Condense Load Benchmarking Report for 40K Messages / Second

Date of Benchmark: June 29–30, 2024 | Test Duration: 24 hours continuous run

Test Environment: Production-grade cluster deployed on Kubernetes (Azure AKS), fully instrumented with Strimzi Kafka Exporter and Prometheus/Grafana for telemetry

Executive Summary

This benchmarking exercise was conducted to validate the platform's capability to sustain streaming of MQTT based IOT devices, at a rate of 40k messages / second. These messages were then published to a Kafka broker through Condense's MQTT source connector over a continuous 24-hour test. The system demonstrated:

› **High Throughput Stability on MQTT Broker –** Sustained ~40K messages/sec for both ingress and egress with zero data loss

› **Massive Concurrent Connectivity on MQTT Broker –** Stable operation with ~40,000 active client connections, 8 topics, and 60 subscriptions throughout the test

› **Operational Resilience on Kafka Broker –** Successfully handled burst traffic with consumption peaks up to 150K messages/sec and managed consumer lag without service degradation

› **Healthy Infrastructure Utilization –** CPU and memory usage remained well below critical thresholds, leaving capacity headroom for further scale.

These results confirm the platform's readiness for **mission-critical, high-volume streaming deployments** where predictable performance, minimal loss, and infrastructure stability are required

Objective

The objective of this benchmarking exercise was to validate the platform's ability to handle sustained, high-volume message processing workloads under production-like conditions. The test was designed to evaluate throughput, connection stability, consumer lag behavior, and infrastructure utilization

Benchmark Summary

Over the 24-hour test window, the system processed incoming and outgoing traffic at sustained high volumes, maintained stable connectivity across tens of thousands of clients, and demonstrated consistent infrastructure performance. Observations are supported by the monitoring snapshots attached in the following sections

Test Configuration & Methodology

// Environment Setup

› **Cluster Type:** Azure AKS Kubernetes cluster, production-grade

› **Node Pool Configuration:** Multiple worker nodes as indicated in CPU/memory monitoring (balanced load across all)

› **Monitoring Tools:**

- Kafka Metrics – Strimzi Kafka Exporter with Grafana dashboards.
- Infrastructure Metrics – Prometheus and Kubernetes Grafana dashboards

›**Broker Deployment:** Kafka and MQTT deployed as StatefulSets with persistent volumes

›**Storage Backend:** Azure-managed premium SSD-backed volumes.

// Kafka Broker Settings (Typical for this Workload Class)

› **Replication Factor:** Configured for production-grade fault tolerance

› **Partitions:** Adequate partition count to ensure parallelism for ~40K msg/sec throughput

› **Acknowledgements:** Set to ensure data durability while maintaining performance.

// Test Traffic Generation

› **Producer Clients:** Simulated traffic generation tools publishing to 8 MQTT topics, sustained for 24 hours

› **Message Size:** Standard payload size representative of production events

› **Consumer Clients:** Subscriptions aligned to all active topics, consuming in realtime to validate egress symmetry

