



IBM Software Group

# WebSphere Plug-in Session Affinity and Load Balancing

Bob Richter ([brichter@us.ibm.com](mailto:brichter@us.ibm.com))  
WebSphere L2 support  
15 October 2013



WebSphere® Support Technical Exchange



# Agenda

- Plug-in Session Affinity
- Plug-in LoadBalancing
- Analysis of Plugin Request Distribution through Plugin log

# Plug-in Session Affinity

- Session Affinity
  - ▶ SessionID Cookie
    - Uses Cookie header
  - ▶ URL Rewriting
    - URL rewriting requires the developer to use special encoding APIs
    - URL rewriting works by storing the session identifier in the page returned to the user.

## Plug-in Session Affinity

- **Session Affinity** allows returning requests to be routed back to the same, if that same server is available.
- Application Server Session cookie is enabled and set through WebSphere Administration console
  - ▶ Application servers -> <Application ServerName> -> Session management -> Cookies
- Default name for the Application Server session cookie is **JSESSIONID**
- Plug-in Session Affinity is handled by the WebSphere Plug-in through a **CloneID** which is contained in a the JSESSIONID cookie enabled and configured by the Application Server

# Plug-in Properties

- Plug-in properties in plugin-cfg.xml used for routing WebSphere servlet requests:
  - ▶ Route
  - ▶ VirtualHost is contained in a VirtualHostGroup
  - ▶ URI is contained in a URIGroup
  - ▶ Transport is contained in a Server
  - ▶ **CloneID** is contained in a Server

## Plug-in - CloneID

- CloneID Generate to Plugin-cfg.xml
- Generate as part of the JSESSIONID cookie
  - ▶ By default , automatically set
  - ▶ You MAY manually set the CloneID using Session Management Custom property
    - **HttpSessionCloneID**
      - See links at the end of presentation

## Plug-in Session Affinity

- JSESSIONID cookie contains
  - ▶ CacheID
  - ▶ SessionID
  - ▶ CloneID
- Only CloneID is used by WebSphere Plug-in for Session Affinity

## Plug-in Session Affinity

- Virtual Host and URI are used to determine if a valid Route for the request exists in the plugin-cfg.xml.
- If JSESSIONID Cookie exists
  - ▶ CloneID parsed from JSESSIONID
  - ▶ CloneID will override any route based on Virtual Host or URI mapping.



## Plug-in Session Affinity

- Plugn-cfg.xml CLONEID

<ServerCluster >

```
<Server CloneID="15d2hi0gn" ConnectTimeout="0"  
ExtendedHandshake="false" LoadBalanceWeight="2"  
MaxConnections="-1" Name="rjrNode07_server1"  
ServerIOTimeout="120"  
WaitForContinue="false">/Server>
```

```
<Server CloneID="15d2hi3ic" ConnectTimeout="0"  
ExtendedHandshake="false" LoadBalanceWeight="2"  
MaxConnections="-1" Name="rjrNode07_server2"  
ServerIOTimeout="120" WaitForContinue="false">
```

## Plug-in Session Affinity

- Plugn-cfg.xml JSESSIONID Affinity Cookie in URIGroup

```
<UriGroup Name="default_host_cluster1_URIs">  
  <Uri AffinityCookie="JSESSIONID"  
    AffinityURLIdentifier="jsessionid" Name="/snoop/*"/>  
</UriGroup>
```

## Plug-in Session Affinity (Set-cookie)

- Plug-in log entry of an Application Server response with JSESSIONID containing cloneID
  - ▶ [Fri Sep 24 14:59:45 2010] 00002d30 00002298 - DETAIL: HTTP/1.1 200 OK
  - ▶ [Fri Sep 24 14:59:45 2010] 00002d30 00002298 - DETAIL: **Set-Cookie: JSESSIONID=0000A0-ItRd37WYeiLGHKH\_kcFp:15d2hi3ic**; Path=/  
**15d2hi3ic**
- CloneID is set on the response from Application Server
  - ▶ Login with **NO** Session Affnity cookie(JSESSIONID)
  - ▶ Request with expired Session Affnity cookie.
- Once CloneID (**15d2hi3ic**) is set in JSESSIONID Cookie then affinity to this particular server will be observed by the plug-in routing.

