Software Defined Measurement for Data Centers

Masoud Moshref[†] **Minlan Yu** **Ramesh Govindan**
University of Southern California
moshrefj,minlanyu,ramesh@usc.edu
[†] Student author

Traffic measurement plays an important role in data centers. Many data center management tasks ranging from performance diagnosis, anomaly detection to traffic accounting and engineering all rely on accurate and on-time measurement of the traffic from different tenants and applications. For example, to perform traffic engineering, we need to correctly detect large flow aggregates and pick better routes for these flows. To reduce the latency of partition/aggregate workloads, we need to quickly identify short traffic bursts (e.g., Incast). These measurement tasks vary in three aspects: which traffic to measure, where to measure, and when to measure.

Measuring multiple granularities of flows: Operators may care about different granularities of flows (e.g., the traffic volume per tenant, per application, per VM, per IP prefix). For example, operators need to track the right granularities of flows (e.g., the source IP prefix instead of individual source IP) that contribute most to the bandwidth usage (i.e., the hierarchical heavy hitter problem). In this case, simply measuring the traffic from individual source IP would be too resource consuming than monitoring IP prefixes. But it is also challenging to pick the right prefix to monitor.

Measuring at multiple switches: In data centers, each switch sees just a part of traffic and has limited resources for measurement (e.g., CPU to collect measurement data and the memory to store the data). Therefore, it is important to coordinate the measurement across switches based on their locations in the topology, their resources, and the traffic through them.

Measuring at multiple time scales: Operators may also measure traffic at multiple time scales. For example, operators are interested in identifying both large flows in seconds or minutes to route them away from other latency-sensitive traffic, and the bursts that happen in milliseconds (e.g., Incast) to avoid significant packet loss. It is a challenging question on how to maintain traffic statistics at multiple time scales with today's commodity switches and limited resources at switches.

We propose to leverage the centralized controller in data centers to automatically distribute the measurement tasks across all the switches. We design and build a *software-defined measurement* architecture that allows the operators specify their measurement task at the controller without worrying about where or how to measure the traffic and without worrying about the resource constraints at switches.

The key challenge to design the software-defined measurement architecture is to identify the right division of labor across the controller and switches. The controller is more flexible to keep traffic information history and perform complex analysis based on the global information, while the switches can capture small time-scale events and measure all the local packets traversing them.

As the first step, we take identifying the big flows as an example and design a hierarchical heavy hitter detection system. A hierarchical heavy hitter (HHH) is the longest IP prefixes that contributes a large amount of traffic (based on a threshold) after excluding any HHH descendants in the prefix tree. A strawman approach is for the controller to dynamically allocate the switch resources to monitor prefixes with larger traffic while keeping an eye on others to monitor their changes in every time interval. However, this approach relies on the controller to make timely decisions based on the traffic it sees in the previous intervals and thus imposes a large bandwidth and processing overhead, especially for identifying small time-scale HHH. Therefore we propose to shift more responsibilities to the switches which can see the traffic in real-time, especially for longer prefixes and small time-scales.

Another challenge is that a data center topology (e.g., fat-tree) has many switches and no switch sees all the traffic. Therefore, monitoring a source IP prefix is a collaborative task of multiple switches. Given the topology and routing information, we provide algorithms for the controller to automatically allocate measurement tasks based on the available resources at switches and create an aggregated report from the results of multiple switches.

Identifying HHH in multiple time-scales is even more challenging. We explore two approaches: (1) When the controller decides to measure an IP prefix, we measure its traffic volumes across all the time-scale at the switches. (2) The controller decides both which prefix to measure and at which time-scale. The former is faster and has lower controller overhead while the latter benefits from the more flexibility in the controller algorithm gained by more information.

For future work, we will extend our algorithms for HHH to general measurement tasks in data centers by leveraging a better division of labor across switches and the controller in our software-defined measurement architecture.