



[Home](#) [Archives](#) [Profile](#) [Subscribe](#)

« [FCoE Case Studies TechBook update](#) | [Main](#) | [Coming soon: The next generation E-Lab Interoperability Navigator!](#) »

04/09/2014

Virtual Storage Networks - Part 1

We've been busy. For the past three years a group of people including Mark Lippitt, Vinh Dinh, Doug Fierro and I have been working on an idea that we're referring to as a Virtual Storage Network or VSN for short. The VSN concept seeks to take advantage of the major disruption currently underway in the networking space thanks to the rise of SDN. In fact, our work was inspired by what the team at Nicira (now NSX) had been able to accomplish when we were first introduced to them 3 years ago. Based on what they had done, we saw an opportunity to dramatically improve how we provide storage services to our end users. As a result, we set a goal to enable the dynamic creation of storage service compositions, automate connectivity between compute and these storage service compositions, and to do so in a way that is compatible with the concept of multi-tenant storage.

It's important to point out that while our work was inspired by Overlay Virtual Networks (e.g., VXLAN), we expect this work to be compatible with any transport that is capable of providing per tenant isolation. As a result, VLANs would also be acceptable today.

One final note before I begin, some of the concepts that I'll be describing as well as some of the Proof of Concept (PoC) work done in the lab would not have been possible (and probably would not even have occurred to us) without the guidance of Dave Cohen and Patrick Mullaney.

You may already have seen this...

At VMworld 2013, Chad Sakac [demonstrated](#) a portion of an EMC advanced development project named "Alta". The purpose of the project was to determine the feasibility of running block based protocols (in this case iSCSI) over a VXLAN based [virtual network](#).



[Erik Smith](#)

[Follow](#)

[ARCHIVES](#)

[September 2022](#)

[August 2020](#)

[April 2019](#)

[December 2017](#)

[November 2017](#)

[July 2017](#)

[April 2017](#)

[September 2016](#)

[July 2016](#)

[February 2016](#)

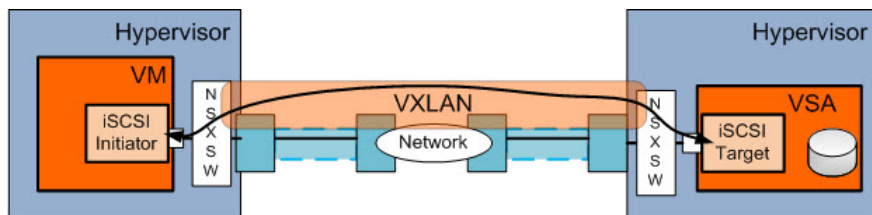
SEARCH

[CATEGORIES](#)

[Subscribe to this blog's feed](#)

DISCLAIMER

This is not an official EMC blog. The opinions expressed here are my personal opinions. Content published here is not read or approved in advance by EMC and does not necessarily reflect the views and opinions of EMC nor does it constitute any official communication of EMC.



While the demo and lab experiments proved that this was possible, the demo intentionally left out a number of details such as why you would want to do this or what other benefits could be realized by following this kind of approach. This blog post and the ones that follow it will attempt to address these questions and also describe the project that we're currently working on named "Blackhawk".

Background

When Fibre Channel was introduced to our Customers back in the mid to late 90s, it solved a couple of connectivity problems related to physical SCSI. First, it increased the number of servers (Initiators) that could be connected to each storage array interface (Target) and second it increased the maximum distance that could be supported between them. As is frequently the case, these scalability and reach improvements came with an associated cost. In this case, the cost was the need to manually define which initiators were allowed to access which targets via the FC fabric. This connectivity definition is referred to as a zone and the process of creating a zone and adding it to a configuration or zone set must be repeated [once per initiator](#). To this day zones are still required in both FC and FCoE environments for reasons that are beyond the scope of this blog post. For now, it will suffice to say that zoning allows for multiple users to simultaneously use an FC/FCoE SAN and their presence reduces the load on the Name Server allowing fabrics to scale to sizes that would otherwise be impossible.

One issue that was discovered early in FC's life was the need for LUN masking.

