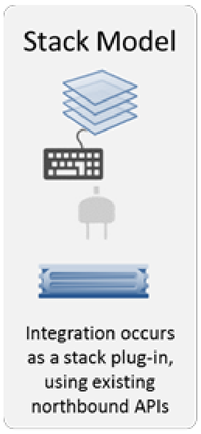
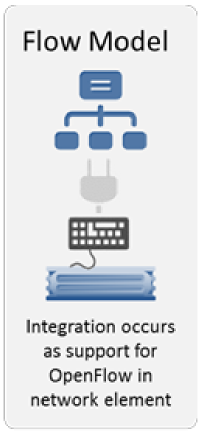
It's On: Stacks versus Flows

Lori MacVittie, 2012-05-11

#OpenStack #CloudStack #OpenFlow #SDN It's a showdown of model versus control – or is it?


Stack Model
Integration occurs as a stack plug-in, using existing northbound APIs


Flow Model
Integration occurs as support for OpenFlow in network element

There's a lot of noise about "wars" in the networking world these days. OpenStack versus CloudStack versus OpenFlow-based SDN.

But while there are definitely aspects of "stacks" that share similarities with "flows", they are not the same model and ultimately they aren't even necessarily attempting to solve the same problems.

Understanding the two models and what they're intended to do can go a long way toward resolving any perceived conflicts.

The Stack Model

Stack models, such as CloudStack and OpenStack, are more accurately placed in the category of "cloud management frameworks" because they are designed with provisioning and management of the infrastructure services that comprise a cloud computing (or highly dynamic) environment.

Stacks are aptly named as they attempt to provide management and specifically automation of provisioning for the complete network stack. Both CloudStack and OpenStack, along with Eucalyptus and Amazon and VMware vCloud, provide a framework API that can (ostensibly) be used to provision infrastructure services irrespective of vendor implementation. The vision is (or should be) to enable implementers (whether service provider or enterprise) to be able to switch out architectural elements (routers, switches, hypervisors, load balancers, etc…) transparently*****. That is, moving from Dell to HP to Cisco (or vice-versa) as an environment's switching fabric should not be disruptive. Physical changes should be able to occur without impacting the provisioning and management of the actual services provided by the infrastructure.

And yes, such a strategy should also allow heterogeneity of infrastructure.

In many ways, such "stacks" are the virtualization of the data center, enabling abstraction of the actual implementation from the configuration and automation of the hardware (or software) elements. This, more than anything, is what enables a comparison with flow-based models.

The Flow Model

Flow-based models, in particular OpenFlow-based SDN, also abstracts implementation from configuration by decoupling the control plane from the data plane. This allows any OpenFlow-enabled device (mostly switches today, as SDN and OpenFlow focus on network layers) to be configured and managed via a centralized controller using a common API.

Flows are "installed" or "inserted" into OpenFlow-enabled elements via OpenFlow, an open protocol designed for this purpose, and support real-time updates that enable on-demand optimization or fault isolation of flows through the network. OpenFlow and SDN are focused on managing the flow of traffic through a network.

Flow-based models purport to offer the same benefits as a stack model in terms of heterogeneity and interoperability. Moving from one OpenFlow-enabled switch to another (or mixing and matching) should ostensibly have no impact on the network whatsoever.

What flow-based models offer above and beyond a stack model is extensibility. OpenFlow-based SDN models using a centralized controller also carry with it the premise of being able to programmatically add new services to the network without vendor assistance. "Applications" deployed on an SDN controller platform (for lack of a better term) can extend existing services or add new ones and there is no need to change anything in the network fabric, because ultimately every "application" distills flows into a simple forwarding decision that can then be applied like a pattern to future flows by the switches.

The Differences

This is markedly different from the focus of a stack, which is on provisioning and management, even though both may be occurring in real-time. While it's certainly the case that through the CloudStack API you can create or delete port forwarding rules on a firewall, these actions are pushed (initiated) ***external*** to the firewall. It is not the case that the firewall receives a packet and asks the cloud framework for the appropriate action, which is the model in play for a switch in an OpenFlow-based SDN.

Another (relatively unmentioned but important) distinction is who bears responsibility for integration. A stack-based model puts the onus on the stack to integrate (via what are usually called "plug-ins" or "drivers") with the component's existing API (assuming one exists). A flow-based model requires the vendor to take responsibility for enabling OpenFlow support natively. Obviously the ecosystem of available resources to perform integration is a magnitude higher with a stack model than with a flow model. While vendors are involved in development of drivers/plug-ins for stacks now, the impact on the product itself is minimal, if any at all, because the integration occurs ***external to the component.*** Enabling native OpenFlow support on components requires a lot more internal resources be directed at such a project.

Do these differences make for an either-or choice?

Actually, they don't. The models are not mutually exclusive and, in fact, might be used in conjunction with one another quite well. A stack based approach to provisioning and management might well be complemented by an OpenFlow SDN in which flows through the network can be updated in real time or, as is often proffered as a possibility, the deployment of new protocols or services within the network.

The War that Isn't

While there certainly may be a war raging amongst the various stack models, it doesn't appear that a war between OpenFlow and *-Stack is something that's real or ever will be The two foci are very different, and realistically the two could easily be deployed in the same network and solve multiple problems. Network resources may be provisioned and initially configured via a stack but updated in real-time or extended by an SDN controller, assuming such network resources were OpenFlow-enabled in the first place.

***** That's the vision (and the way it should be) at least. Reality thus far is that the OpenStack API doesn't support most network elements above L3 yet, and CloudStack is tightly coupling API calls to components, rendering this alleged benefit well, not a benefit at all, at least at L4 and above.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.	F5 Networks	F5 Networks Ltd.	F5 Networks
Corporate Headquarters	Asia-Pacific	Europe/Middle-East/Africa	Japan K.K.
info@f5.com	apacinfo@f5.com	emeainfo@f5.com	f5j-info@f5.com