

Project Acceleration: Test Infrastructure



Jason Rahm, 2013-04-04

Note: As of 11.4, WebAccelerator is now a part of BIG-IP Application Acceleration Manager.

This is article eight of ten in a series on DevCentral's implementation of WebAccelerator. Join Colin Walker and product manager Dawn Parzych as they discuss the ins and outs of WebAccelerator. Colin discusses his take on implementing the technology first hand (with an appearance each from Jason Rahm and Joe Pruitt) while Dawn provides industry insight and commentary on the need for various optimization features.

- [Part 1 - The Basics](#)
- [Part 2 - TCP Optimization & Compression](#)
- [Part 3 - WebAccelerator Building Blocks](#)
- [Part 4 - Intelligent Browser Referencing](#)
- [Part 5 - Image Optimization](#)
- [Part 6 - Content Reordering](#)
- [Part 7 - Tips, Tricks, & Gotchas](#)
- [Part 8 - Test Infrastructure](#)
- [Part 9 - Programmatic Performance Testing with HTTPWatch](#)
- [Part 10 - Conclusions](#)

Objectives

We had a policy in place for WebAccelerator, but frequently had issues of the 'ghost in the machine' type, which were probably the result of carrying forward a policy that had not been analyzed or tweaked much from the very early 9.4.x deployment. So when Dawn approached us about working together to get some real world WA stats for use in presentations, we saw it as an opportunity to get a great performing policy in return for being the guinea pigs. However, the thought of hunt and peck policy changes for a world-wide audience of testers at all hours of the day didn't much sound appealing, so we sought to provide an environment with no local changes necessary and as little change on the user end as possible. We did have some road blocks, though.

Dawn Says...



My motto has always been to test one thing at a time to see the impact, if you change multiple parameters at once and something goes wrong it is much harder to troubleshoot. However with WebAccelerator I have never turned on one feature at a time, I've always just gone to one of the pre-defined policies and turned it on. We talk a lot about how some features only impact first time visits and others repeat visits, or that the feature is designed to reduce bandwidth versus the page load time but I've never had the data. When I approached the DevCentral team about creating an environment where we could look at the impact of one feature at a time they were excited. Of course we would then want to measure the effectiveness of these features from real world users, luckily at F5 there was a willing pool of global acceleration experts that we could tap to determine the impact that latency and bandwidth has the various features. I have to say I was a little surprised by the results that we obtained. We will cover those in a later video.

Road Block #1

The biggest road block was IP space. This needed to be publicly available, but we don't have an endless or even a small amount of freely available IPs in our environment, so we had to be creative on how to deploy multiple policies behind a single IP. The solution to this road block was to use the vip targeting vip solution and have a front-end vip that mapped traffic to a variety of back-end vips, each with its own WA policy. However, we ended up with two IPs instead of one because you can't switch a TCP profile, it's either on or off, so the baseline solution is on a separate public vip than the rest of the solution.

Road Block #2

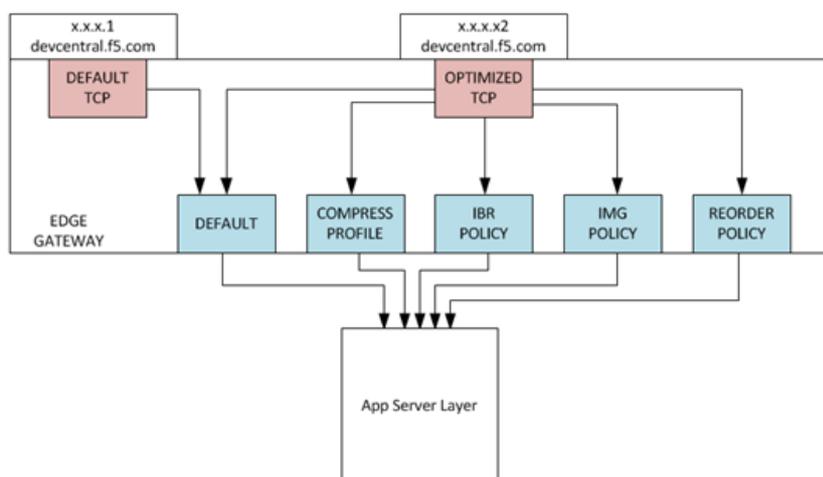
The initial ideas on policy switching centered around using a plugin like Modify Headers in firefox, and whereas it is possible to change and even insert headers in all the major browsers, we faced two issues with this. One, not all browsers persist these headers through a close and re-open scenario, and we needed this for collecting accurate first and repeat visits. Second, I ignored the fact that some of our testers would be coming from mobile devices that likely would not have this level of control. The solution we landed on utilized cookies instead of headers.

