Achieving Scalability Through Fewer Resources

Lori MacVittie, 2011-04-01

Sometimes it's not about how many resources you have but how you use them

The premise upon which scalability through cloud computing and highly virtualized architectures is built is the rapid provisioning of additional resources as a means to scale out to meet demand. That premise is a sound one and one that is a successful tactic in implementing a scalability strategy.

But it's not the *only* tactic that can be employed as a means to achieve scalability and it's certainly not the most efficient means by which demand can be met.

WHAT HAPPENED to EFFFICIENCY?

One of the primary reasons cited in surveys regarding cloud computing drivers is that of efficiency. Organizations want to be more efficient as a means to better leverage the resources they do have and to streamline the processes by which additional resources are acquired and provisioned when necessary. But somewhere along the line it seems we've lost sight of enabling higher levels of efficiency for existing resources and have, in fact, often ignored that particular goal in favor of simplifying the provisioning process.

After all, if scalability is as easy as clicking a button to provision more capacity in the cloud, why wouldn't you?

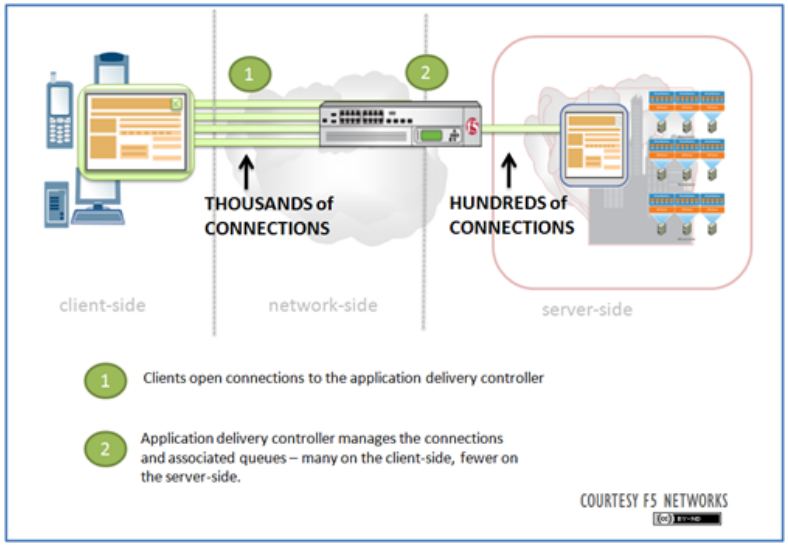
The answer is, of course, that it's not as efficient and in some cases it may be an unnecessary expense.

The danger with cloud computing and automated, virtualized infrastructures is in the tendency to react to demand for increases in capacity as we've always reacted: throw more hardware at the problem. While in the case of cloud computing and virtualization this has morphed from hardware to "virtual hardware", the result is the same – we're throwing more resources at the problem of increasing demand. That's not necessarily the best option and it's certainly not the most efficient use of the resources we have on hand.

There are certainly efficiency gains in this approach, there's no arguing that. The process for increasing capacity can go from a multi-week, many man-hour manual process to an hour or less, automated process that decreases the operational and capital expenses associated with increasing capacity. But if we want to truly take advantage of cloud computing and virtualization we should also be looking at optimizing the use of the **resources** we have on hand, for often it is the case that we have more than enough capacity, it simply isn't being used to its full capacity.

CONNECTION MANAGEMENT

Discussions of resource management generally include compute, storage, and network resources. But they often fail to include connection management. That's a travesty as TCP connection usage is increases dramatically with modern application architectures and TCP connections are resource heavy; they consume a lot of RAM and CPU on web and application servers to manage. In many cases the TCP connection management duties of a web or application server are by far the largest consumers of resources; the application itself actually consumes very little on a per-user basis.



Optimizing those connections – or the use of those connections – then, should be a priority for any efficiency-minded organization, particularly those interested in reducing the operational costs associated with scalability and availability. As is often the case, the tools to make more efficient the use of TCP connections is likely already in the data center and has been merely overlooked: the application delivery controller.

The reason for this is simple: most organizations acquire an application delivery controller (ADC) for its load balancing capabilities and tend to ignore all the bells and whistles and additional features (value) it can provide. Load balancing is but one feature of application delivery; there are many more that can dramatically impact the capacity and performance of web applications if they employed as part of a comprehensive application delivery strategy.

An ADC provides the means to perform TCP multiplexing (a.k.a. server offload, a.k.a. connection management). TCP multiplexing allows the ADC to maintain millions of connections with clients (users) while requiring only a fraction of that number to the servers. By reusing existing TCP connections to web and application servers, an ADC eliminates the overhead in processing time associating with opening, managing, and closing TCP connections every time a user accesses the web application. If you consider that most applications today are Web 2.0 and employ a variety of automatically updating components, you can easily see that eliminating the TCP management for the connections required to perform those updates will decrease not only the number of TCP connections required on the server-side but will also eliminate the *time* associated with such a process, meaning better end-user performance.

INCREASE CAPACITY by DECREASING RESOURCE UTILIZATION

Essentially we're talking about increasing capacity by decreasing resource utilization without compromising availability or performance. This is an application delivery strategy that requires a broader perspective than is generally available to operations and development staff. The ability to recognize a connection-heavy application and subsequently employ the optimization capabilities of an application delivery controller to improve the efficiency of resource utilization for that application require a more holistic view of the entire architecture.

Yes, this is the realm of devops and it is in this realm that the full potential of application delivery will be realized. It will take someone well-versed in both network and application infrastructure to view the two as part of a larger, holistic delivery architecture in order to assess the situation and determine that optimization of connection management will benefit the application not only as a means to improve performance but to increase capacity without increasing associated server-side resources.

Efficiency through optimization of resource utilization is an excellent strategy to improving the overall delivery of applications whilst simultaneously decreasing costs. It doesn't require cloud or virtualization, it simply requires a better understanding of applications and their underlying infrastructure and optimizing the application delivery infrastructure such that the innate behavior of such infrastructure is made more efficient without negatively impacting performance or availability. Leveraging TCP multiplexing is a simple method of optimizing connection utilization between clients and servers that can dramatically improve resource utilization and immediately increase capacity of **existing** "servers".

Organizations looking to improve their bottom line and do more with less ought to closely evaluate their application delivery strategy and find those places where resource utilization can be optimized as a way as to improve efficiency of the use of existing resources before embarking on a "throw more hardware at the problem" initiative.

- Long Live(d) AJAX
- Cloud Lets You Throw More Hardware at the Problem Faster
- WILS: Application Acceleration versus Optimization
- Two Different Sock(et)s
- What is server offload and why do I need it?
- 3 Really good reasons you should use TCP multiplexing
- SOA and Web 2.0: The Connection Management Challenge
- The Impact of the Network on AJAX
- The Impact of AJAX on the Network

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com