

[The Grid Approach to Desktop Virtualization](#)

Posted By [Bernd Harzog](#) on February 19, 2010

Both VMware and Citrix, the two clear early leaders in desktop virtualization have built solutions that leverage the infrastructure of server virtualization. However there are a couple of vendors who have come up with a completely different approach. These two vendors, [Kaviza](#), and Synchron use a grid computing approach reminiscent of Citrix Presentation Server (now XenApp) that delivers some interesting benefits, and also comes with some tradeoffs. The grid approach to desktop virtualization is characterized by the following attributes:

- While the grid approach to desktop virtualization (what we call hosted virtual desktops or HVD) absolutely requires a hypervisor on the back end servers, it does not really matter what the hypervisor is, or from which vendor it is. Some of these solutions can even mix and match hypervisors from competing vendors within one group of virtualized desktops.
- The choice of server hardware is opened up and can be much more flexible. Within one group of servers supporting a virtualized desktop you can mix and match different generations of CPU's, different CPU vendors, and servers from different vendors. In general any server that can run a hypervisor from any of the supported hypervisor vendors is an acceptable member of the pool of servers that can host virtual desktops
- Shared virtual machine storage (SAN, NFS, iSCSI) is not required. The servers that comprise the back end pool of resources for the hosted virtual desktops can make use of their local storage for the guests. In general not a great deal of storage is required since these solutions typically treat the guest as a transient entity that is created as the user logs on, and then destroyed after the user logs off
- Shared storage of some form, is of course required. A file server of some type is needed to store user data, the user profiles and any other customizations to settings that are not stored in the registry.

The tradeoffs to this approach are twofold. The first is that without shared virtual machine storage, a VMotion of a guest from one server to another is not possible. This means that a variety of VMotion derived capabilities are not present. This includes the ability to manually move the guests from a host that needs maintenance, the ability to move them when the system senses an imminent hardware failure (HA), and the ability to move them for dynamic load balancing reasons (DRS). The second tradeoff is that since guests are typically created and destroyed, the management of the user customizations, applications customizations, and user data becomes of paramount importance. Therefore with this approach steps needs to be taken to ensure that either 1) users will not be allowed to persistently customize their environments (the call center scenario), or that 2) what is known as the Windows Profile is carefully managed so that it changes to it are kept up to date in an incremental manner and so that the profile does not get corrupted.

Without a doubt this analysis will come back to how big of an issue is it for there to be no shared virtual machine storage, no VMotion, no HA and no DRS for virtual desktops? There is a very interesting data point that is illustrative in terms of helping us understand how important (or not important) these features are for virtualized desktops (no one doubts how essential they are for virtualized servers). That data point is the success of Citrix Presentation Server (XenApp). XenApp is in use by over 230,000 customers, and there are several million XenApp servers in production. There has never been VMotion, HA, and DRS for XenApp (or for that matter Terminal Services) – unless of course you virtualize the XenApp servers. But back to the main point, the method by which this issue was managed for XenApp has been proven to have been very acceptable for a large number of users over a large number of years. That method is to set a server to accept no more incoming connections, let it drain of users as they log off, force off the remaining users, do the maintenance, and then put the server back in the pool. If this method was good enough for millions of XenApp servers, why is it not good enough for the virtual desktop sessions in a hosted virtual desktop environment? Especially if as a result the need for shared virtual storage can be eliminated.

This is an area where being owned by EMC is clearly not in VMware's best interests. EMC sees VMware as a driver of more high end storage sales (which server virtualization has certainly proven to be). If VMware were not owned by EMC, VMware would probably be free to at least offer an option to customers that does not require shared virtual machine storage. However given the current situation we will have to leave the exploration of this alternative up to creative startups like [Kaviza](#) and Synchron. The wild card in this equation is certainly Citrix. Being the company that made XenApp successful and being the company that appears to have no further interest in XenServer as the back end (which requires a SAN), Citrix is certainly freed up to pursue the same approach for desktop virtualization that they successfully leveraged with XenApp. Could it happen? Could Citrix change the rules of the game in desktop virtualization with a "back to the future move". Only time will tell. One thing is for sure. The approach is available right now from creative startups who have already seen the opportunity.

About the author

Bernd Harzog is the Analyst at The Virtualization Practice for Performance and Capacity Management and Virtualization Management. Bernd is also the CEO and founder of APM Experts a company that provides strategic marketing services to vendors in the virtualization performance management, and end user experience management markets. Prior to these two companies, Bernd was the CEO of RTO Software, the VP Products at Netuitive, a General Manager at Xcellenet, and Research Director for Systems Software at Gartner Group. Bernd has an MBA in Marketing from the University of Chicago.