v.10 - FastHTTP and Cookie Persistence

Joe Pruitt, 2009-15-04

With BIG-IP version 10, the FastHTTP profile has been redesigned to allow for a more apples-to-apples feature set with the HTTP profile. This article will discuss the FastHTTP profile and what has been added in v10.

Background

The Fast HTTP profile is a configuration tool designed to speed up certain types of HTTP connections. This profile combines selected features from the TCP, HTTP, and OneConnect profiles into a single profile that is optimized for the best possible network performance. When you associate this profile with a virtual server, the virtual server processes traffic packet-by-packet, and at a significantly higher speed.

You might consider using a FastHTTP profile when:

- You do not need features such as remote server authentication, SSL traffic management, and TCP optimizations, nor HTTP features such as data compression, pipelining, and RAM Cache.
- You do not need to maintain source IP addresses.
- You want to reduce the number of connections that are opened to the destination servers.
- The destination servers support connection persistence, that is, HTTP/1.1, or HTTP/1.0 with Keep-Alive headers. Note that IIS servers support connection persistence by default.
- You need basic iRule support only (such as limited Layer 4 support and limited HTTP header operations). For example, you can use the iRule events CLIENT_ACCEPTED, SERVER_CONNECTED, and HTTP_REQUEST.

A significant benefit of using a Fast HTTP profile is in the way in which the profile supports connection persistence. Using a Fast HTTP profile ensures that for client requests, the BIG-IP system can transform or add an HTTP Connection header to keep connections open. Using the profile also ensures that the BIG-IP system pools any open server-side connections. This support for connection persistence can greatly reduce the load on destination servers by removing much of the overhead caused by the opening and closing of connections.

The Fast HTTP profile sits in the middle ground between the Fast L4 and the more feature rich content aware HTTP profile.

Profile Settings

To get an idea on what the Fast HTTP profile can do, it will help to take a look at the different profile settings. Below are a list of the different configuration settings in the Fast HTTP profile.

General Settings

- *Reset on Timeout* - Specifies, when checked (enabled), that the system sends a TCP RESET packet when a connection times out, and deletes the connection.
- *Idle Timeout* - Specifies the number of seconds after which a connection is eligible for deletion, when the connection has no traffic.
- *Maximum Segment Size Override* - Specifies a maximum segment size (MSS) override for server-side connections. The default is 0, which corresponds to an MSS of 1460. You can specify any integer between 536 and 1460.
- *Client Close Timeout* - Specifies the number of seconds after which the system closes a client connection, when the system either receives a client FIN packet or sends a FIN packet to the client. This setting overrides the Idle Timeout setting. The default is 5 seconds.
- *Server Close Timeout* - Specifies the number of seconds after which the system closes a client connection, when the system either receives a server FIN packet or sends a FIN packet to the server. This setting overrides the Idle Timeout setting. The default is 5 seconds.
- *Unlcean Shutdown* - Specifies how the system handles closing connections.
- *Force HTTP 1.0 Response* - Specifies, when checked (enabled), that the server sends responses to clients in the

- *Force HTTP 1.0 Response* - Specifies, when checked (enabled), that the server sends responses to clients in the HTTP/1.0 format. This effectively turns off client chunking and pipelining. The default is disabled.

OneConnect

- *Maximum Pool Size* - Specifies the maximum number of connections to a load balancing pool. A setting of **0** specifies that a pool can accept an unlimited number of connections. The default is **2048**.
- *Minimum Pool Size* - Specifies the minimum number of connections to a load balancing pool. A setting of **0** specifies that there is no minimum. The default is **0**.
- *Ramp-Up Increment* - Specifies the increment in which the system makes additional connections available, when all available connections are in use. The default is **4**.
- *Maximum Reuse* - Specifies the maximum number of times that the system can re-use a current connection. The default is **0**.
- *Idle Timeout Override* - Specifies the number of seconds after which a server-side connection in a OneConnect pool is eligible for deletion, when the connection has no traffic. This setting overrides the **Idle Timeout** that you specify in the Settings section of the Fast HTTP profile. The default is setting **Disabled**.
- *Replenish* - Specifies, when enabled, that the system will not keep a steady-state maximum of connections to the back-end unless the number of connections to the pool have dropped beneath the Connection Pool Minimum specified in the profile.

HTTP

- *Parse Requests* - Specifies, when checked (enabled), that the system parses the HTTP data in the connection stream. The default is enabled. Note that if you are using a Fast HTTP profile for non-HTTP traffic, you should disable this setting to shield against dynamic denial-of-service (DDOS) attacks.
- *Maximum Header Size* - Specifies the maximum amount of HTTP header data that the system buffers before making a load balancing decision. The default is **32768**.
- *Maximum Requests* - Specifies the maximum number of requests that the system can receive on a client-side connection, before the system closes the connection. A setting of **0** specifies that requests are not limited. The default is **0**.
- *Insert X-Forwarded For* - Specifies whether the system inserts the **XForwarded For:** header in an HTTP request with the client IP address, to use with connection pooling.
- *Request Header Insert* - Specifies a string that the system inserts as a header in an HTTP request. If the header exists already, the system does not replace it.

v.10 Enhancements

In version 10, the code was optimized but there were no GUI level features added to the profile's configuration. But, the exciting news is that the Fast HTTP profile is now compatible with the Cookie Persistence Profile. Previously, if you opted to use Fast HTTP on your virtual server, you were not allowed to associate any other profiles with it. This is no longer true as it is now compatible with the Cookie Persistence profile. Cookie persistence uses a HTTP cookie stored on a client's computer to allow the client to reconnect to the same pool member previously visisted at a website. There are four methods of cookie persistence including HTTP Cookie Insert, HTTP Cookie Rewrite, HTTP Cookie Passive and Cookie Hash. See the "Configuration Guide for BIG-IP Local Traffic Management" for more information about the Configuration settings for the Cookie persistence profile.

If you wish to use cookie persistence with the Fast HTTP profile, all you need to do is to assign both profiles as resources to your Virtual Server configuration.

The Fast HTTP Profile is not for everyone but if you are in a situation where transaction performance is all that matters and you need to get at a bit of the layer 7 HTTP content, then the Fast HTTP profile may be just for you.

Get the Flash Player to see this player.
20090415-v10_FastHTTPProfile.mp3

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