Load Balancing For Developers: Improving Application Performance With ADCs

Don MacVittie, 2010-08-10

If you've never heard of my Load Balancing For Developers series, it's a good idea to start here. There are quite a few installments behind us, and I'm not going to look back in this post any more than I must to make it readable without going back… Meaning there's much more detail ***back there*** than I'll relate here.

Again after a lengthy sojourn covering other points of interest, I return to Load Balancing For Developers with a more holistic view – application performance. Lori has talked a bit about this topic, and I've talked about it in the form of Load Balancing benefits and algorithms, but I'd like to look more architecturally again, and talk about those difficult to uncover performance issues that web apps often face.

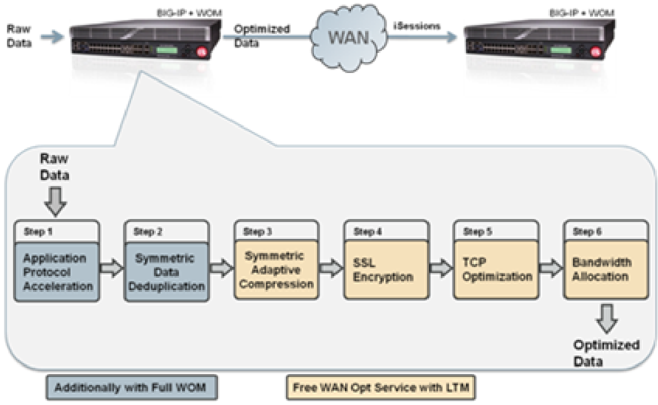
You're the IT manager for the company's Zap-n-Go website, it has grown nearly exponentially since launch, and you're the one responsible for keeping it alive. Lately it's online, but your users are complaining of sluggishness. Following the advice of ***some guy on the Internet***, you put a load balancer in about a year ago, and things were better, but after you put in a redundant data center and Global Load Balancing services, things started to degrade again. Time to rethink your architecture before your product gets known as Zap-N-Gone… Again.

Thus far you have a complete system with multiple servers behind an ADC in your primary data center, and a complete system with multiple servers behind an ADC in your secondary data center. Failover tests work correctly when you shut down the primary web servers, and the database at the remote location is kept up to date with something like Data Guard for Oracle or Merge Replication Services for SQL Server. This meets the business requirement that the remote database is up-to-date except for those transactions in-progress at the moment of loss. This makes you highly HA, and if your ADCs are running as an HA pair and your Global DNS – Like our GTM product - is smart enough to switch when it notices your primary site is down, most users won't even know they've been shoved off to the backup datacenter. The business is happy, you're sleeping at night, all is well.

Except that slowly, as usage for the site has grown, performance has suffered. What started as a slight lag has turned into a dragging sensation. You've put more web servers into the pool of available resources – or better yet, used your management tools (in the ADC and on your servers) to monitor all facets of web server performance – disk and network I/O, CPU and memory utilization. And still, performance lags.

Then you check on your WAN connection and database, and find the problem. Either the WAN connection is overloaded, or the database is waiting long periods of time for responses from the secondary datacenter. If you have things configured so that the primary doesn't wait for acknowledgment from the secondary database, then your problem might be even more sinister – some transactions may never get deposited in the secondary datacenter, causing your databases to be out of synch.

And that's a problem because you need the secondary database to be as up to date as possible, but buying more bandwidth is a monthly overhead expense, and sometimes it doesn't help – because the problem isn't always about bandwidth, sometimes it is about latency. In fact, with synchronous real-time replication, it is almost always about latency. Latency, for those who don't know, is a combination of how far your connection must travel over the wire and the number of "bumps in the wire" that have been inserted. Not actually the number of devices, but the number and their performance. Each device that touches your data – packet inspection, load balancing, security, whatever the reason – adds time to the delivery window. So does traveling over the wires/fiber. Synchronous replication is very time sensitive. If it doesn't hear back in time, it doesn't commit the changes, and then the primary and secondary databases don't match up.



So you need to cut down the latency and improve the performance of your WAN link. Conveniently, your ADC can help. Out-of-the-box it should have TCP optimizations that cut down the impact of latency by reducing the number of packets going back and forth over the wire. It may have compression too – which cuts down the amount of data going over the wire, reducing the number of packets required, which improves the “apparent” performance and the amount of data on your WAN connection. They might offer more functionality than that too. And you’ve already paid for an HA pair – putting one in each datacenter – so all you have to do is check what they do “out of the box” for WAN connections, and then call your sales representative to find out what other functionality is available. F5 includes some functionality in our LTM product, and has more in our add-on WAN Optimization Module (WOM) that can be bought and activated on your BIG-IP. Other vendors have a variety of architectures to offer you similar functionality, but of course I work for and write for F5, so my view is that they aren’t as good as our products… Certainly check with your incumbent vendor before looking for other solutions to this problem.

We have seen cases where replication was massively improved with WAN Optimization. More on that in the coming days under a different topic, but just the thought that you can increase the speed and reliability of transaction-based replication (and indeed, file/storage replication, but again, that’s another blog), and *you as a manager or a developer do not have to do a thing to your code.* That implies the other piece – that this method of improvement is applicable to applications that you have purchased and do not own the source code for.

So check it out… At worst you will lose a few hours tracking down your vendor’s options, at best you will be able to go back to sleep at night.

And if you’re shifting load between datacenters, as I’ve mentioned before, Long Distance vMotion is improved by these devices too. F5’s architecture for this solution is here – PDF deployment guide. This guide relies upon the WOM functionality mentioned above.

And encryption is supported between devices. That means if you are not encrypting your replication, that you can start without impacting performance, and if you are encrypting, you can offload the work of encryption to a device designed to handle it.

And bandwidth allocation means you can guarantee your replication has enough bandwidth to stay up to date by giving it priority.

But you won't care too much about that, you'll be relaxing and dreaming of beaches and stock options… Until the next emergency crops up anyway.

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