Mike Nelson - SSV Part 2 article

[Optional title] Achieving Virtualization Success – Part 2

Now that you have your virtualization plan down, testing the servers you plan on moving to virtualization is the next important step. Doing some basic testing is probably the most opportune way to get firsthand knowledge of how the installed applications will perform and what impact may or may not be felt from the user's perspective. And, while testing may not have the same application or user load as production, it does give you some valuable insight into what may be coming ahead. After all, it is exactly that – testing.

When I move forward with a plan for virtualizing some of the "Heavy Hitters", as I mentioned in the previous article, onto virtualized hardware, the testing phase is equally as important as all the others. By performing some quick scenarios and load tests with all the application pieces installed, and hopefully some willing users, you can start turning the knobs and dials within the hypervisor and the applications themselves to start the fine tuning process. Keep in mind that it is also reality that not everyone has the luxury of having a full-blown, completely stocked testing lab to work with, or even the available users to do he actual application testing. Thankfully, some software manufacturers even produce load testing software, both from the application vendors themselves or by third-party tools, to get a more accurate appraisal of the environment. I have always said that while performance testing and monitoring may be considered by some to be a science, I consider it more of an artform, which allows for creativity.

The user experience should be a partial consideration when planning and implementing your tests, but in my opinion, if the applications run as well or better than when they were hosted on physical hardware, there isn't much to worry about when it comes to the users expectations. That is not to say that if the application is visually intensive, and is not a true client-server application, but rather a simple RDP session accessed application, that some concessions may have to be made to ensure success. Those factors will need to be addressed for each server, and consequently, each application.

There are some other things to consider and questions to ask before you proceed with testing. Another point to keep in the back of your mind is that some will most likely come to the surface later after testing is done and the data is compared. Here's a short list of what I look for:

- Host Affinity. Is it possibly a requirement from the vendor? Does the application require that the server it is hosted on not be on the same host, or even in the same DRS or HA cluster as another? Is it a requirement, or does it just make sense?
- Resource Affinity. Again, a requirement from the vendor? Is it necessary to dedicate resources of the Core Five - CPU, RAM, Disk, NIC, or USB - to ensure performance or support (It used to be only the Core Four, but now USB has entered the mix)?
- Shares and Resource Pools. In most of my experience, these actually don't enter into the equation until the end, but it should be considered up front.
- Dedicated Clusters. I've implemented some clusters with a few as three hosts and as many as 16, and it has worked out well in most situations. Isolating a set of servers that work as a group or have common, and possibly interacting applications between them can be beneficial, not only from a performance perspective, but also from a management one.
- Application Stacking. Maybe you won't be hosting just a single big application on the server. Maybe you'll want to add two or three others. But right now, those two or three others are all on separate physical servers. With the bigger applications, I would normally advise against it. But with the ultra-fast and capacity heavy hosts that can be built today and supported by the

hypervisors, it is a very feasible concept that should be examined. Stacking can be a great thing, but it can also ruin your day as an I.T. Admin. Which brings me to my last point….

- Doesn't play well with others or by themselves. Reality check time. Some applications just don't play well in the same sandbox as others. It's a fact and one that we know from the physical server world all too well. Maybe it not only doesn't play well with other applications on the same guest, host, or cluster, but maybe it just cannot be virtualized period. I find it hard to believe, but I do know that there are some out there, and hopefully only time and good developers will move them forward.

Lastly, while entering In your testing phase, be sure to perform the basic prerequisites required to successfully measure the performance of a server. Things like creating a solid baseline of both servers and the attempt to create load that is as close as what is being done on the physical server today. Sometimes it's not entirely possible to do this completely and with 100% accuracy, but you should be able to get close. Remember; think of it as an art, not as a science (in my opinion, of course).