## Plug-in Session Affinity (compare CloneID)

- CLONEID is parsed from JSESSIONID cookie and compared to the CLONEID in the Plugin-cfg.xml
- Plug-in “Trace” level log entries compare request CloneID to the Application Server CloneID in Plugin-cfg.xml
  - ▶ [Fri Sep 24 14:59:56 2010] 00002d30 00002298 - TRACE:  
ws\_server\_group: serverGroupFindClone: Comparing  
curCloneID '**15d2hi3ic**' to server clone id '**15d2hi0gn**'
  - ▶ [Fri Sep 24 14:59:56 2010] 00002d30 00002298 - TRACE:  
ws\_server\_group: serverGroupFindClone: Comparing  
curCloneID '**15d2hi3ic**' to server clone id '**15d2hi3ic**'

## Plug-in Session Affinity (Match CloneID)

- Match CloneID in JSESSIONID cookie to plugin-cfg.xml CLONEID and Application Server is selected
  - ▶ Server is selected base on CloneID match
  - ▶ [Fri Sep 24 14:59:56 2010] 00002d30 00002298 - TRACE: ws\_server\_group: serverGroupFindClone: **Match for clone 'rjrNode07\_server2'**
  - ▶ Application Server rjrNode07\_server2 is selected

## Plug-in Session Affinity (JSESSIONID Header)

- Plug-in “detail” log level entry, POST request and JSESSIONID cookie with CLONEID.
  - ▶ [Fri Sep 24 14:59:56 2010] 00002d30 00002298 -  
DETAIL: **POST**  
/PlantsByWebSphere/servlet/AccountServlet?action=login&updating=false HTTP/1.1
  - ▶ [Fri Sep 24 14:59:56 2010] 00002d30 00002298 -  
DETAIL: **Cookie: JSESSIONID =0000A0-ItRd37WYeiLGHKH\_kcFp:15d2hi3ic**

## Plug-in Session Affinity

- SessionID will expiration based on Session Timeout
- CloneID does NOT Expire
- Plug-in does not check SessionID for expiration
  - ▶ Plugin Only checks for CloneID to match to Application Server
- CLONEID is not reset when Application Server Sets SESSIONID in the JSESSIONID cookie
- This can affect Plugin Load Balancing requests
  - ▶ Requests with expired JSESSIONID will be routed as an Affinity Request rather than handled as a NEW request
  - ▶ Periodic Manual SET of the CLONEID can prevent this scenario

# Plug-in Load Balancing

- Round Robin
  - ▶ Distribute work across cluster members based on configured LoadBalanceWeights
  - ▶ Round Robin works best with Web servers that have a single process.
- Random
  - ▶ Distribute work across cluster member by random assignment
  - ▶ Recommended for Web Servers with Multi-process environments
- Plug-in Load Balancing is only within each cluster



## Plug-in Load Balancing (Round Robin only)

### ▪ **LoadBalanceWeight**

- ▶ Static weights initially configured by Administrator in the plugin-cfg.xml
- ▶ WebSphere Administration Console
  - Servers -> Clusters-> cluster Members
- ▶ Weight associated with each server in cluster
  - Number between 0 to 100
  - LoadBalanceWeight="8"

# Plug-in Load Balancing (Round Robin only)

- **Internal Router Weights**

- ▶ Based on initial LoadBalanceWeights
- ▶ Dynamically updated based Plug-in Algorithm

- **Sticky and Non-Sticky requests -> Session Affinity**

- ▶ Applies to both Round Robin and Random
- ▶ Sticky request
  - Session Affinity is established and request needs to be route to affinity Server
- ▶ Non-Sticky Request
  - Session Affinity is NOT established
  - New request

## Plug-in Load Balancing (Round Robin only)

- Initial Internal Router Tables Weights are set by dividing LoadBalanceWeights by the Greatest Common Divisor (GCD)
- As each request is routed to a cluster member (Application Server)
  - ▶ Internal router table weight of the application server gets decremented by 1
  - ▶ Decrement by 1 for ALL requests (affinity and non-affinity)
- Non-sticky requests are not routed to any cluster member whose present Internal Router Table Weight is  $\leq 0$ .
- Sticky request are routed to a cluster member whose Internal Router Table Weight is  $\leq 0$ 
  - ▶ Internal Router Table weights can be a negative value (-).

## Plug-in Load Balancing (Round Robin only)

- When the Internal Router Table Weights of **all** the cluster members are  $\leq 0$ 
  - ▶ Plug-in component **resets** ALL cluster members Internal Router Table Weights (when **all** weights are  $\leq 0$ )
  - ▶ Plug-in resetting of Internal Router Table Weights may **not** take the Internal Table Router Weights to their original starting values!

