There is more to it than performance.

Don MacVittie, 2012-31-07

Did you ever notice that sometimes, “high efficiency” furnaces aren’t? That some things the furnace just doesn’t cover – like the quality of your ductwork, for example?

The same is true of a “high performance” race car. Yes, it is VERY fast, assuming a nice long flat surface for it to drive on. Put it on a dirt road in the rainy season, and, well, it’s just a gas hog. Or worse, a stuck car.

I could continue the list. A “high energy” employee can be relatively useless if they are assigned tasks at which brainpower, not activity rate, determines success…

But I’ll leave it at those three, I think you get the idea.

The same is true of your network. Frankly, increasing your bandwidth in many scenarios will not yield the results you expected. Oh, it will improve traffic flow, and overall the performance of apps on the network will improve, the question is “how much?”

It would be reasonable – or at least not unreasonable – to expect that doubling Internet bandwidth should stave off problems until you double bandwidth usage. But often times the problems are with the overloading apps we’re placing upon the network. Sometimes, it’s not the network at all.

Check the ecosystem, not just the network.

When I was the Storage and Servers Editor over at NWC, I reviewed a new (at the time) HP server that was *loaded*. It had a ton of memory, a ton of cores, and could make practically any application scream. It even had two gigabit NICs in it. But they weren’t enough. While I had almost nothing bad to say about the server itself, I did observe in the summary of my article that the network was now officially the bottleneck. Since the machine had high speed SAS disks, disk I/O was not as bi a deal as it traditionally has been, high-speed cached memory meant memory I/O wasn’t a problem at all, and multiple cores meant you could cram a ton of processing power in.

But team those two NICs and you’d end up with slightly less than 2 Gigabits of network throughput. Assuming 100% saturation, that was really less than 250 Megabytes per second, and that counts both in and out. For query-intensive database applications or media streaming servers, that just wasn’t keeping pace with the server.

Now here we are, six or so years later, and similar servers are in use all over the globe… Running VMs. Meaning that several copies of the OS are now carving up that throughput.

So start with your server. Check it first if the rest of the network is performing, it might just be the problem.

And while we’re talking about servers, the obvious one needs to be mentioned… Don’t forget to check CPU usage. You just might need a bigger server or load balancing, or these days, less virtuals on your server.

Heck, as long as we’re talking about servers, let’s consider the app too. The last few years for a variety of reasons we’ve seen less focus on apps whose performance is sucktacular, but it still happens. Worth looking into if the server turns out to be the bottleneck.

Old gear is old.

I was working on a network that deployed an ancient Cisco switch. The entire network was 1 Gig, except for that single switch. But tracing wires showed that switch to lie between the Internet and the internal network. A simple error, easily fixed, but an easy error to have in a complex environment, and certainly one to be aware of. That switch was 10/100 only. We pulled it out of the network entirely, and performance instantly improved.

There's necessary traffic, and then there's…

Not all of the traffic on your network needs to be. And all that does need to be doesn't have to be so bloated.

Look for sources of UDP broadcasts. More often than you would think, applications broadcast that you don't care about. Cut them off.

For other traffic, well there is Application Delivery Optimization. ADO is improving application delivery by a variety of technical solutions, but we'll focus on those that make your network and your apps ***seem*** faster. You already know about them – compression, caching, image optimization… In the case of back-end services, de-duplication. But have you considered what they do other than improve perceived or actual performance?

Free Bandwidth

Anything that reduces the size of application data leaving your network also reduces the burden on your Internet connection. This goes without saying, but as I alluded to above, we sometimes overlook the fact that it is not just application performance we're impacting, but the effectiveness of our connections – connections that grow more expensive by leaps and bounds each time we have to upgrade them.

While improving application performance is absolutely a valid reason to seek out ADO, delaying or eliminating the need to upgrade your Internet connection(s) is another. Indeed, in many organizations it is far easier to do TCO justification based upon deferred upgrades than it is based upon "our application will be faster", while both are benefits of ADO.

New stuff!

And as time wears on, SPDY, IPv6, and a host of other technologies will be more readily available to help you improve your network. Meanwhile, check out gateways for these protocols to make the transition easier.

In Summation

There are a lot of reasons for apps not to perform, and there are a lot of benefits to ADO. I've listed some of the easier problems to ferret out, the deeper into your particular network you get, the harder it is to generalize problems. But these should give you a taste for the types of things to look for. And a bit more motivation to explore ADO. Of course I hope you choose F5 gear for ADO ***and*** overall application networking, but there are other options out there. I think. Maybe. 😊