

Can application collapsing and/or stacking be greener than virtualization?

By Ron Kaminski

There is a lot of hype and hyperbole about the hardware density improvements and the supposed “green” aspects that are attributed to virtualization. It is important to realize that many of these same improvements are possible without any virtualization at all. There are costs, risks and rewards in many different strategies, so let’s explore a few.

Let’s start by defining some terms.

- **“Virtualization”** is taking existing system images and moving them under some vendor’s virtualization scheme, usually also handing that vendor a significant chunk of cash at the same time, which I call the “virtualization tax”.
- **“Application Collapsing”** is reducing the number of operating system (hereafter OS) images needed by an application, collapsing it down to fewer machines.
- **“Application Stacking”** is taking multiple small unrelated applications, often with similar recovery priority needs, and stacking them up together on the same machine.

When you look at an application that was probably written many years ago, when machines were much smaller, designers would often recommend multiple separate machines for the database servers, web servers, application servers and load balancers, etc. and depending on the application’s importance and your firm’s level of paranoia, there can also be failover hardware copies and possibly extra machines added for “scaling” if there was growth uncertainty. The increased speed of modern servers and disk technologies have left many of these multi-node machines maxing out at 3% to 7% utilizations, and by now many of these application’s scaling needs are now long known. The result is what many of us see every day, a lot of extremely low use servers, and a lot of operating system patching, machine maintenance and machine proliferation.

The “**virtualization** hype” crowd would have you believe that simply virtualizing all of those separate images would let you run them on just a few virtual hosts, and that is the way to get better hardware utilizations. What few talk about is sometimes these applications will encounter “issues” that vary by virtualization vendor, but that can impact performance, throughput or even cause application outages. I recently encountered application failures related to the way that a certain virtualization vendor tried to reclaim what it viewed as “extra memory” but what the application viewed as memory that it wanted to allocate. What we learned changed the way that we set up dozens of other virtual images, to avoid similar issues in future.

How many other issues of this type are lurking out there in the virtual world, and when will they strike your applications? Many virtual environments are becoming increasingly complex as versions go by and new features are added. Have you invested in staff training and staffing levels that let you know about or find these new issues quickly? Also, no one denies that most virtualization schemes add overhead to disk IOs, the slowest part of most modern business systems. Does making the slowest part of our processing even slower sound like a good idea to anyone?

Another route to the same higher hardware density is **application collapsing**, simply running previously separate functions together on the same OS image, and often with a surprising performance benefit. Let's face it, the slowest part of a chatty multi-node application is often the network links, and when you run the application server, database server and, web server together on the same OS image, all of that network lag disappears, greatly enhancing performance and the remaining modern machine is probably kicking over at 6% utilization. True, there will be some minimal re-engineering costs, but if you seize on this opportunity to get the old application up to a modern version of the OS too, you multiply the effectiveness of your spending and possibly lower future maintenance costs as well. Every application that we have done this with was thrilled by the performance gains and the operations folks loved having less OS images to maintain.

A good friend of mine points out that sometimes organizational policy will forbid application collapsing. These policies tend to be present at sites undergoing massive growth with massive web presences and security issues and often are not sites with forests of dull, little used application servers. Let's be clear, application collapsing works best with applications with a well known history that are not facing rapid growth. That said, it does work well with a significant portion of the servers in many large firms.

Having tried it multiple times, I can tell you that the most controversial way to reduce OS image count is **application stacking**. That isn't to say that it is any more risky if managed properly, but it can be politically unpalatable to the various project teams. All project leaders see their role as reducing risk, and having plenty of extra hardware that nobody else can get to makes them calm. Multiply that by hundreds of applications and you have a modern firm's empty hardware sprawl.

As a firm believer in "eating my own dog food" I decided to lead by example and allowed completely unrelated applications to be placed on my capacity planning servers, which are busy all night but have relatively low used during the day. It seemed like a perfect place for an "office hours" application to live, with the early evening left for maintenance activities. What happened? Well, my capacity planning servers had an outage because the DBAs unloaded the other application's database for maintenance and filled one of my capacity planning disk drives! This highlights a risk that virtualization does protect you from, which is unintended outages from folks not sticking to their own volumes. We have since declared volumes to be owned by applications and we don't allow folks to use another's and now it is working well.

When you look deeper at application stacking, you will end up grouping applications by their restore importance, so high priority servers aren't waiting for you to also recover a low priority application stacked with them. You will also find some applications that don't play well together due to various conflicts, but actually far fewer than you might think. You might also encounter contracts that forbid anything else to run with an application. We then put that vendor on an "avoid in future" list, and try hard to not sign contracts like that again.

It is important to remember that these are not "either/or" choices. There is no reason that you can't collapse an application first and then virtualize the result. Then you get the benefits of the virtualization

vendor's automated failover and availability features (reducing the need for idle failover machines for important applications), but you paid far less "virtualization tax" to get it.

In the end there are benefits to all three strategies, either separate or in combination. Just don't feel like you can't be "green IT" without parting with a lot of green to a virtualization vendor. Maybe you can get all of the benefits of each strategy, at a lower total cost. Your shareholders will thank you.