week33

In the SDN model, the controller platform is a critical pillar of the architecture, and, as such, efforts are being devoted to turn SDN controllers into high-performance, scalable, distributed, modular, and highly-available programmer-friendly software. Distributed controller platforms, in particular, have to address a variety of challenges. Deserving special consideration are the latency between forwarding devices and controller instances, fault-tolerance, load-balancing, consistency and synchronization, among other issues. Operators should also be able to observe and understand how the combination of different functions and modules can impact their network.

As the SDN community learns from the development and operational experiences with OpenFlow controllers, further advancements are expected in terms of raw performance of controller implementations, including the exploitation of hierarchical designs and optimized buffer sizing One approach to increase the performance of controllers is the IRIS IO engine, enabling significant increases in the flow-setup rate of SDN controllers. Another way of reducing the control plane overhead is by keeping a compressed copy of the flow tables in the controller’s memory .

Modularity & Flexibility

A series of ongoing research efforts target the modular and flexible composition of controllers. RAON proposes a recursive abstraction of OpenFlow controllers where each controller sees the controllers below as OpenFlow switches. Open research issues include the definition of suitable interfaces between the different layers in such a hierarchy of controllers.

Other open issues to be further investigated in this context are the East/westbound APIs, and their use in enabling suitable hierarchical designs to achieve scalability, modularity, and security. For instance, each level of a hierarchy of controllers can offer different abstractions and scopes for either intra- and inter-data center routing, thus increasing scalability and modularity. Similarly, from a security perspective, each hierarchical level may be part of a different trust domain. Therefore, east/westbound interfaces between the different layers of controllers should be capable of enforcing both intra- and inter-domain security policies.

Another important observation is that, currently, the lack of modularity in most SDN controllers forces developers to re- implement basic network services from scratch in each new application. As in software engineering in general, lack of modularity results in controller implementations that are hard to build, maintain, and extend – and ultimately become resistant to further innovations, resembling traditional “hardware- defined” networks. As surveyed in Section IV-G, SDN programming abstractions introduce modularity in SDN applications and simplify their development altogether. Further research efforts try to achieve modularity in SDN control programs. Other contributions towards achieving modular controllers can be expected from other areas of computer science and best practices of modern cloud-scale software applications.

Interoperability and application portability

Similarly to forwarding device vendor agnosticism that stems from standard southbound interfaces, it is important to foster interoperability between controllers. Early initiatives towards more interoperable control platforms include portable programming languages such as Pyretic and east/westbound interfaces among controllers. However, these efforts are yet far from fully realizing controller interoperability and application portability.

In contrast to Pyretic, PANE, Maple, and Corybantic, which are restricted to traffic engineering applications and/or impose network state conflict resolution at the application level (making application design and testing more complicated), Statesman proposes a framework to enable a variety of loosely-coupled network applications to co-exist on the same control plane without compromising network safety and performance. This framework makes application development simpler by automatically and transparently resolving conflicts. In other words, Statesman allows a safe composition of uncoordinated or conflicting application’s actions.

Another recent approach to simplify network management is the idea of compositional SDN hypervisors. Its main feature is allowing applications written in different languages, or on different platforms, to work together in processing the same traffic. The key integration component is a set of simple prioritized lists of OpenFlow rules, which can be generated by different programming languages or applications.