// Measurement Method

› **Throughput –** Measured at Kafka broker ingress and egress points

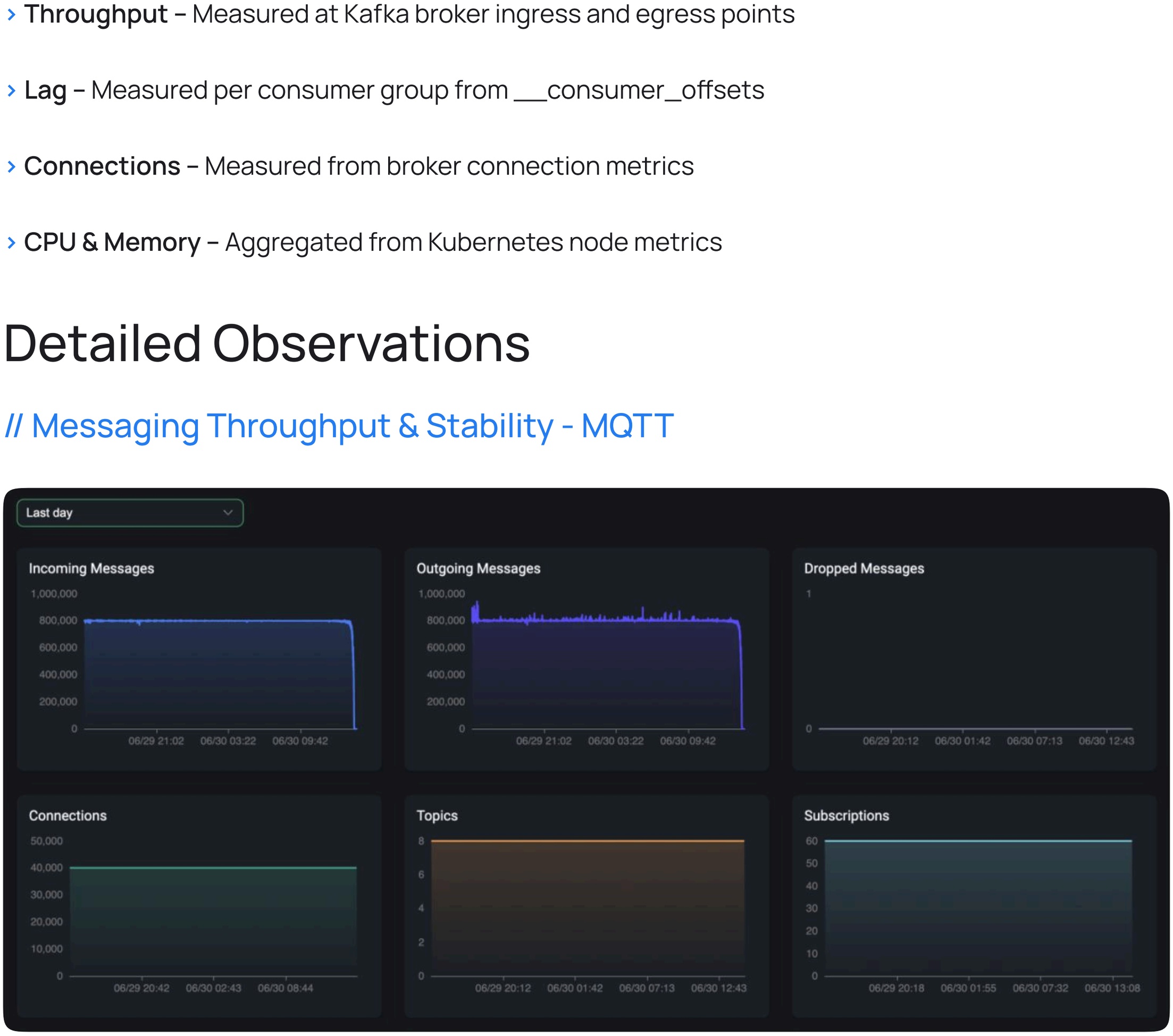
› **Lag –** Measured per consumer group from ___consumer_offsets

