Team 8 Presentation
by Arturo Urbina & Gilberto Diaz

Creating a Scalable Storage Architecture

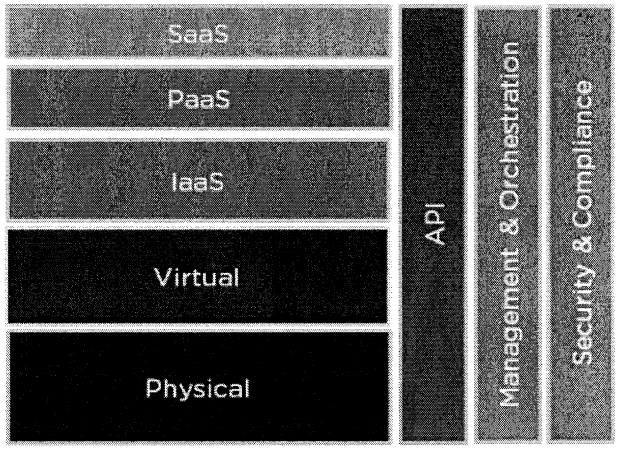
The Complete Cloud From the Ground Up

Top	Examples
Exposable Cloud API	Microsoft Azure, vCloud Director (part of vCloud) and vCloud API, AWS, others sections of the management services found below and their related APIs
Cloud Management Services (for automated IaaS, PaaS and SaaS)	Eucalyptus, Cloud Stack, vCloud (vSphere), Microsoft Systems Center 2012, Citrix's CloudPlatform
Host hypervisors (possibly heterogeneous in the future)	ESXi, Xen, Hyper-V
Physical Hardware	Dell or HP servers
Bottom	

(1)

Figure 1 shows the fundamental structure and components of the cloud computing stack.

Figure 1. System Architecture



This presentation is about designing and implementing hypervisor level storage architecture that enables the third layer of the stack (the management services) to act upon the lower layers with the greatest ease possible, that is, to implement the hardware architecture required by the management services necessary for cloud computing. To facilitate this process, the standard solution is to separate layers and the tasks within each layer as much as possible.

.

The root of the problem: How does one store data?
Hypervisor Storage options:

- local based storage -parallel SCSI, ATA
- network -NFS, iSCSI, Fibre Channel over Ethernet (FBoE)
- traditional SAN such as Fibre Channel

Why use network based storage or Fibre Channel? Why not just use the preconfigured local datastores in ESXi/Hyper-V or configure the local storage for XCP/XenServer/?

Problems when using local storage (severe):

- If storage fails then the VMs fail
- If host fails then the VMs fail
- Ability to create backups depends on the hypervisor interface

Overall problem: The layers between the physical storage and a VM's virtual storage are not significantly abstracted and hence local-storage based configurations are **not scalable**.

Industry Standard Solution:
Try to divide the host and storage so that each object is resilient to a failure of the other.

Reasons to use network storage:

- Easier to create robust storage systems, minimizing the possibility of total storage failure
  - Nodes dedicated to storage simplify maintenance
- Possible pickup of a VM after an host failure
  - This part can be automated or made transparent to the upper layers of a cloud when using g a management platform like vSphere, MS Systems Center, OpenStack, CloudStack
  - e.g. possible for the loss of a host to have no effect on upper layers
- Centralized storage simplifies backup process of storage (redundant storage arrays)
  - Possible pickup of a VM after a storage array failure
- cheaper than Fibre Channel (unless one is using FCoE)
- Notice how this solution is well supported (VMware has a 222 page guide on implementing ESXi with every conceivable storage container compatible with vCenter (2).) and and has many specific implementation options.

Solution 2:
An alternative approach to this problem was presented on Tuesday with the "Sheepdog Project" that attempts to solve the issue by merging the host and the storage objects instead of trying to separate them. The approach was to try and creating redundant local storage by replicating data to each host in the cluster thus theoretically allowing the adding and drooping of hosts seamlessly. Notice how this approach is not ready to implement.

Why this cannot work:

