Stateless Network Functions

Murad Kablan (student)†, Blake Caldwell (student)†, Richard Han†, Hani Jamjoom‡, Eric Keller†
†University of Colorado, ‡IBM Watson Research Center
(murad.kablan, blake.caldwell, eric.keller)@colorado.edu, richard.han@cs.colorado.edu, jamjoom@us.ibm.com

1 Introduction

Newly virtualized network functions (like firewalls, routers, and intrusion detections systems) continue to maintain important flow state, requiring traditional development and deployment life cycles. At the same time, many cloud-scale applications are being re-architected to cleanly separate state, enabling these applications to be more agile and support the so-called continuous deployment model.

In this poster, we propose that network functions should be redesigned to be stateless—modeling their state management after emerging cloud-scale applications. As shown in Figure 1, our architecture decouples the state from processing elements—effectively creating a tiering that is similar to web services. This comes with two key challenges: decoupling the state and achieving performance in the face of decoupled state.

For decoupling the state, we are leveraging past work which have demonstrated the ability to extract state from network functions [1,2,5]. To achieve acceptable performance, we are leveraging advances in low-latency systems such as RAMCloud [4], and at the same time exploring efforts to improve performance (e.g., caching) without compromising the consistency of the state.

2 Initial Prototype and Future Work

We have built an initial prototype based on Click [3] as the network function implementation. We have integrated an interface to RAMCloud into Click. We have also modified the mazu-nat example in the Click source tree, specifically the IPRewriter element, to store a table in RAMCloud rather than in a local storage data structure. Finally, RAMCloud cluster is connected to the network processing cluster via Infiniband.

As future work, we intend to fully implement and explore designs for further improving performance. We also intend to explore a greater collection of network functions.

References