Language-Directed Hardware Design for Network Performance Monitoring

March 30, 2017

Network performance monitoring today is restricted due to existing switch mechanisms, forcing operators to rely heavily on endpoints with poor visibility into the network core. Switch vendors have added progressively more monitoring features to switches, but the current trajectory of adding rigid features is unsustainable given the ever-changing demands of network operations.

Instead, we ask what switch hardware primitives are required to efficiently support an expressive language of network performance questions. We believe the resulting switch hardware design could address a wide variety of current and future performance monitoring needs.

We present a performance query language, Marple, modeled on familiar functional operators like map, filter, fold and zip. Marple is backed by a novel programmable key-value store primitive on switch hardware. The key-value store performs flexible aggregations at line rate (e.g., a moving average of queueing latencies per flow), and scales to millions of keys. We also present a Marple compiler that targets a P4-programmable software switch and a simulator for high-speed switch hardware. Our results show that Marple can express novel switch queries that could previously run only on end hosts, while Marple queries only occupy a modest fraction of a switch's hardware resources.