Although the exact details are also outside the scope of this post, it was created to solve an issue caused by a pair of UNIX admins who were unable to nicely share the Logical Units (LUs) being presented behind a single target interface that both had been zoned to have access to. Since this sort of problem limited the value of the connectivity benefits described earlier, a couple of EMC engineers discussed alternatives and came up with the concept of using the Initiator WWPN to limit which Logical Unit Numbers (LUNs) could be accessed. The downside to this solution was that it required administrative configuration, but this was acceptable given the connectivity benefits our customers could realize with FC.

Since then, due to an overwhelming customer demand for automation, there have been many attempts to create a solution that would automate the storage provisioning process (e.g., FC zoning, LUN creation and LUN masking), my personal favorites are [TDZ](#) and the [UFC](#). That having been said, there are also products such as EMCs ViPR that do a good job and are actually available for our customers to use today. In the future, we can expect to see other approaches to zoning automation appear in the market. One example is the work that is being done on [OpenStack Cinder](#) by a group of Engineers from companies including Brocade, EMC and HP. See this [blueprint](#) for more information.

While all of this work will certainly help automate the storage provisioning process, the problem is the solutions used to realize them will only be

applicable in environments that both use FC and have the connectivity automation (e.g., ViPR, Cinder FC Zoning service) software installed. This is concerning to us because many customers, especially those in the Service Provider space, are not interested in using FC in their environments and as a result we cannot rely solely on FC solutions to automate connectivity to our storage arrays. Also many service providers, want to “roll their own” orchestration layer and this means solving the problem solely with ViPR or OpenStack may not be applicable to the fastest growing segment in the IT space.

I should point out that OpenStack somewhat automates connectivity between compute and Storage today when iSCSI is used. I view this as a good thing but as I’ll describe later on, this approach doesn’t completely solve the connectivity automation problem.

So with that in mind, what I’m about to describe is an architectural enhancement that ViPR, vCAC, OpenStack, etc may (or may not) eventually decide to embrace. The reason we feel this enhancement is needed should come as no surprise, competition...

Enter Server SAN

I tend to get very excited when I hear people describe [Server SAN](#) as a panacea. I have no doubt that ScaleIO, VMware VSAN, etc will prove to be useful and economical in many situations going forward ([others are even a bit more optimistic](#)). The specific use cases that intrigue me today are dev/test, remote office locations (where an array may not make sense) and maybe with VDI. That having been said, it’s interesting to note that for the most part, these Server SAN solutions do not provide many features that customers consider “Enterprise class” such as remote replication, dedup, FAST or multi-tenant isolation. I also have concerns about the amount of network bandwidth that will be consumed to replicate data between storage nodes and the CPU cycles that are consumed not only today but in the future as the various Server SAN vendors try to add Enterprise class features. To be fair, I do realize that not all environments are 100% utilized and therefore something approaching equilibrium will eventually be found between environments that are suitable for Server SAN and those that are not.

One area where a Server SAN solution, particularly VSAN, seems to excel is ease of use. While listening to the VSAN user panel at VMworld last year, I was blown away with not only how simple VMware had made the actual provisioning process, but also with how well this approach would lend itself to automatically scale without the need to provision networking and storage. Listening to the moderator and panelists talk, one could imagine data centers filled with “Pods” consisting of nothing but compute and network resources. Although these pods would have inherent scalability limitations, they could scale to a reasonable size, be deployed repetitively, be managed as a Federation and eliminate all need for external storage arrays... Since I work for a company that is known for world class Enterprise Storage Arrays, you can imagine I found this kind of talk, ah, um, ALARMING.. :-). Especially since at least a few companies have figured out how to get some of these concepts to actually work (e.g., [Amazon](#)). After listening to these panelists, a number of things prevented me from giving into despair and “jumping out of the nearest window”:

1. I believe that customers value enterprise class features,
2. Server SAN solutions will find these features difficult to implement,

3. These Server SAN solutions are unable to meet many of the following multi-tenant storage requirements that were gathered by working with a number of Service Providers interested in providing IaaS solutions.

