■ Ambient API Performance Report

Medium Load Test Analysis - 30 Concurrent Users

■■ PERFORMANCE STATUS: NEEDS IMPROVEMENT System shows performance issues requiring optimization before production use

■ Test Configuration Overview

Parameter	Value	Description
Target API	Ambient API	Medical conversation processing
Environment	QA	https://innovationz-qa.myqone.com
Test Type	Medium Load	Standard performance validation
Concurrent Users	30	Simultaneous API requests
Test Duration	89.3 seconds	Actual execution time
Spawn Rate	5 users/sec	User ramp-up rate
Total Requests	108	Combined requests executed
Test Date	2025-07-24	11:41:25 - 11:42:54

■ Key Findings Summary

Performance Overview: The 30-user medium load test reveals significant performance challenges that require immediate attention. While the core API functionality remains stable with 100% success rate for business logic requests, response times are critically high and system resource utilization is concerning.

Critical Issues Identified:
- Average response time: 18.9 seconds (target: <2 seconds)
- Memory utilization: 86.8% (concerning level)
- Health check failures: 100% (infrastructure issue)
- Performance degradation evident under load

Positive Aspects:
- Core API endpoint: 0% failure rate (92/92 successful)
- System stability maintained under load
- CPU utilization reasonable: 26.5%
- Consistent throughput: 1.21 req/sec

■ Executive Summary

Performance Metrics Overview

Metric	Value	Status	Target	Assessment
Total Requests	108	■ Complete	All	Test completed successfully
Success Rate	85.2%	■■ Moderate	>99%	Health check issues affect overall rate
Core API Success	100%	■ Excellent	>99%	Business logic performs reliably
Avg Response Time	18.9s	■ Critical	<2s	Requires immediate optimization
Median Response Time	23.7s	■ Critical	<1.5s	Consistently slow responses
95th Percentile	26.9s	■ Critical	<3s	Performance cliff evident
Throughput	1.21 req/s	■■ Low	>10 req/s	Limited processing capacity
CPU Usage	26.5%	■ Good	<70%	CPU resources available
Memory Usage	86.8%	■ Critical	<80%	High memory consumption

■ Detailed Performance Analysis

Response Time Distribution

Response Time Analysis:
The response time distribution reveals a concerning pattern where all response times significantly exceed industry standards. The core API responses range from 4.3 to 27.3 seconds, indicating fundamental performance bottlenecks in the medical conversation processing pipeline.

Key Observations:

- Minimum response time: 4.3 seconds (still 2x target)
- Average response time: 22.1 seconds for core API
- Maximum response time: 27.3 seconds (approaching timeout)
- Response time consistency: Poor variation (600% range)

Performance Implications:
Such response times would result in poor user experience, with users likely to abandon requests before completion. The processing appears to be CPU-intensive with potential inefficiencies in the AI model execution or data processing pipeline.

Endpoint Performance Breakdown

Endpoint	Requests	Failures	Failure Rate	Avg Response	Status
Health Check (/health)	16	16	100%	734ms	■ CRITICAL
Core API (/Ambient/...)	92	0	0%	22,079ms	■■ SLOW
Total Aggregated	108	16	14.8%	18,917ms	■ NEEDS WORK

System Resource Analysis

Resource Utilization Pattern:
The system resource analysis reveals a mixed picture. CPU utilization remains reasonable at 26.5%, suggesting the bottleneck is not CPU-bound processing power. However, memory usage at 86.8% indicates potential memory management issues or inefficient data structures.

Resource Assessment:
- CPU: 26.5% average - Room for increased load
- Memory: 86.8% average - CRITICAL level, risk of memory pressure
- Peak Memory: 88.3% - Approaching system limits
- I/O Pattern: Likely memory or network bound operations

Scaling Implications:
The high memory usage suggests that increasing load beyond 30 users would likely trigger memory pressure, potentially causing system instability or failures.

■ Error Analysis

Error Pattern Analysis:
All 16 errors (14.8% error rate) originated from health check endpoint failures, returning HTTP 404 responses. This indicates an infrastructure configuration issue rather than application logic problems.

