API Management for the Software-Defined Network

apigee

Contents

Executive Summary

Software-defined networking (SDN) is changing the economics of large-scale datacenters. Similar to the effect of Linux on the operating system market, OpenFlow is driving disruptive commoditization of expensive network switches, routers, and controllers. As networking shifts to software control, the entire stack of compute, storage and networking now becomes a software-defined datacenter.

Where is the software that defines the network? It resides anywhere it needs to, and it uses APIs to read the state of the network and to change the network's behavior in tandem with changes in application workloads. These APIs are presented by SDN controllers, network management software, and network analytics engines. Apigee provides embedded components and management services that make it easy to program the network through APIs.

SDN Overview

Software-Defined Networking (SDN) is a shift in network-based computing and communications based on breaking existing physical boundaries on switches, routers, and controllers through well-defined APIs. This makes it possible to dynamically define all aspects of a network through software. The larger the network (often: the larger the datacenter), the larger the gains in network performance, network resource efficiency, human resource efficiency, and capital expense efficiency.

SDN has been evolving for many years, starting with "Active Networking" concepts from Nortel in 1998 and "Application Networking" concepts from Cisco in 2004. It has been brought to the tipping point through the rapid emergence of OpenFlow, an open source project begun at Stanford University, expanded at UC Berkeley and put into large-scale production at Google.

OpenFlow is both software and a protocol that specifies an API for programming the flow tables[1] of a network switch. Previously, these flow tables were not programmable remotely or by third parties, as network switches included a proprietary operating system and native programs which controlled the flow tables. With OpenFlow, the switch only manages flow tables, and the OS and programs execute on a different machine. This removes constraints on control software, as it can now be written in any language and run on any operating system and any hardware – including commodity chipsets and virtualized hardware.

As a result, the rules of the networking industry are being rewritten. Switches are getting dumber, cheaper, and more capable. Similar to the effect that Linux on x86 had on Unix and proprietary chipsets (driving them out of business through cost commoditizatio[n2]), OpenFlow enables the intelligence required to manage LANs and WANs to run in software while pushing the physical execution to the switch. A 48-port 10 GbE switch costs roughly $5K; a 48-port 10 GbE router costs roughly $20K, and the only difference is the control software. That $20K router can now be replaced by a $5K switch and a free, open source controller (Open vSwitch) running on Linux on a $2K x86 machine.

While value is being destroyed in the physical controller layer, it is being created in the software control and network management layer. New capabilities to make networks perform faster, route data based on business needs, and enable Internet-scale application communication are emerging as a result of breaking the hardware layering of modern networks. These new capabilities are based on the

[1] Flow tables are simple 5-column tables contained in memory by the switch that imply packet switching rules based on source IP, destination IP, MAC address, source port and destination port

[2] Some proprietary Unix and proprietary chipsets remain in the market in legacy systems and in edge cases where they retain a value/cost advantage, but overall Unix market share loss to Linux between 1995 and 2010 was over 90%.

ability to broadly capture and analyze network data, to feed the analysis back into network control and management programs, and to enable new apps to participate in defining the network's behavior.

The business impact to large-scale datacenters is massive. The cost of network reconfiguration is reduced by orders of magnitude in time and money, both in equipment and people. Investments and demonstrations by Google in their custom 10 GbE inter-datacenter SDN showcase these principles on the largest scale currently known on earth.

Core API Concepts in The SDN Revolution

The software that defines the network does so via APIs. The API is the control point for each component of the network: OpenFlow switches, SDN controllers, network management systems, and network analytics. API-based software can be written in a range of languages due to the ubiquity of HTTP, and it can be iterated rapidly, independent of the production and deployment cycles of network hardware.

While virtualization is a systems model, APIs are an abstract model. Virtualization enables existing physical systems descriptions to be reused in a logical environment. APIs enable complete resource abstraction. For applications which are bound to physical or logical systems, virtualization is required in order to scale. For applications written to an API, resource binding can be completely dynamic, resolving to virtualized resources or to abstracted resources at the moment of the API request. This late binding can happen at the networking layers, rather than in application code (as it was envisioned and practiced in earlier phases of distributed computing): within software-defined networks, the networking layer can reduce the loss of performance inherent in late binding because the binding is realized by pushing changes to network topology (packet routing). The more broadly the network is programmable – in terms of the physical span of switches and routers that can be commanded by software – the more broadly the late-bound targets can be distributed.

An API is a programming model. The ease of learning of a programming model is directly responsible for adoption, as is the value of the underlying service. The ease of learning has two effects on adoption: first, it sets the absolute ceiling for the number of developers who can adopt it, and second, it is related to speed of adoption and resulting network effects (critical mass, community support, momentum). The value of the underlying service, if it is well understood, exerts a strong influence on adoption independent of the ease of learning. The SDN revolution is an opportunity to engage a large number of developers in adopting high value services. The burden is on us as an industry to offer a sensible programming model and drive broad understanding of the value of the services. A new breed of developer-administrators has already emerged in the DevOps movement, and they are ready to expand their capabilities to the software-defined datacenter.

Apigee's Application Networking Heritage

Apigee was founded in 2005 with a vision of full-stack application networking at Internet scale. The company was founded by Internet pioneers who developed protocols like BGP and technologies like Optical Switching and DSL Routing. They believed that scaling application communication would require network intelligence at the app layer to control network hardware at the physical and transport layers.

