Detecting Trading Trends in Streaming Financial Data using Apache Flink

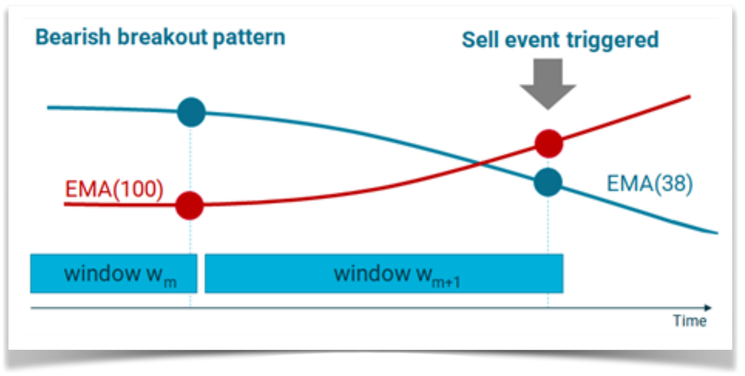
BOSTON UNIVERSITY

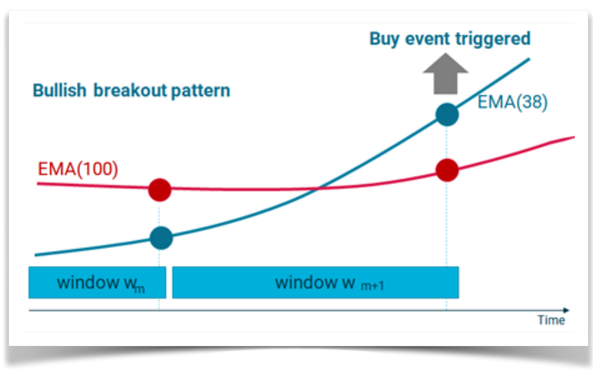
CASP Systems

DEBS 22: 16th Conference on Distributed and Event-Based Systems

Modern financial analytics rely on identifying **breakout patterns** based on the **Exponential Moving Average (EMA)** in the development of an instrument's price early on, so as to buy while the price is low and sell before a downtrend begins.

$$EMA^{j}_{w_i} = \left[Close_{w_i} \cdot \left(\frac{2}{1+j}\right)\right] + EMA^{j}_{w_{i-1}}\left[1 - \left(\frac{2}{1+j}\right)\right]$$

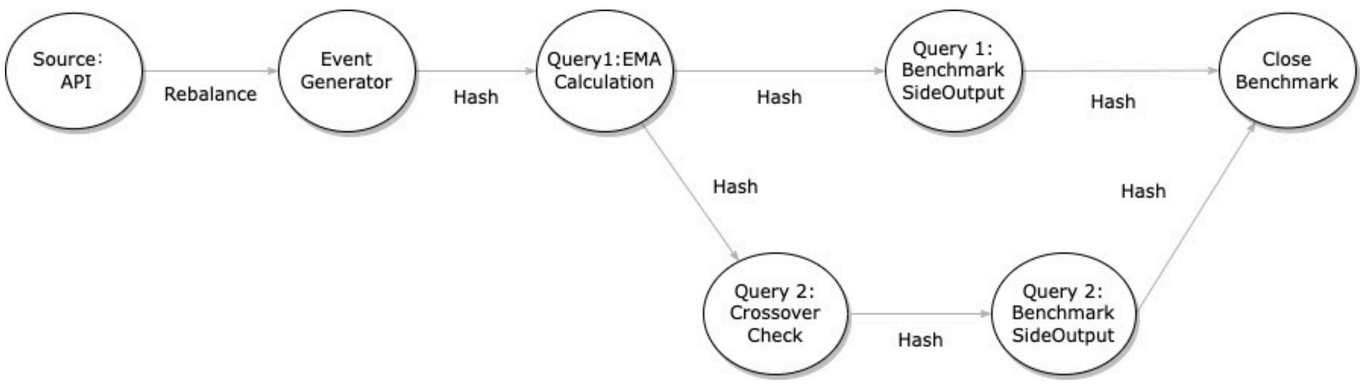


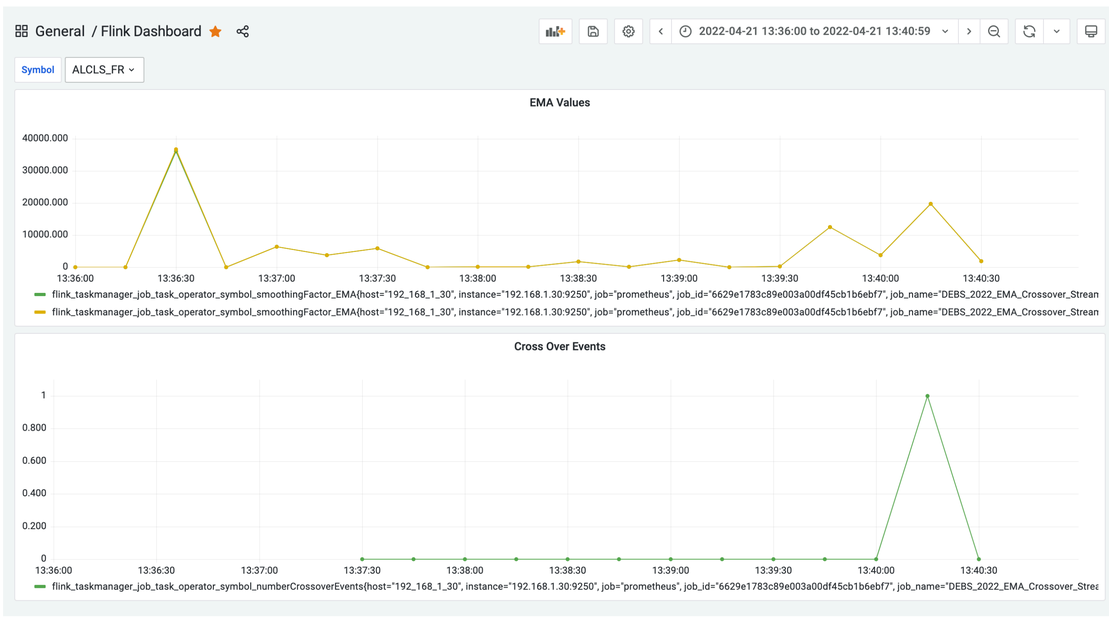


Smart Visualization of EMA values & Crossover Events with Grafana tool over Prometheus DB.