Error Details:
- Error Type: HTTP 404 - Not Found
- Affected Endpoint: GET /health
- Error Rate: 100% for health checks
- Business Logic Impact: None (core API unaffected)
- Pattern: Consistent failures throughout test duration

Root Cause Assessment:
The health check endpoint appears to be misconfigured or unavailable. This is likely an infrastructure issue that can be resolved independently of the core API performance optimization efforts.

Impact on Overall Results:
While these errors affect the overall success rate, they do not impact the core business functionality. The 85.2% overall success rate would be 100% if health check issues were resolved.

■ Recommendations & Action Plan

■ IMMEDIATE ACTIONS (Week 1)

1. Infrastructure Fixes
- Fix health check endpoint configuration (404 errors)
- Review load balancer and routing configuration
- Ensure all monitoring endpoints are properly exposed

2. Memory Optimization
- Investigate high memory usage (86.8% critical)
- Profile application for memory leaks
- Optimize data structures and caching mechanisms
- Consider garbage collection tuning

3. Performance Baseline
- Establish continuous performance monitoring
- Set up automated alerts for response time >5s
- Create performance regression testing pipeline

■ SHORT-TERM IMPROVEMENTS (Month 1)

1. API Response Time Optimization
- Target: Reduce average response time from 19s to <5s
- Profile AI model inference performance
- Implement response caching for similar requests
- Optimize database queries and connection pooling
- Consider asynchronous processing for heavy operations

2. Scalability Improvements
- Implement horizontal scaling capabilities
- Add auto-scaling based on load metrics
- Optimize resource allocation and limits
- Test with higher user loads (50-100 users)

3. Monitoring Enhancement
- Implement real-time performance dashboards
- Add detailed application performance monitoring (APM)
- Set up alerting for performance degradation
- Create automated performance reports

■ LONG-TERM STRATEGY (Quarter 1)

1. Architecture Review
• Evaluate microservices architecture for better scalability
• Consider event-driven architecture for async processing
• Implement proper caching layers (Redis/Memcached)
• Design for cloud-native deployment patterns

2. Performance Targets
• Target response time: <2 seconds average
• Target throughput: >10 requests/second
• Target success rate: >99.5%
• Target user capacity: 500+ concurrent users

3. Continuous Improvement
• Implement performance testing in CI/CD pipeline
• Regular load testing with increasing user counts
• Performance optimization sprints
• Capacity planning and forecasting

■ Conclusion

■ FINAL PERFORMANCE ASSESSMENT

The 30-user medium load test reveals a system that demonstrates **functional reliability** but suffers from **significant performance challenges**. While the core business logic operates without failures, the response times are critically high and system resource utilization indicates scalability concerns.

■ STRENGTHS IDENTIFIED:
• Core API functionality: 100% reliability (92/92 successful)
• System stability under load maintained
• Predictable resource usage patterns
• CPU headroom available for optimization

■ CRITICAL AREAS FOR IMPROVEMENT:
• Response time optimization (19s → target <2s)
• Memory management efficiency (86.8% → target <70%)
• Infrastructure configuration (health check failures)
• Overall throughput capacity (1.2 → target >10 req/s)

■ PRODUCTION READINESS VERDICT:
The system is **NOT READY** for production deployment in its current state. However, the issues identified are addressable through focused optimization efforts. The stable core functionality provides a solid foundation for performance improvements.

■ RECOMMENDED NEXT STEPS:
1. **Week 1:** Address infrastructure issues and memory optimization
2. **Month 1:** Implement response time optimizations and scalability improvements
3. **Quarter 1:** Complete architecture review and establish long-term performance strategy

■ SUCCESS CRITERIA:
Before production deployment, the system should achieve <2s response times, >99% success rates, and support 100+ concurrent users with stable resource utilization.

Combined Performance Analysis Report - 30 Users Medium Load Test
Generated: 2025-07-24 11:51:06
Test Execution: 2025-07-24 11:41:25 - 11:42:54
Report Version: 1.0 | Framework: BDD Performance Testing
STATUS: OPTIMIZATION REQUIRED BEFORE PRODUCTION