Apigee built a hardware-accelerated system which controlled layers 2 through 7 of the networking stack, using algorithms which provided deep inspection into app traffic from network packets to XML and JSON payloads. Performance improvements in commodity hardware and the shift to cloud computing led Apigee to transform the system into software.

The "packets to payloads" approach paid off when the growth of cloud service providers and mobile devices created large amounts of XML and JSON network traffic, now called APIs. Enterprises found that their existing Web infrastructure was blind to the API traffic that their mobile apps and partnerships depended on. Apigee Enterprise, Apigee's API management platform, solved this problem, and evolved over time to include analytics, developer channel, and application services, delivered on premises or from the cloud.

The vision of completely application-aware networking was not fully realized, however. Enterprises already relied on large equipment providers to manage lower layers of their networks. These networks were proprietary and well-protected from innovation. Software-defined networking enables collaboration between application-aware network management and foundational networking elements and controllers. In this environment, Apigee enables dynamic datacenters that adapt to shifts in demand for specific applications, optimizing network resources, routes, and API services.

Apigee has the combination of networking, application, analytics, and enterprise capabilities to elevate a software-defined network into an application-optimized datacenter. It is a necessary complement to the array of SDN technology now available in the market.

Apigee's API Management for SDN

Apigee's API management for SDN includes two primary components: embedded API management software and stand-alone API platform for multi-vendor SDN network integration and analytics.

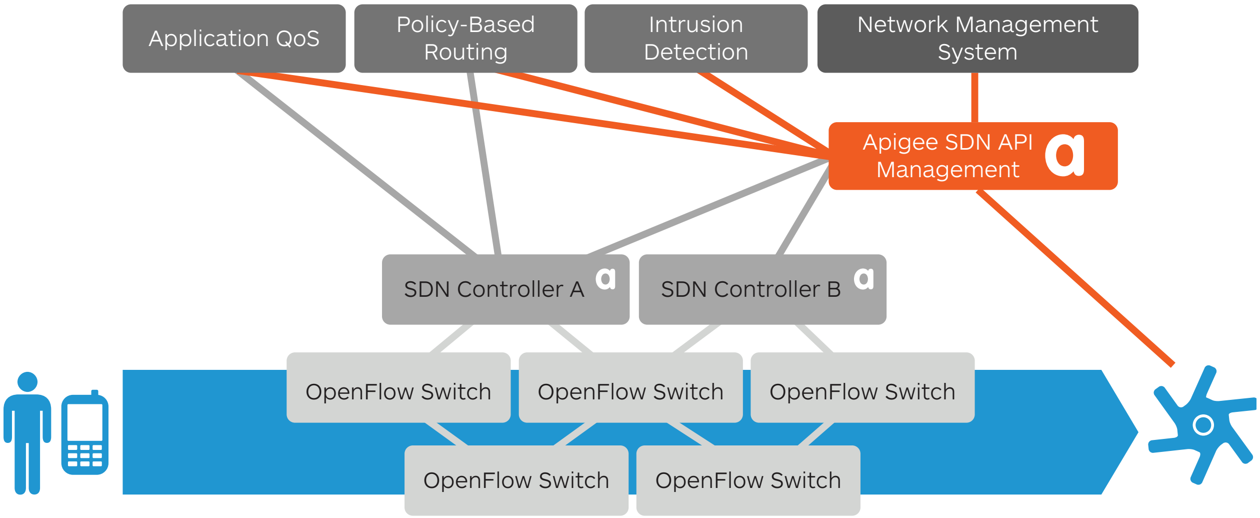

Figure 1 Connecting the SDN ecosystem and optimizing end-user experience.

SDN controller software includes the "Northbound" APIs that enable applications to interact with the controller. Apigee's embeddable API management software makes SDN controller software easier to program and easier to manage, while gaining control, visibility, and analytics of apps that use their Northbound APIs. These applications will bring to reality much of the innovation and new capabilities that will define the programmable datacenter.

Apigee's embedded API management software for SDN controllers features:

- Caching
- Security
- Prepackaged analytics

Apigee's API platform for multi-vendor SDN is stand-alone software that can integrate network management systems with SDN controllers from multiple vendors through real-time API transformations. This software includes network analytics, enabling dynamic policies on the controller itself, as well as network-based programs that can use trends to trigger a change in network behavior. Apigee's software reads network traffic into a domain model and publishes network traffic analytics as an API.

Apigee's API platform for multi-vendor SDN features:

- API transformation
- Prebuilt transformations for all known SDN controllers
- Continuous/scheduled updates & extensions of pre-built transformations
- Custom transformations
- Controller enrollment
- Discovery of controllers
- Network traffic analytics available via API
- Definition of custom events & entities via API
- Bindings to Ruby, Python, C, Java, and Javascript

For further inquiry

If you'd like more information on Apigee's API management for SDN, please email **info@apigee.com** or send a message via Twitter to **@apigee**.

About Apigee

Apigee is the leading provider of API technology and services for enterprises and developers. Hundreds of companies including AT&T, Bechtel, eBay, Korea Telecom, Telefonica and Walgreens, as well as tens of thousands of developers use Apigee to simplify the delivery, management and analysis of APIs and apps. Apigee's global headquarters are in Palo Alto, California, and it also has offices in Bangalore, India; London; and Austin, Texas. To learn more, go to **apigee.com**.

Find Best Practices to Accelerate your API Strategy

Scale, Control and Secure your Enterprise

Build Cutting-Edge Apps and Intuitive APIs