of agent lifecycle events (creation, execution, termination) under concurrent operation.

Governance Continuity: Ability of constraint enforcement and policy evaluation to continue functioning during resource pressure and degraded conditions.

Audit Trail: Immutable log of governance decisions, constraint evaluations, and operational events. Required for post-incident analysis and compliance verification.

Unified Memory: Apple Silicon architecture feature providing shared memory pool for CPU and GPU. Relevant to memory pressure testing and state management behavior.

Document Classification: Research Validation Report **Distribution**: Research and evaluation use **Revision**: 1.0 **Date**: January 4, 2026

Contact: Joseph C. McGinty Jr. Chief Innovation Officer ResilientMind AI LLC CAGE: 14JQ9 UEI: NW3SNPP7QWF4 https://ariaos.dev

@page { margin: 1in; } body { font-family: Helvetica, Arial, sans-serif; font-size: 11pt; line-height: 1.5; } h1 { font-size: 20pt; } h2 { font-size: 14pt; margin-top: 1.5em; border-bottom: 1px solid #333; } table { border-collapse: collapse; font-size: 10pt; } th, td { border: 1px solid #999; padding: 0.4em; } th { background: #f0f0f0; }