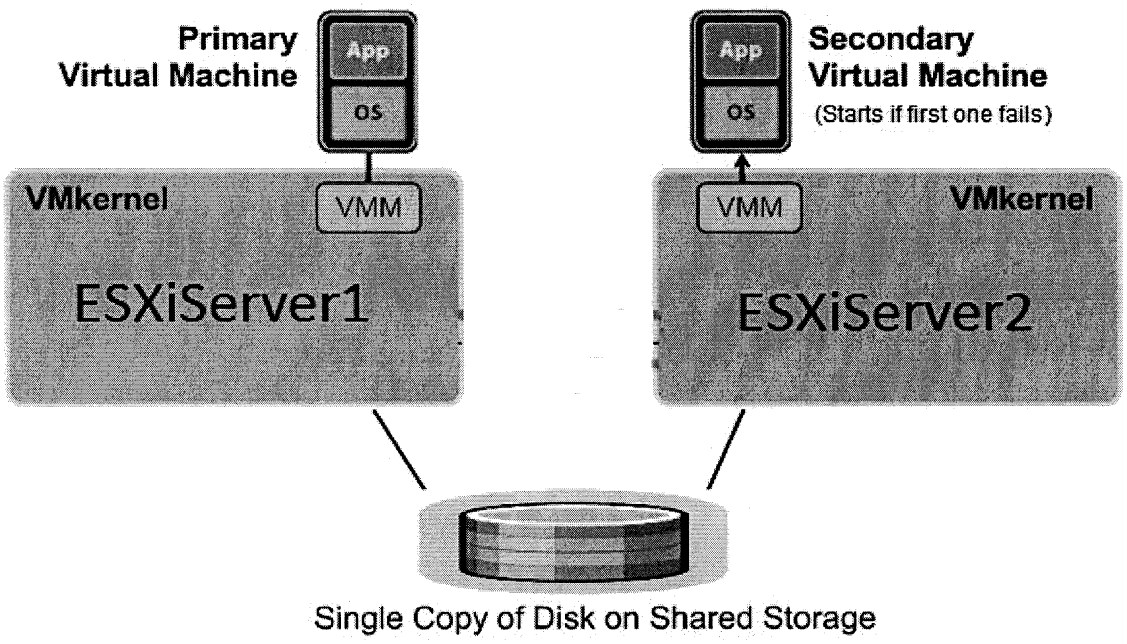
- E1) Concept of a stack is that the upper layers (cloud management) should be agnostic to the bottom layers (hosts used). For example, in the TCP/IP stack, Ethernet (Layer 2) is agnostic as to whether it's delivering IP Packets or FCoE packets.
- E2) Installation requirements for XCP/CloudStack (very flexible hypervisor) essentially require non-local storage configurations.
- Conclusion: Layers in cloud "stacks" are not fully independent at this time and hence as the industry moves forward with cloud computing and refines the stack to be more layer agnostic, approaches that do not fit this model will cease development.
- Also: Spend time to learn one of the cloud stacks in detail instead of focusing on various ones. J

Lesson: Different ways to solve the same problem, but which approach you take is usually dictated by what is best supported.

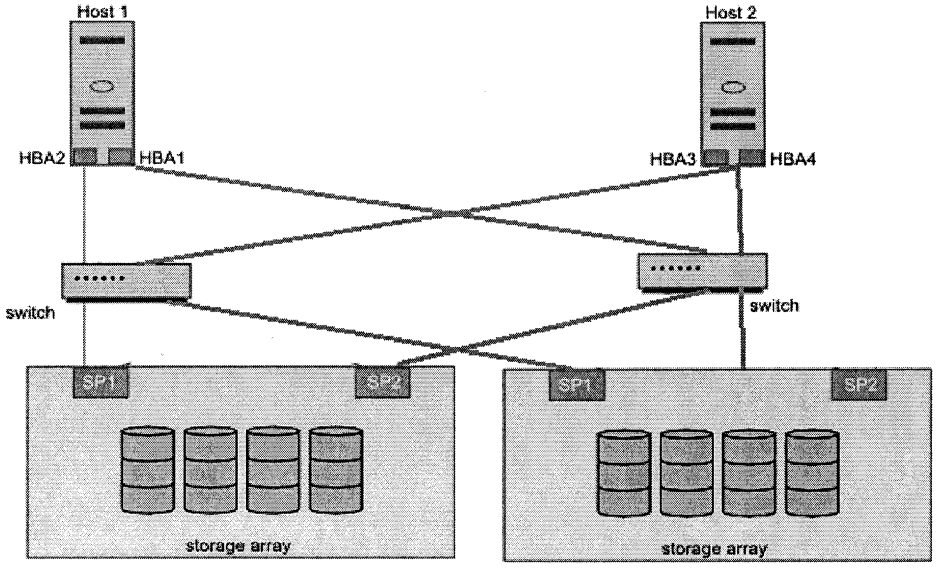
Demo:
Specifically talking about ESXi using NFS shares (on CentOS and FreeNAS)

- NFS is very well supported and inexpensive to implement. NFS also maintains an architecture that still has some scalability by supporting upper layer cloud features and hence is the best choice for cloud implementations where the budget doesn't allow for iSCSI or a full SAN.
- Centralized NFS shares also make it possible to duplicate the entire storage server easily.
- Cloud management software would manage syncronization between backups of the storage and various hosts however. Without the management software, this must be done manually.

Implementation Guide for Demo:
https://docs.google.com/open?id=0Bx3It8OSZLKcVUlDVlVCS09SbXc or tinyurl.com/bvbovyx



(3)



(4)

Sources:

1) http://www.vmware.com/files/pdf/vcat/Architecting-VMware-vCloud.pdf
2) http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-50-storage-guide.pdf
3) tinyurl.com/cyoyogv
4) tinyurl.com/bun8agb