Advanced Load Balancing for Developers – ADCs, What’s the Difference?

Don MacVittie, 2009-07-07

So a while back I covered Load Balancing for Developers, trying to help developers who don't yet have exposure to load balancing to understand the when/where/how of load balancing.

I took a bit of a break to do some BIG-IP/TMOS V.10 work, and figure it's about time (since I've been gently prodded by readers a couple of times) to move on with the advanced applications.

But first, a moment of silence for Borland, who this time surely is breathing its last. I'll try not to reminisce too much here, but their DOS IDE was the best out there, hands down. And their compiler was better than most (I think TopSpeed beat it, but not by much).

When the term Application Delivery Controller (ADC) came about, most people didn't have a clue what the differentiators between simple load balancing and ADCs were, but time has passed, and people like Lori have set us all straight on what exactly it takes to be an ADC… And I'm going to glom off those super-intelligent individuals and give you a run-down of what you can hope to get from an ADC that you wouldn't get (or aren't getting) from your Load Balancer.

First, the easy part. An ADC *is* a load balancer, just a very advanced one. In a manner very similar to the way that C++ *evolved from* C, just with a bunch of pre-processor stuff as the early implementations, ADCs evolved from load balancers, just with a bunch of pre-, post-, and intra- processing stuff as the early implementations. So ADCs are really Load Balancers++. This means the same thing it does for C++… In short, if a load balancer can do it, so can an ADC. The opposite is not always true.

In 2007, Gartner did their Magic Quadrant for ADCs with a description of functionality they looked for. It's a great starting place, so we'll use it.

For the rest of these, you might find one or two of them implemented in your load balancer, but you can expect all or nearly all of them in your ADC.

Gartners' list of ADC functions is large, and the list keeps growing, but there are a few general groups you can apply. We'll just mention them here, and cover them in future installments. (side note: Gartner's list actually shrank in 2008, but is more general and less redundant)

Compression and Optimization – server offload, data compression, connection multiplexing, Caching, content pre-fetching, protocol-specific optimizations, etc.

Security and NAT – Network Address Translation, SQL verification, Application Firewall, DOS/DDOS protection, fingerprint masquerading, application proxy, etc.

Transformation – NAT falls into this category for some implementations, redirection of URLs does also, but transforming the payload is a complex and effective version of this one also.

Did I miss anything? Oh yeah, most important one to me…

Programmability – scripting or API interfaces that allow you to manage the load balancer at a granular level (eg: more than On/Off) dynamically. This is big in automating the data center – bringing up another instance is only helpful if it all happens when needed and goes away when not.

We'll take a look at these over the next few weeks, how you can use them to improve your site's performance, reliability, and security. All while load balancing, of course… Though you don't *have* to load-balance to utilize and ADC, since they do a ton more that load balancing these days.

Until then,

Don.