## Plug-in Load Balancing (Round Robin only)

- Algorithm is used to recalculate the Internal Router Table Weights
- Find minimum number “m” such that equation is true for all Application Servers
  - ▶  $(w + m * s) > 0$ 
    - w = current Internal Router Table Weight
    - s = starting Internal Router Table Weight
- Details at:
  - ▶ <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg21219567>

# Plug-in Load Balancing - Example

- Round Robin LoadBalanceWeight example using 3 servers in cluster (server1, server2 , server3)
  - ▶ **Configured Weights**
    - Server1 (8)
    - Server2 (6)
    - Server3 (18)
  - ▶ **Router Table Weights (4,3,9)**
    - Server1 (4)
    - Server2 (3)
    - Server3 (9)
  - ▶ GCD=2
- The Plug-in routes requests over a period of time but based on affinity, resources and other factors the **current Internal Router Table Weights** are:
  - ▶ Server1 (-20) - Negative because we continued to decrement for each request after "0"
  - ▶ Server2 ( -40) - Negative because we continued to decrement for each request after "0"
  - ▶ Server3 (0)
- Note: Based on the current Internal Route Table Weights, we can see that we routed more requests to server1 and server2 than server3

## Plug-in Load Balancing - Example

- Plug-in Algorithm ( $w + m * s > 0$ )
  - ▶ Find minimal number “m”
  - ▶ w is the current Internal Router Table Weight (-20,-40,0)
  - ▶ S is the starting Internal Router Table Weight (4,3,9)
  - ▶ M=14
- **NEW Internal Router Table Weights** are recalculated using the algorithm
  - ▶ Server 1 ( $-20 + 14 * 4$ ) = **36**
  - ▶ Server2 ( $-40 + 14 * 3$ ) = **2**
  - ▶ Server 3 ( $0 + 14 * 9$ ) = **126**

## Plug-in Load Balancing - Summary

- Example Summary
  - ▶ Initial Internal Router Table Weight was (4, 3,9)
  - ▶ Actual distributions shows (-20 , -40 , 0)
    - This means the most number of request went to
      - **Server1** -20 (+ 4 initial Router Weight) = **24 requests**)
      - **Server2** -40 (+3 Initial Router Weight) = **43 requests**)
      - **Server3** 0 (+9 Initial Router Weight) = **9 requests**)
- To adjust for the unbalanced actual requests new weigths:
  - ▶ Server 3 - new Weight 126
  - ▶ Server 1 - new Weight 36
  - ▶ Server2 - New Weight 2



# Plug-in Log Stats Entry

- Monitor Plug-in Load Balancing by enabling plug-in loglevel="Stats"
- Key information in Plug-in Stat Log Entry
  - ▶ **PendingRequest**
    - Actual real time pending requests. If this number grows you may have some type of delay in app Server this number will go up and Down and in good environment may always be ZERO.
  - ▶ **FailedRequest**
    - Cumulative failed requests for Application Server in specific Web Server Process.

## Plug-in Log Stats Entry

- Key information in Plug-in Stat Log Entry (Cont'd)
  - ▶ **AffinityRequest**
    - Session Affinity request with JSESSIONID rather than NEW or NON-STICKY requests
  - ▶ **TotalRequests**
    - Total Requests
- Stat Log Entry's after each request is handled
- We only need the **last** STATS log entry for **each Application Server** for **each Web Server Process** for review

# Plug-in Log Stats Entry

- **Process 000000b0** - Server1 and Server2
  - ▶ [Tue Sep 10 10:52:26 2013] 000000b0 00000304 - STATS: ws\_server: serverSetFailoverStatus: Server **server1** : pendingRequests 0 failedRequests 0 **affinityRequests 632 totalRequests 637**
  - ▶ [Tue Sep 10 10:46:40 2013] 000000b0 00000506 - STATS: ws\_server: serverSetFailoverStatus: Server **server2** : pendingRequests 0 failedRequests 0 **affinityRequests 842 totalRequests 846**
- **Process 000000f8** - Server1 and Server2
  - ▶ [Tue Sep 10 10:52:39 2013] 000000f8 0000191a - STATS: ws\_server: serverSetFailoverStatus: Server **server1** : pendingRequests 0 failedRequests 0 **affinityRequests 666 totalRequests 671**
  - ▶ [Tue Sep 10 10:46:18 2013] 000000f8 00000708 - STATS: ws\_server: serverSetFailoverStatus: Server **server2** : pendingRequests 0 failedRequests 0 **affinityRequests 872 totalRequests 877**