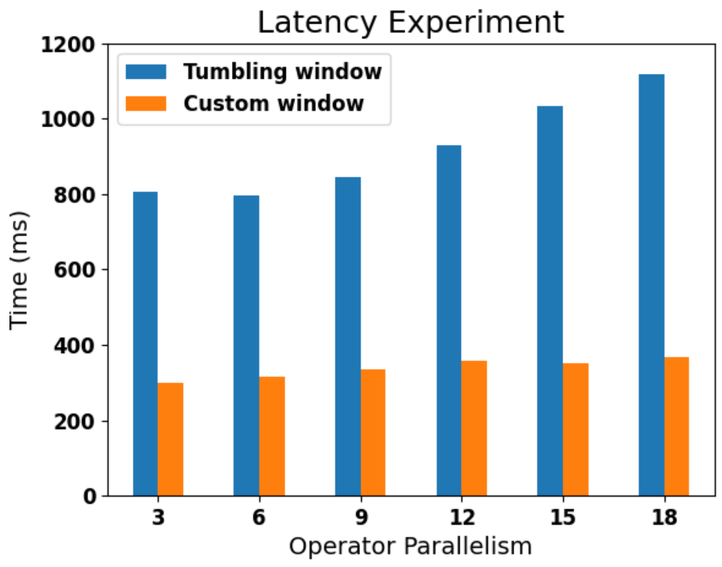
Handling high-volume streams of event notifications with the help of **Apache Flink** framework we test our design on DEBS benchmarking platform and observe a **throughput** of **45 batches/sec** & an average **latency** of **120ms.**

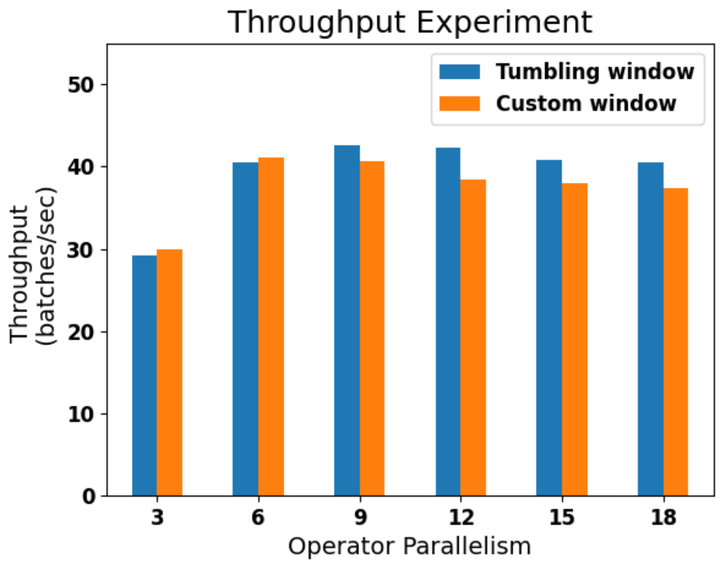
Our Design Data Flow Architecture

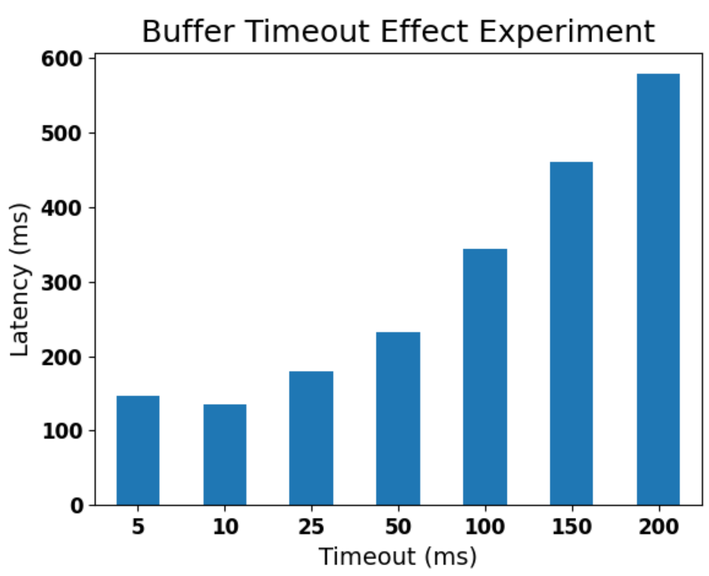




Experimental Results







Lessons Learned

1. Custom Windowing decreases latency up to **3X**.
2. Reducing size of intermediate objects increases throughput & decreases latency by **10X**.
3. Event generator parallelism boosts throughput up to **3X**.
4. Buffer timeout reduction decreases latency up to **2X**.

Emmanouil Kritharakis
ekrithar@bu.edu

Shengyao Luo
jaxluo@bu.edu

Vivek Unnikrishnan
viveku@bu.edu

Karan Vombatkere
kvombat@bu.edu

Check out our code!