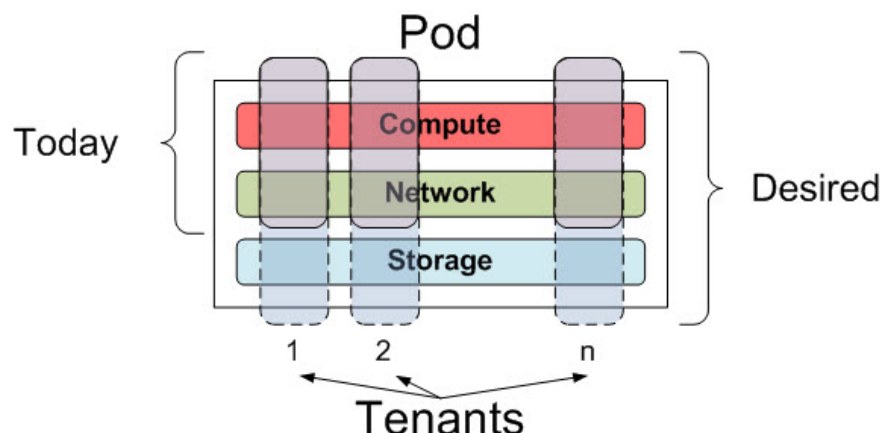
Service Providers and multi-tenant storage requirements

For a while now we've been talking with some of our customers in the service provider space about their requirements for next generation multi-tenant storage services and how these storage services are likely to be used and exposed to their end users. Before I dive into these storage requirements, it's important to note that this is not an exhaustive list. The requirements I'll be sharing are only those directly related to storage connectivity. You'll notice that absent in the requirements and the subsequent solutions are most of the details that describe how the storage or storage service will be consumed by each tenant. This will undoubtedly strike some of you as odd because I spent a fair bit of time at the beginning of this post describing how Customers want things to be automated. My answer is simply this, in order to create the necessary higher level abstractions and provide things like automated connectivity and differentiated per-tenant services; the underlying infrastructure must undergo some fundamental changes. The remainder of this post and the ones that follow will be dedicated to describing the requirements we captured, the fundamental infrastructure related changes needed to support those requirements and how connectivity automation will be used to facilitate this.

It's important to note that the entire concept of multi-tenancy and its application to storage is still very much in its infancy. The requirements that I'm going to share and the solutions being proposed to address them are just that, proposals. Although we've spent a fair bit of time in the lab trying to get different pieces of this to work, there's a TON left to do. As a result, please think of the remainder of this VSN series as an attempt to start a discussion and not a lecture on how things ought to be done.

Storage and Tenants

One last point that I have to make before I get into the storage requirements themselves is the current state of tenancy and how it relates to storage. From the perspective of many in the industry, the concept of a tenant really only includes Compute and Network elements. Oh sure, you can allocate storage to a particular tenant and other tenants will not be able to access that storage, but the concept of tenancy typically doesn't extend all the way to the storage tier.



At best, today the tenant concept can be partially and somewhat clumsily extended to include network elements using a variety of means, some of which will be discussed in the following section. Again, fully extending a tenant's "personality" all the way to the Storage tier matters because many of the requirements I'll be describing below cannot be fully satisfied otherwise.

Requirements for multi-tenant storage

As stated before, the following requirements were gathered by speaking with several of our customers who are interested in multi-tenancy. The order does not indicate priority.

1. Namespace isolation
2. Prevent noisy neighbor problems
3. Provide Bandwidth / IOPS / response time guarantees
4. Tenant traffic identification
5. Storage Network Service Insertion (e.g., encryption)
6. Authentication (e.g., CHAP)
7. Do not rely on Guest OS resident iSCSI initiators

In [part 2](#) of this VSN series, I will provide an in-depth explanation for each of these requirements. In [part 3](#), I'll describe a couple of topologies that either partially or fully satisfy these requirements and then in part 4, I'll describe how we think we can automate connectivity.

Thanks for reading!

Posted at 10:06 AM | [Permalink](#)



Comments

You can follow this conversation by subscribing to the [comment feed](#) for this post.

The comments to this entry are closed.

[Brass Tacks](#) [Powered by TypePad](#)