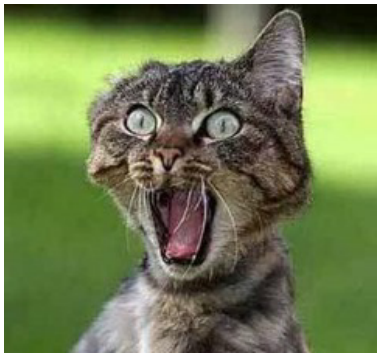
Virtual Apostasy

Lori MacVittie, 2013-10-04

#SDN #NFV #devops When all you have is a hypervisor, everything looks like it should be virtualized.



Yes, I'm about to say something that's on the order of heresy in the church of virtualization. But it has to be said and I'm willing to say it because, well, as General Patton said, "If everyone is thinking the same... someone isn't thinking."

The original NFV white paper cited in the excellent overview of the SDN and NFV relationships "NFV and SDN: What's the Difference?" describes essentially two problems it attempts to solve: rapid provisioning and operational costs.

The reason commodity hardware is always associated with NFV and with SDN is that, even if there existed a rainbow and unicorns industry-wide standard for managing network hardware there would still exist significant time required to acquire and deploy said hardware. One does not generally have extra firewalls, routers, switches, and application network service hardware lying around idle. One might, however, have commodity (cheap) compute available on which such services could be deployed.

Software, as we've seen, has readily adapted to distribution and deployment in a digital form factor. It wasn't always so after all. We started with floppies, moved to CD-ROM, then DVD and, finally, to neat little packages served up by application stores and centralized repositories (RPM, NPM, etc...).

Virtualization arrived just as we were moving from the physical to digital methods of distribution and it afforded us the commonality (abstraction) necessary to enable using commodity hardware for systems that might not otherwise be deployable on that hardware due to a lack of support by the operating system or the application itself. With the exposure of APIs and management via centralized platforms, the issue of provisioning speed was quickly addressed. Thus, virtualization is the easy answer to data center problems up and down the network stack.

But it isn't the only answer, and as SDN has shown there are other models that provide the same agility and cost benefits as virtualization without the potential downsides (performance being the most obvious with respect to the network).

ABSTRACT the ABSTRACTION

Let's abstract the abstraction for a moment. What is it virtualization offers that a similar, software-defined solution would not? If you're going to use raw compute, what is it that virtualization provides that makes it so appealing?

Hardware agnosticism comes to mind as a significant characteristic that leads everyone to choose virtualization as nearly a deus-ex machina solution. The idea that one can start with bare metal (raw compute) and within minutes have any of a number of very different systems up and running is compelling. Because there are hardware-specific drivers and configuration required at the OS level, however, that vision isn't easily realized. Enter virtualization, which provides a consistent, targetable layer for the operating system and applications.

Sure, it's software, but is standardizing on a hypervisor platform all that different from standardizing on a hardware platform?

We've turned the hypervisor into our common platform. It is what we target, what we've used as the "base" for deployment. It has eliminated the need to be concerned about five or ten hundred different potential board-level components requiring support and provided us a simple base platform upon which to deploy. But it hasn't eliminated dependencies; you can't deploy a VM packaged for VMware on a KVM system or vice-versa. There's still some virtual diaspora in the market that requires different targeted packages. But at least we're down to half-a-dozen from the hundreds of possible combinations at the hardware level.

But is it really virtualization that enables this magical deployment paradigm or is it the ability to deploy on common hardware it offers that's important? I'd say its the latter. It's the ability to deploy on commodity hardware that makes virtualization appealing. The hardware, however, still must exist. It must be racked and ready, available for that deployment. In terms of compute, we still have traditional roadblocks around ensuring compute capacity availability. The value up the operational process stack, as it were, of virtualization suddenly becomes more about readiness; about the ability to rapidly provision X or Y or Z because it's pre-packaged for the virtualization ***platform***. In other words, it's the readiness factor that's key to rapid deployment. If there is sufficient compute (hardware) available and *if* the application/service/whatever is pre-packaged for the target virtualization platform *then* rapid deployment ensues.

Otherwise, you're sitting the same place you were before virtualization.

So there's significant planning that goes into being able to take advantage of virtualization's commoditization of compute to enable rapid deployment. And if we abstract what it is that enables virtualization to be the goodness that it is we find that it's about pre-packaging and a very finite targeted platform upon which services and applications can be deployed.

The question is, is that the only way to enable that capability?

Obviously I don't think so or I wouldn't be writing this post.

COMPLACENCY is the GREAT INHIBITOR of INNOVATION

What if we could remove the layer of virtualization, replacing it instead with a more robust and agile operating system capable of managing a bare metal deployment with the same (or even more) alacrity than a comparable virtualized system?

It seems that eliminating yet another layer of abstraction between the network function and, well, the network would be a good thing. Network functions at layer 2-3 are I/O bound; they're heavily reliant on fast input and output and that includes traversing the hardware up through the OS up through the hypervisor up through the... The more paths (and thus internal bus and lane traversals) a packet must travel in the system the higher the latency. Eliminating as many of these paths as possible is one of the keys*** to continued performance improvements on commodity hardware such that they are nearing those of network hardware.

If one had such a system that met the requirements - pre-packaged, rapid provisioning, able to run on commodity hardware - would you really need the virtual layer?

No.

But when all you have is a hypervisor...

I'm not saying virtualization isn't good technology, or that it doesn't make sense, or that it shouldn't be used. What I am saying is that perhaps we've become too quick to reach for the hammer when confronted with the challenge of rapid provisioning or flexibility. Let's not get complacent. We're far too early in the SDN and NFV game for that.

* Notice I did not say Sisyphean. It's doable, so it's on the order of Herculean. Unfortunately that also implies it's a long, arduous journey.

** That may be a tad hyperbolic, admittedly.

*** The operating system has a lot - ***a lot*** - to do with this equation, but that's a treatise for another day

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

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com