Sonata: Scalable Streaming Analytics for Network Telemetry

Abstract

Current solutions for network telemetry are ill-suited for security or performance analysis as they answer monitoring queries at insufficient levels of granularity. This limitation stems from the fact that these systems are difficult to scale as the number of queries or volume of data increases. Worse yet, queries are written in different, low-level, fixed protocols that create a disconnect between the collection and analysis phases of monitoring. We observe that modern network telemetry systems can benefit from a few key observations: (1) monitoring queries should be expressed in a single API, (2) telemetry systems should make use of *all* computational capacity available, and (2) only a small portion of the total traffic satisfies a query for most applications.

This talk presents the design, implementation, and evaluation of *Sonata*, a stream-based network telemetry system (see Figure 1) that allows network operators to express their network monitoring queries as a sequence of dataflow operations over packet tuples without worrying about *where* & *how* their queries get executed. Given a query, *Sonata*'s runtime automatically determines the optimal plan for executing queries in a scalable manner. A query plan (1) executes the input query by iteratively zooming-in from coarser to finer refinement levels, on portions of traffic that satisfy the query at each level—winnowing out the uninteresting traffic in each iteration; (2) partitions each refined queriy across the dataplane and the streaming targets which each execute a subset of the dataflow operators.

We describe the design and implementation of target-specific drivers, especially the drivers for P4-based targets which are capable of executing a subset of dataflow operations in the dataplane. More specifically, we will describe: (1) how we use P4's primitives, such as match/action tables, registers etc., to configure the dataplane for executing individual dataflow operators, (2) how we combine these operators in sequence to configure the packet processing pipeline for a query, and finally (3) how we combine these pipelines together to execute multiple dataflow queries in the dataplane.

We have released our open-source software publicly, with instructions for network operators to express and test network monitoring queries over a testbed environment [3].

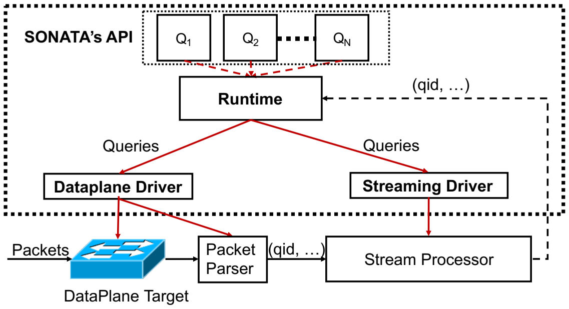

Figure 1: *Sonata's architecture consists of: (1) Application Interface for expressing monitoring queries as a sequence of dataflow operations on tuples, (2) Runtime that determines a scalable execution plan for each query, (3) Target-specific drivers that configure the targets to execute queries.*

Speaker's Bio. Arpit Gupta is currently a fourth year Ph.D. student in Department of Computer Science at Princeton University. At Princeton, he works under the supervision of Nick Feamster and mentorship of Jennifer Rexford. His research focuses on the intersection of Internet Routing, Software Defined Networks (SDN), Big Data, and Network Security. He is one of the co-creators of software-defined IXPs (SDXs) [1,2]. Before Princeton, he completed his Master's and Bachelor's degrees in Computer Science from NC State University and and the Indian Institute of Technology, Roorkee, India; respectively.

References

[1] A. Gupta, R. MacDavid, R. Birkner, M. Canini, N. Feamster, J. Rexford, and L. Vanbever. An industrial-scale software defined internet exchange point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 1–14, Santa Clara, CA, March 2016. USENIX Association.

[2] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. SDX: A Software Defined Internet Exchange. In *ACM SIGCOMM*, pages 579–580, Chicago, IL, 2014. ACM.

[3] SONATA Github. `https://github.com/Sonata-Princeton/SONATA-DEV`.