Revisiting the Design of Data Stream Processing Systems on Multi-Core Processors (Paper Critique)

The rapid increasing demand in the handling of real-time data streams has led to the development of many data stream processing systems such as Apache Storm, Flink, Spark Streaming, Samza and S4. The paper sought to revisit the previous design aspects on a modern scale-up server using different applications as a micro benchmark to execute profiling studies on Apache Storm and Flink. The main goal of the study was to evaluate the common design aspects of DSP systems on scale-up architectures using profiled results so that their results and findings can be applicable to many other DSP systems, rather than to compare the absolute performance of individual systems.

The authors revisited common design aspects of modern data stream processing systems on modern multi-socket multi-core architectures, namely a) pipelined processing with message passing, b) on-demand data parallelism, and c) JVM-based implementation. After profiling studies were conducted, results showed that the designs have not fully utilized the scale-up architectures in these aspects: a) Creating a design that supports both pipelined and data parallel processing leads to a very complicated massively parallel execution model which leads to high front-end delays on a single CPU socket; b) Continuous message passing design mechanisms amidst operators acutely limits the scalability of DSP systems on multi-socket multi-core architectures.

The authors presented two optimizations which they believe will address these performance issues and demonstrate promising performance improvements.