

# ANALYSIS REPORT

## 1. Methodology

### Tools

- **k6:** load testing to simulate metric ingestion from 10,000 servers.  
ramp to 1k RPS then, ramp to 5k RPS then ramp to 10k RPS, then hold 10k RPS
- **ghz:** 10k concurrency with 30k total requests  
ramp up for concurrency 1k then 5k then 10k sustain

### Hardware Specs

- single thread node js for both approach

### Test Duration and Repetition

- both of scenarios tried **3 times for 1 min duration**

---

## 2. Quantitative Analysis

### Graphs to include:

metric	Approach1	Approach2	Winner
Avg Latency (ms)	465.38µs	1.05 s	Approach1
P95 Latency (ms)	1.06ms	1.45 s	Approach1
P90 Latency (ms)	512µs	1.51 s	Approach1
Throughput (req/s)	6498.667178/s	8246 rps	Approach2
Error Rate (%)	0.00%	0.00%	both

---

## 3. Qualitative Analysis

- **Approach 1**
  - Simple and straightforward implementation.
- **Approach 2**
  - More complex but optimized but lower network usage.
- **Scalability Limitations**
  - TCP queue may struggle to handle 10,000 simultaneous requests.
  - Streaming can become memory-intensive under extreme load.
- **Real-world Considerations**
  - Hitting 10,000 concurrent requests may overwhelm limited TCP queues.

- Network latency can impact overall performance and responsiveness
- 

## 4. Lessons Learned

System design must consider real-world constraints like TCP queue limits and network latency.

Stress testing is critical to uncover scalability bottlenecks before deployment.