# Bare Metal Blog: Test for reality.

**Don MacVittie, 2012-30-11**

#BareMetalBlog #F5 Test results provided by vendors and "independent testing labs" often test for things that just don't matter in the datacenter. Know what you're getting.

When working in medicine, you don't glance over a patient, then ask them "so how do you feel when you're at your best?" You ask them what is wrong, then run a ton of tests – even if the patient thinks they know what is wrong – then let the evidence determine the best course of treatment.

Sadly, when determining the best tools for your IT staff to use, we rarely follow the same course. We invite a salesperson in, ask them "so, what do you do?", and let them tell us with their snippets of reality or almost-reality why their product rocks the known world. Depending upon the salesperson and the company, their personal moral code or corporate standards could limit them to not bringing up the weak points of their products to outright lying about its capabilities.

"But Don!", you say, "you're being a bit extreme, aren't you?" Not in my experience I am not. From being an enterprise architect to doing comparative reviews, I have seen it all. Vendor culture seems to infiltrate how employees interact with the world outside their HQ – or more likely (though I've never seen any research on it), vendors tend to hire to fit their culture, which ranges from straight-up truth about everything to wild claims that fall apart the first time the device is put into an actual production-level network.

The most common form of disinformation that is out there is to set up tests so they simply show the device operating at peak efficiency. This is viewed as almost normal by most vendors – why would you showcase your product in less than its best light? and as a necessary evil by most of the few who don't have that view – every other vendor in the space is using this particular test metric, we'd better too or we'll look bad. Historically, in network gear, nearly empty communications streams have been the standard for high connection rates, very large window sizes the standard for manipulating throughput rates. While there are many other games vendors in the space play to look better than they are, it is easy to claim you handle X million connections per second if those connections aren't actually doing anything. It is also easier to claim you handle a throughput of Y Mbps if you set the window size larger than reality would ever see it.

Problem with this kind of testing is that it seeps into the blood, after a while, those test results start to be sold as actual… And then customers put the device into their network, and needless to say, they see nothing of the kind. You would be surprised the number of times when we were testing for Network Computing that a vendor downright failed to operate as expected when put into a live network, let alone met the numbers the vendor was telling customers about performance.

One of the reasons I came to F5 way back when was that they did not play these games. They were willing to make the marketing match the product and put a new item on the roadmap for things that weren't as good as they wanted. We're still out there today helping IT staff understand testing, and what testing will show relevant numbers to the real world. By way of example, there is the Testing Configuration Guide on F5 DevCentral.

As Application Delivery Controllers have matured and solidified, there has been much change in how they approach network traffic. This has created an area we are now starting to talk more about, which is the validity of throughput testing in regards to ADCs in general. The thing is, we've progressed to the point that simply "we can handle X Mbps!" is no longer a valid indication of the workloads an ADC will be required to handle in production scenarios. The real measure for application throughput that matters is requests per second. Vendors generally avoid this kind of testing, because response is also limited by the capacity of the server doing the actual responding, so it is easy to get artificially low numbers.

At this point in the evolution of the network, we have reached the reality of that piece of utility computing. Your network should be like electricity. You should be able to expect that it will be on, and that you will have enough throughput to handle incoming load. Mbps is like measuring amperage… When you don't have enough, you'll know it, but you should, generally speaking, have enough. It is time to focus more on what uses you put that bandwidth to, and how to conserve it. Switching to LED bulbs is like putting in an ADC that is provably able to improve app performance. LEDs use less electricity, the ADC reduces bandwidth usage… Except that throughput or packets per second isn't measuring actual improvements of bandwidth usage. It's more akin to turning off your lights after installing LED bulbs, and then saying "lookie how much electricity those new bulbs saved!"

Seriously, do you care if your ADC can delivery 20 million Megabits per second in throughput, or that it allows your servers to respond to requests in a timely manner? Seriously, the purpose of an Application Delivery Controller is to facilitate and accelerate the delivery of applications, which means responses to requests. If you're implementing WAN Optimization functionality, throughput is still a very valid test. If you're looking at the Application Delivery portion of the ADC though, it really has no basis in reality, because requests and responses are messy, not "as large a string of ones as I can cram through here". From an application perspective – particularly from a web application perspective – there is a lot of "here's a ton of HTML, hold on, sending images, wait, I have a video lookup…" Mbps or MBps just doesn't measure the variety of most web applications. But users are going to feel requests per second just as much as testing will show positive or negative impacts. To cover the problem of application servers actually having a large impact on testing, do what you do with everything else in your environment, control for change. When evaluating ADCs, simply use the same application infrastructure and change only the ADC out. Then you are testing apples-to-apples, and the relative values of those test results will give you a gauge for how well a given ADC will perform in your environment.

In short, of course the ADC looks better if it isn't ***doing*** anything. But ADCs do a ton in production networks, and that needs to be reflected in testing. If you're fortunate enough to have time and equipment, get a test scheduled with your prospective vendor, and make sure that test is based upon the usage your actual network will expose the device to. If you are not, then identify your test scenarios to stress what's most important to you, and insist that your vendor give you test results in those scenarios. In the end, you know your network far better than they ever will, and you know they're at least not telling you the whole story, make sure you can get it.

Needless to say, this is a segue into the next segment of our #BareMetalBlog series, but first I'm going to finish educating myself about our use of FPGAs and finish that segment up.

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

| | | | |
|---|---|---|---|
| F5 Networks, Inc.<br>Corporate Headquarters<br>info@f5.com | F5 Networks<br>Asia-Pacific<br>apacinfo@f5.com | F5 Networks Ltd.<br>Europe/Middle-East/Africa<br>emeainfo@f5.com | F5 Networks<br>Japan K.K.<br>f5j-info@f5.com |