WILS: Moore&rsquo;s Law + Application (Un)Scalability = Virtualization

Lori MacVittie, 2012-23-07

#Virtualization was inevitable.

One of the interesting side effects of having been a developer before migrating to a more network-focused view of the world* is it’s easier to understand the limitations and constraints posed on networking-based software, such as web servers.

During the early days of virtualization adoption, particularly related to efforts around architecting more scalable applications, VMware (and others) did a number of performance and capacity-related tests in 2010 that concluded “lots of little web servers” scale and perform better than a few “big” web servers.

> Although virtualization overhead varies depending on the workload, the observed 16 percent performance degradation is an expected result when running the highly I/O intensive SPECweb2005 workload. But when we added the second processor, the performance difference between the two CPU native configuration and the virtual configuration that consisted of two virtual machines running in parallel quickly diminished to 9 percent. As we further increased the number of processors, the configuration using multiple virtual machines did not exhibit the scalability bottlenecks observed on the single native node, and the cumulative performance of the configuration with multiple virtual machines well exceeded the performance of a single native node.
>
> -- “Consolidating Web Applications Using VMware Infrastructure” [PDF, VMware]

The primary reason for this is session management and the corresponding amount of memory required. Capacity is a simple case of being constrained by the size of the data required to store the session. Performance, however, is a matter of computer science (and lots of math). We could go through the Big O math of hash tables versus linked lists versus binary search trees, et al, but suffice to say that in general, most algorithmic performance degrades the larger N is (where N is the number of entries in the data store, regardless of the actual mechanism) with varying performance for inserts and lookups.

Thus, it is no surprise that for most web servers, hard-coded limitations on the maximum number of clients, threads, and connections exist. All these related to session management and have an impact on capacity as well as performance. One assumes the default limitations are those the developers, after extensive experience and testing, have determined provide the optimal amount of capacity without sacrificing performance.

It should be noted that these limitations do not scale along with Moore’s law. The speed of the CPU (or number of CPUs) does impact performance, but not necessarily capacity – because capacity is about sessions and longevity of sessions (which today is very long given our tendency toward Web 2.0 interactive, real-time refreshing applications).

This constraint does not, however, have any impact whatsoever on the growth of computing power and resources. Memory continues to grow as do the number of CPUs, cores, and speed with which instructions can be executed.

What the end result of this is that “scale up” is no longer really an option for increasing capacity of applications. Adding more CPUs or more memory exposes the reality of diminishing returns. The second 4GB of memory does not net you the same capacity in terms of users and/or connections as the first 4GB, because performance degrades in conjunction with increase in memory utilization. Again, we could go into the performance characteristics of the underlying algorithms where resizing and searching of core data structures becomes more and more expensive, but let’s leave that to those so inclined to dig into the math.

The result is it shouldn’t have been a surprise when research showed “lots of little web servers”, i.e. scale out, was better than “a few big web servers”, i.e. scale up.

Virtualization - or some solution similar that enabled operators to partition out the increasing amount of compute resources in such a way as to create “lots of little web servers” – was inevitable because networked applications simply do not scale along with Moore’s Law.

*A Master’s degree in Computer Science doesn’t hurt here, either, at least for understanding the performance of algorithms and their various limitations

- Lots of Little Virtual Web Applications Scale Out Better than Scaling Up
- Consolidating Web Applications Using VMware Infrastructure
- To Take Advantage of Cloud Computing You Must **Unlearn, Luke**.
- It’s 2am: Do You Know What Algorithm Your Load Balancer is Using?
- Virtual Machine Density as the New Measure of IT Efficiency