

Advanced Load Balancing For Developers. The Network Dev Tool



Don MacVittie, 2012-03-02

It has been a while since I wrote an installment of Load Balancing for Developers, and now I think it has been too long, but never fear, this is the grad-daddy of Load Balancing for Developers blogs, covering a useful bit of information about Application Delivery Controllers that you might want to take advantage of. For those who have joined us since my last installment, feel free to check out the entire list of blog entries (along with related blog entries) [here](#), though I assure you that this installment, like most of the others, does not require you to have read those that went before.



ZapNGo! Is still a growing enterprise, now with several dozen complex applications and a high availability architecture that spans datacenters and the cloud. While the organization relies upon its web properties to generate revenue, those properties have been going along fine with your Application Delivery Controller (ADC) architecture.

Now though, you're seeing a need to centralize administration of a whole lot of functions. What worked fine separately for one or two applications is no longer working so well now that you have several development teams and several dozen applications, and you need to find a way to bring the growing inter-relationships under control before maintenance and hidden dependencies swamp you in a cascading mess of disruption.

With maintenance taking a growing portion of your application development manhours, and a reasonably well positioned test environment configured with a virtual ADC to [mimic your production environment](#), all you need now is a way to cut those maintenance manhours and reduce the amount of repetitive work required to create or update an application. Particularly update an application, because that is a constant problem, where creating is less frequent.

With many of the threats that your ZapNGo application will be known as ZapNGone eliminated, now it is efficiencies you are after. And believe it or not, these too are available in an ADC. Not all ADC's are created equal, but this discussion will stay on topics that most ADCs can handle, and I'll mention it when I stray from generic into specific – which I will do in one case because only one vendor supports one of the tools you can use, but all of the others should be supported by whatever ADC vendor you have, though as always, check with your vendor directly first, since I'm not an expert in the inner workings of every one.

There is a lot that many organizations do for themselves, and the array of possibilities is long – from implementing load balancing in source code to security checks in the application, the boundaries of what is expected of developers are shaped by an organization, its history, and its chosen future direction. At ZapNGo, the team has implemented a virtual test environment that as close as possible mirrors production, so that code can be implemented and tested in the way it will be used. They use an ADC for load balancing, so that they don't have to rewrite the same code over and over, and they have a policy of utilizing a familiar subset of ADC functionality on all applications that face the public.

The company is successful and growing, but as always happens in companies in that situation, the pressures upon them are changing just by virtue of their growth. There are more new people who don't yet have intimate knowledge of the code base, network topology, security policies, whatever their area of expertise is. There are more lines of code to maintain, while new projects are being brought up at a more rapid pace and with higher priorities (I've twice lived through the "Everything is high priority? Well this is highest priority!" syndrome while working in IT. Thankfully, most companies grow out of that fast when it's pointed out that if everything is priority #1, nothing is). Timelines to complete projects – be they new development, bug fixes, or enhancements are stretching longer and longer as the percentage of gurus in the company is down and the complexity of the code and the architecture it runs on is up.

So what is a development manager to do to increase productivity? Teaming newer developers with people who've been around since the beginning is helping, but those seasoned developers are a smaller and smaller percentage of the workforce, while the volume of work has slowly removed them from some of the many products now under management. Adopting coding standards and standardized libraries helps increase experience portability between projects, but doesn't do enough.

Enter offloading to the ADC. Some things just don't *have* to be done in code, and if they don't *have* to be, at this stage in the company's growth, IT management at ZapNGo (that's you!) decides they won't be. There just isn't time for non-essential development anymore.

Utilizing a policy management tool and/or an [Application Firewall](#) on the ADC can improve security without increasing the code base, for example. And that shaves hours off of maintenance projects, while standardizing on one or a few implementations that are simply selected on the ADC. Implementing [Web Application Acceleration](#) protocols on the ADC means that less in-code optimization has to occur. Performance is no longer purely the role of developers (but of course it is still a concern. No Web Application Acceleration tool can make a loop that runs for five minutes run faster), they can allow the Web Application Acceleration tool to shrink the amount of data being sent to the users' browser for you. Utilizing a [WAN Optimization](#) ADC tool to improve the performance of bulk copies or backups to a remote datacenter or cloud storage... The list goes on and on.

The key is that the ADC enables a lot of opportunities for App Dev to be more responsive to the needs of the organization by moving repetitive tasks to the ADC and standardizing them. And a heaping bonus is that it also does that for operations with a different subset of functionality, meaning one toolset gives both App Dev and Operations a bit more time out of their day for servicing important organizational needs. Some would say this is all part of DevOps, some would say it is not. I leave those discussions to others, all I care is that it can make your apps more secure, fast, and available, while cutting down on workload.

And if your ADC supports an SSL VPN, your developers can work from home when necessary. Or more likely, if your code is your IP, a subset of your developers can. Making ZapNGo more responsive, easier to maintain, and more adaptable to the changes coming next week/month/year. That's what ADCs do. And they're pretty darned good at it.

That brings us to the one bit that I have to caveat with *F5 only*, and that is iApps. An iApp is a constructed configuration tool that asks a few questions and then deploys all the bits necessary to set up an ADC for a particular application. Why do I mention it here? Well if you have dozens of applications with similar characteristics, you can create an iApp Template and use it to rapidly bring new applications or new instances of applications online. And since it is abstracted, these iApp templates can be designed such that AppDev, or even the business owner, is able to operate them



Meaning less time worrying about what network resources will be available, how they're configured, and waiting for operations to have time to implement them (in an advanced ADC that is being utilized to its maximum in a complex application environment, this can be hundreds of networking objects to configure – all encapsulated into a form). Less time on the project timeline, more time for the next project. Or for the post deployment party. One of the two. That's it for the *F5 only* bit.

And knowing that all of these items are standardized means less things to get mis-configured, more surety that it will all work right the first time. As with all of these articles, that offers you the most important benefit... A good night's sleep.

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