› **Connections –** Measured from broker connection metrics

› **CPU & Memory –** Aggregated from Kubernetes node metrics

Detailed Observations

// Messaging Throughput & Stability - MQTT



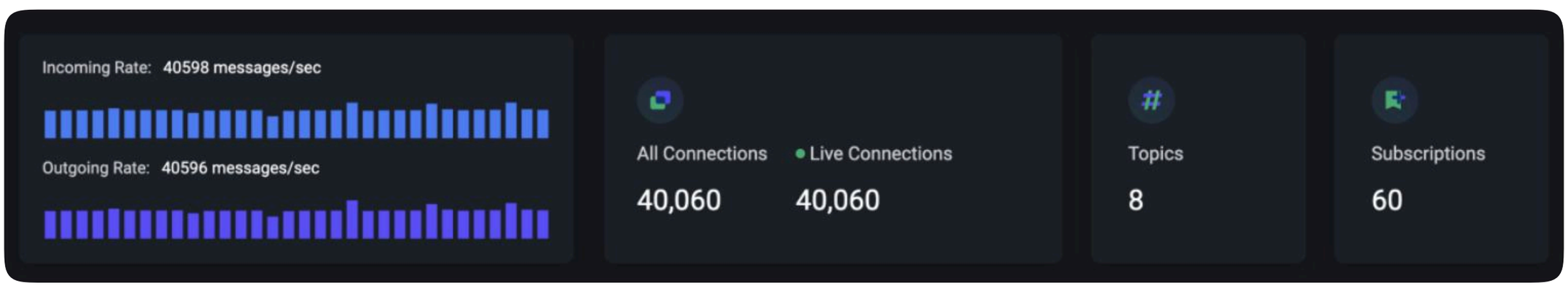
› **Sustained High Throughput –** Both incoming and outgoing message streams maintained a consistent rate of approximately 40K messages/sec (the graph above depicts the incoming and outgoing messages grouped for every 20 seconds, hence shows ~800k messages/ 20 seconds) throughout the test window

› **Zero Message Loss –** No dropped message was recorded, confirming extremely high delivery reliability

› **Stable Connection Load –** Approximately 40,000 concurrent connections were maintained without fluctuation

› **Consistent Topic & Subscription Activity –** All 8 topics and 60 subscriptions remained active and stable across the test period

> Interpretation: Confirms ability to handle heavy workloads without degradation in message delivery or client session stability



› **High Ingress Rate –** Steady 40,598 messages/sec incoming

› **Balanced Egress Rate –** Outgoing matches ingress at 40,596 messages/sec

› **Large-Scale Connectivity –** 40,060 live connections active concurrently

› **Active Topics & Subscriptions –** 8 topics and 60 subscriptions in continuous use

> Interpretation: Confirms sustained operational equilibrium with no message backlog

// Kafka-Level Metrics

› **Consistent Ingestion Rate** – Average ingestion sustained at 40K–45K messages/sec

› **High Consumption Peaks** – Spikes up to 150K messages/sec during peak demand processed without loss

› **Variable Consumer Lag** – Peaks up to ~1.9M messages during bursts, without operational impact

› **Operational Stability** – No downtime despite lag fluctuation, indicating strong recovery capability

> Interpretation: Demonstrates burst-handling capacity and stability under variable load

// Infrastructure Resource Utilization

› **Balanced CPU Utilization** – Average CPU usage below 55% capacity

› **Uniform Node Distribution** – No hotspot concentration on specific nodes

› **Stable Memory Usage** – Memory held steady at ~370 GiB, far below the max threshold

› **No Critical Spikes** – Consistent performance for the full test duration

> Interpretation: Ample resource headroom for further scaling

Benchmark Conclusion

This 24-hour benchmarking exercise confirms that Condense can reliably sustain ~40,000 messages per second with zero message loss, while maintaining ~40,000 concurrent MQTT connections and stable topic and subscription activity. End-to-end throughput remained consistent across MQTT and Kafka, demonstrating predictable performance and strong delivery guarantees under continuous, enterprise-scale production load.

The platform also proved resilient under variability, successfully handling burst consumption peaks up to 150,000 messages per second without downtime or service degradation. Infrastructure utilization stayed well within safe operating limits, with balanced CPU and memory usage and sufficient headroom to support further enterprise growth and scale.

Overall, these results confirm that Condense is enterprise-grade and production-ready for mission-critical, high-volume streaming data pipelines, where predictable throughput, operational stability, fault tolerance, and scalability are non-negotiable. This benchmark strongly validates Condense as a real-time data backbone for enterprise and industrial workloads operating at sustained scale.