

A Benchmark Suite for Streaming Data Ingestion on Slurm-Based HPC Systems

Authors:

Julian Kunkel, Sepehr Mahmoodianhamedani

Background:

Modern scientific instruments, distributed simulations, and IoT-style workloads increasingly generate continuous data streams that must be ingested and processed at high rates. In HPC environments, evaluating such ingestion systems is challenging because performance depends on the interaction between the streaming platform, distributed clients, cluster scheduling, system configuration, and monitoring infrastructure.

Aims:

This work aims to develop a reproducible benchmark suite for evaluating high-throughput streaming ingestion in HPC/cloud convergence scenarios. Rather than focusing only on one messaging system, the goal is to provide a reusable methodology for controlled experiments, campaign-based parameter studies, and comparable performance reporting.

Method:

The benchmark suite provides a local-to-HPC workflow for running ingestion experiments. It uses MPI-based producer and consumer workloads to generate ingress, egress, and simultaneous read/write pressure. Apache Kafka is currently used as the first system under test. The suite supports automated setup, topic lifecycle management, metric collection, artifact organization, and report generation. It also enables campaign-style runs that vary important parameters such as broker count, partitioning, replication, client parallelism, payload size, and workload type.

Results:

Initial validation was performed through local tests and GWDG/NHR Slurm runs, including single benchmark cases and multi-case campaigns. The benchmark collects broker-side, client-side, and system-level metrics to support reproducible performance analysis. Early multi-broker experiments demonstrate that the suite can evaluate realistic ingestion workloads and reveal configuration-dependent throughput behavior.

Conclusion:

The presented benchmark suite provides a foundation for systematic evaluation of streaming ingestion systems on HPC infrastructures. The current Kafka backend demonstrates the feasibility of the approach, while the overall design can be extended toward other publish/subscribe and streaming technologies. This work supports future studies of data-intensive scientific workflows at the intersection of HPC and cloud-native systems.