How to Monitor and Mitigate Backpressure in Stream Benchmarking

Iain Dixon, Matthew Forshaw, Joe Matthews (first.last@ncl.ac.uk)

Stream processing systems (SPS) allow for near real-time processing of stream data from external producers with uncontrollable record production. SPS have mechanisms which maintain the internal rate of a system while dealing with arrival rate fluctuations. Load shedding drops records entering the system if the offered load, or the uncontrollable arrival of records, exceeds the ingestion capacity. Backpressure slows down and halts internal data flow, preventing buffers from overflowing and records already processed upstream from being dropped. However, the presence of load shedding and backpressure in SPS benchmarks can mislead practitioners as to the system performance in deployment.

SPS benchmarks allow a practitioner to replicate streams and compare different system deployments to determine an optimal solution given responsiveness or resource constraints. Closed-loop benchmarks - where the record generator is not separated from the system under test (SUT) - are popular and easy to implement. Backpressure is an internal mechanism and, as the benchmark generator represents an external data producer, should not reach the generator. However, under closed-loop, backpressure can reach the generator in a way that is not evident to conventional metrics.

Backpressured generators can result in benchmarks overrunning the expected duration, which provides misleading latency and throughput results. Given simplistic workloads with uniform arrival rates this issue is obvious, but with more realistic random arrival rates it is impossible to determine if the SPS or the benchmark is the cause. Additionally, built-in backpressure metrics are not granular enough to provide a warning before backpressure is active. Instead, by monitoring for the cause of backpressure, we can flag benchmark tests where this occurs. When looking at benchmark generation loops, we determined that many practitioners are not efficiently utilising the generation time. Instead of spacing records out throughout a loop, benchmarks send all the desired records at the beginning of the loop and wait for the remainder which can induce backpressure.

To mitigate the effects of backpressure on the generator in a closed-loop SPS benchmark, we propose implementing a load-shedding mechanism within the generator. This replicates the behaviour of an SPS denying incoming records when at full capacity and prevents the benchmark from overrunning. In addition, we propose spacing out messages over the generation loop to prevent unintentional spikes from the generator which may induce backpressure. This approach ensures confidence in the realism of the benchmark framework as well as that the latency and throughput of the SUT are solely attributed to its performance, without being influenced by the benchmark generator.

Our contribution is three-fold: 1) to demonstrate how throughput and latency fail to capture this coordinated omission-like problem and how the causation of backpressure is better, 2) to demonstrate how load-shedding mechanisms in closed-loop benchmarks mitigates the effects of the generator on benchmark results, and 3) to demonstrate how better utilisation of generation time can prevent some instances of backpressured generators.