# Plug-in Log Stats Entry

- Process **000000b0**
  - ▶ Server1
    - **totalRequests 637** – **affinityRequests 632**
    - **New Requests = 5**
  - ▶ Server2
    - **totalRequests 846** - **affinityRequests 842**
    - **New Requests = 4**
- Process **000000f8**
  - ▶ Server1
    - **totalRequests 671** – **affinityRequests 666**
    - **NEW Requests = 5**
  - ▶ Server2
    - **totalRequests 877** - **affinityRequests 872**
    - **New Requests = 5**

# Plug-in Load Balancing

- **Taking Application Server in Cluster Gracefully Offline**

- ▶ When LoadBalanceWeight is set to zero, this is a signal to the Plug-in to stop sending **new** requests to that Application Server
- ▶ This has the effect of taking an Application Server in cluster gracefully offline.
- ▶ Sticky requests on that server will continue
  - As Sessions get terminated (because the user explicitly logs out or because the session idle timeout is triggered), the server, in time, will no longer have any active sessions.

- See this link for details:

- ▶ [http://www.ibm.com/developerworks/websphere/techjournal/0412\\_vansickel/0412\\_vansickel.html](http://www.ibm.com/developerworks/websphere/techjournal/0412_vansickel/0412_vansickel.html)

# Plug-in Load Property

- Plug-in Property **MaxConnections**

- ▶ Represent the Max Number of Connections between Web Server and each Application Server on **each web server process**
  - Default (-1) means unlimited
- ▶ Most effective use of MaxConnections is to prevent backend Application Server from accepting more connections than they can reliably handle
  - Limit on number of connections
- ▶ See this link for details:
  - [http://publib.boulder.ibm.com/htpserv/ihsdiag/ihs\\_performance.html#WebSpherePlugIn](http://publib.boulder.ibm.com/htpserv/ihsdiag/ihs_performance.html#WebSpherePlugIn)

# Plug-in Load Balancing

- HttpSessionCloneID
  - ▶ Session Management Custom Property
    - [http://pic.dhe.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=%2Fcom.ibm.websphere.nd.multiplatform.doc%2Finfo%2Fae%2Fae%2Frprs\\_custom\\_properties.html](http://pic.dhe.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=%2Fcom.ibm.websphere.nd.multiplatform.doc%2Finfo%2Fae%2Fae%2Frprs_custom_properties.html)
  - ▶ Configuring a unique HTTP session clone ID for each application server using scripting
    - [http://pic.dhe.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=%2Fcom.ibm.websphere.nd.doc%2Finfo%2Fae%2Fae%2Fxml\\_httpsessionclone.html&resultof=%22httpsessionclone.html%22](http://pic.dhe.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=%2Fcom.ibm.websphere.nd.doc%2Finfo%2Fae%2Fae%2Fxml_httpsessionclone.html&resultof=%22httpsessionclone.html%22)

# Plug-in Load Balancing

- Plug-in specific links on Round Robin Load Balance
  - ▶ <http://www-01.ibm.com/support/docview.wss?rs=0&uid=swg21219567>
  - ▶ <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg21318463>
- Redbook old but good
  - ▶ <http://www.redbooks.ibm.com/abstracts/TIPS0235.html>



# Plug-in Load Balancing

- General Reference for setting up clusters and Workload Management
  - ▶ [http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun\\_srvgrp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun_srvgrp.html)
- Web Server Tuning
  - ▶ [http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tpf\\_tunewbserv.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tpf_tunewbserv.html)
- Plug-in MaxConnections
  - ▶ [http://publib.boulder.ibm.com/httperv/ihsdiag/ihs\\_performance.html#WebSpherePlugIn](http://publib.boulder.ibm.com/httperv/ihsdiag/ihs_performance.html#WebSpherePlugIn)

# Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:  
[http://www.ibm.com/software/websphere/support/supp\\_tech.html](http://www.ibm.com/software/websphere/support/supp_tech.html)
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:  
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere Community:  
<http://www.websphereusergroup.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:  
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:  
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:  
<http://www.ibm.com/software/support/einfo.html>

# Connect with us!

## 1. Get notified on upcoming webcasts

Send an e-mail to [wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com) with subject line “wste subscribe” to get a list of mailing lists and to subscribe

## 2. Tell us what you want to learn

Send us suggestions for future topics or improvements about our webcasts to [wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com)

## 3. Be connected!

Connect with us on [Facebook](#)

Connect with us on [Twitter](#)

# Questions and Answers