Road Block #3

Originally we had plans to test SPDY as well in this effort, but in the DevCentral environment WebAccelerator is deployed on an Edge Gateway, and since SPDY is an EA feature, it can only be licensed on a platform with an LTM base license, which Edge Gateway does not have. When SPDY goes GA, we'll have the opportunity to test that as well.

Configuration Overview

Now that all the roadblocks had been identified, we could move on to the design and implementation phases. The overview of the plan in diagram form:



Essentially, we ended up with two front-end virtual servers and five on the back-end.

Front-End Virtuals

The front-end virtuals are the client-connection and SSL termination points. As discussed in the roadblocks, the only reason the Default TCP virtual exists is for the baseline since we cannot switch the tcp profiles on a single virtual server in flight. It has no compression or caching and as implied, a default TCP stack. It also has a single iRule applied to send all requests to the Default back-end virtual server, which is just a pass-through. This is done simply to keep all the paths (from vertical perspective) the same for comparisons.

```
when HTTP_REQUEST {  
  virtual /pm_wa/v.dc.wa_default_0  
}
```

The Optimized TCP front-end virtual has a custom TCP stack and an iRule that performs a few functions:

1. Inserts a header based-on URL requested
2. Switches to the appropriate back-end policy based on included header
3. Enables/Disables compression based on policy path

```
when HTTP_REQUEST {
  ## Insert Cookies for policy switching
  set setcookie ""
  switch [string tolower [HTTP::uri]] {
    "/none" {
      set setcookie " \Set-Cookie\ \X-WA-Policy=none; Expires=Thu, 01 Jan 1970 00:00:01 GMT\""
    }
    "/tcp" {
      set setcookie " \Set-Cookie\ \X-WA-Policy=tcp\""
    }
    "/compress" {
      set setcookie " \Set-Cookie\ \X-WA-Policy=compress\""
    }
    "/ibr" {
      set setcookie " \Set-Cookie\ \X-WA-Policy=ibr\""
    }
    "/img" {
      set setcookie " \Set-Cookie\ \X-WA-Policy=img\""
    }
    "/reorder" {
      set setcookie " \Set-Cookie\ \X-WA-Policy=reorder\""
    }
    "/spdy" {
      set setcookie " \Set-Cookie\ \X-WA-Policy=spdy\""
    }
  }
  if { [string length $setcookie] > 0 } {
    HTTP::uri "/"
    set cmd "HTTP::respond 302 Location \https://devcentral.f5.com[HTTP::uri]\ $setcookie"
    eval $cmd
    return
  }
  if { [string tolower [HTTP::uri]] eq "/current" } {
    HTTP::respond 200 content "Current cookie setting for X-WA-Policy is: [HTTP::cookie "X-WA-Policy"
  }

  set vip "/pm_wa/v.dc.wa_default_0"
  switch [string tolower [HTTP::cookie "X-WA-Policy"]] {
    "tcp" {
      set vip "/pm_wa/v.dc.wa_default_0"
      COMPRESS::disable
    }
    "compress" {
      set vip "/pm_wa/v.dc.wa_compress_1"
      COMPRESS::enable
    }
    "ibr" {
      set vip "/pm_wa/v.dc.wa_ibr_2"
      COMPRESS::enable
    }
  }
}
```

```
    COMPRESS::enable
  }
  "img" {
    set vip "/pm_wa/v.dc.wa_img_3"
    COMPRESS::enable
  }
  "reorder" {
    set vip "/pm_wa/v.dc.wa_reorder_4"
    COMPRESS::enable
  }
  "spdy" {
    set vip "/pm_wa/v.dc.wa_spdy_5"
    COMPRESS::enable
  }
  default {
    set vip "/pm_wa/v.dc.wa_default_0"
    COMPRESS::disable
  }
}
virtual $vip

when HTTP_RESPONSE {
  HTTP::header insert "X-DC-Virtual" $vip
}
```

We left the SPDY path in the iRules for future use, but there is no actionable WA policy attached currently.

Back-End Virtuals

The back-end virtual servers only perform two functions: apply the appropriate WA policy and send traffic on down to the application stack.

Administrative Access

Given that we were deploying this infrastructure in our production environment, we were leery to give unfettered access to individuals to the entire platform. We utilized partitions and created roles within these partitions to allow Dawn and team to get in and configure and test in the pm_wa partition without opening up access to the remainder of the platform. This worked well for all parties, everyone was able to do their thing without compromising our other applications.

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