User Interface - Feature #2683

reverse proxy implementation

09/04/2015 03:34 PM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Sergey Ivanovskiy	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Deployment and Management Improvements		
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to User Interface - Feature #1811: implement the AJAX client driver	**Closed**	**11/01/2013**	**03/06/2014**
Related to User Interface - Feature #3236: implement port range support for t...	**Closed**		
Related to User Interface - Feature #3287: local web client installation	**New**	**04/27/2017**	
Related to Runtime Infrastructure - Feature #5170: add support for cloud-base...	**New**		

History

#1 - 09/04/2015 03:53 PM - Greg Shah

One feature of our web client implementation (both ChUI and GUI) is the fact that we dynamically spawn the client and redirect the initial session from the P2J server to the newly spawned embedded web server on the client.

That spawning can be done on the same physical system as the P2J server, but it can easily be done on a separate/dedicated set of client systems by way of our remote spawning approach.

When the web client is being accessed from the Internet or from any situation where the P2J server and clients reside behind a firewall, we will need to make it easy for our solution to work. The normal way that this is done is to implement a reverse proxy in a DMZ:

```
Internet                DMZ                                 internal network
----------          -------------                          ----------------------
browser(s)  <--->  reverse_proxy  <--->  firewall  <---->  protected_web_servers
```

We need to make it easy to deploy our technology. Although one can require customers to manage an nginx or apache reverse proxy implementation, one idea is for us to implement our own simple Jetty reverse proxy. if not that, then we need to provide extensive/detailed instructions on how to setup/configure nginx or apache for the same usage.

Our spawning approach makes this a bit tricky, since today we allow the ports to be completely dynamic (assigned by the OS when the client initializes the embedded jetty) OR we have only a single/hard coded port. The dynamic port approach is useful for an internal server but for a firewalled environment, we could provide a range of valid ports and the server could allocate the port before spawning the client so that the client would come up requesting a specific port.

The port range idea has the benefit of being reasonable for the implementation of a static configuration in the reverse proxy. It has 2 downsides: it must be properly kept in sync with the P2J server's configuration AND it puts an upper boundary on the number of simultaneous web clients that can be in use.

Jetty does provide some built-in proxy support:

http://www.eclipse.org/jetty/documentation/current/proxy-servlet.html
http://download.eclipse.org/jetty/stable-9/apidocs/org/eclipse/jetty/proxy/ProxyServlet.html
https://reachmnadeem.wordpress.com/2014/01/23/configuring-jetty-servlet-proxy/

At a minimum, we could make our own small, mostly pre-configured web proxy that can be started and managed easily. We might even be able to make it self-configure by communicating with the P2J server and pulling its configuration dynamically.

This dynamic configuration could even be extended to supporting the current dynamic ports approach, with the key problem being that the firewall would have to be configured to allow the reverse proxy to access any port on the client system.

I am open to other ideas as well.

#2 - 03/31/2016 03:14 PM - Greg Shah

- Assignee set to Sergey Ivanovskiy

#3 - 04/01/2016 11:39 AM - Greg Shah

- Status changed from New to WIP

- Parent task deleted (#1811)

#4 - 04/01/2016 12:16 PM - Sergey Ivanovskiy

Greg, please could you provide me with references to how to configure dynamic ports for the web client? In our bzr repository we have a similar configuration with the given port number 7449.

```
        <node class="container" name="webClient">
          <node class="boolean" name="enabled">
            <node-attribute name="value" value="TRUE"/>
          </node>
          <node class="boolean" name="embedded">
            <node-attribute name="value" value="FALSE"/>
          </node>
          <node class="string" name="host">
            <node-attribute name="value" value="192.168.1.35"/>
          </node>
          <node class="integer" name="port">
            <node-attribute name="value" value="7449"/>
          </node>
          <node class="integer" name="maxBinaryMessage">
            <node-attribute name="value" value="32894"/>
          </node>
          <node class="integer" name="maxIdleTime">
            <node-attribute name="value" value="90000"/>
          </node>
          <node class="integer" name="watchdogTimeout">
            <node-attribute name="value" value="120000"/>
          </node>
        </node>
```

#5 - 04/01/2016 12:33 PM - Greg Shah

The way I recall it, you can just remove the port from the webClient portion of the directory and it will be set to 0, which tells it to use a random port.

#6 - 04/01/2016 12:35 PM - Sergey Ivanovskiy

Thank you, I could read the code.

#7 - 04/01/2016 01:43 PM - Greg Shah

Are you asking something different from what I answered?

#8 - 04/01/2016 01:56 PM - Sergey Ivanovskiy

No, exactly what I asked that dynamic ports are assigned by the Socket API if we set it zero or leave it to be unset.
Please correct me if we must use domain names for our internal network servers in order to be able to redirect requests to the started web spawner. Thus if we have some name for P2j server, we must expose it via proxy to the external network. For instance, p2j.goldencode.com can be mapped by the reverse proxy server to the started spawner and web.goldencode.com can be mapped to the P2j web server. Now we use the ajax client and it can be a problem. We must substitute the web server authorization response to the virtual reference. Correct?

#9 - 04/01/2016 02:05 PM - Sergey Ivanovskiy

Greg, do you plan to develop a standalone custom reverse proxy based on Jetty?

#10 - 04/01/2016 02:13 PM - Sergey Ivanovskiy

We can't support the trusted authorization schema to connect P2J server via the reverse proxy because it gives an anonymous access to P2J server or we can relay on a provided login, can't?

#11 - 04/01/2016 03:48 PM - Greg Shah

> Please correct me if we must use domain names for our internal network servers in order to be able to redirect requests to the started web spawner. Thus if we have some name for P2j server, we must expose it via proxy to the external network. For instance, p2j.goldencode.com can be mapped by the reverse proxy server to the started spawner and web.goldencode.com can be mapped to the P2j web server

I'm not exactly sure what you mean here by "must".

Yes, we need to support DNS names if they are in use. BUT we should not require that DNS names are in use.

In general, I would expect the reverse proxy to have its own external DNS name. The proxy would know how to contact the P2J server and would know how to proxy the spawned clients as well.

I'm also not sure what you mean by "to redirect requests to the started web spawner". Do you mean the spawned client?

> do you plan to develop a standalone custom reverse proxy based on Jetty?

Yes. But I was hoping to re-use their reverse proxy support. My idea is more about properly configuring it to make it easy to deploy in front of our server and spawned clients.

> We can't support the trusted authorization schema to connect P2J server via the reverse proxy because it gives an anonymous access to P2J server

Yes, I agree this is not a good idea. But if the customer's web app is setup to be in session with us and do the trusted spawning, then we may still need the reverse proxy for the spawned clients.

> or we can relay on a provided login, can't?

Yes, although this is not urgently needed for the customer since they will only be using the trusted auth mode.

#12 - 04/02/2016 02:13 AM - Sergey Ivanovskiy

Thank you, now the task is clear enough. Yes, "the started web spawner" should be the spawned client. Planning to take your embedded_web_client_sample as an example of how to connect P2j server and to load the required configuration and to develop a standalone web application based on org.eclipse.jetty.proxy.ProxyServlet.

#13 - 04/04/2016 10:01 AM - Greg Shah

Please provide more details of your planned implementation. I assume this approach is for a standalone "outside the firewall" reverse-proxy.

Another idea to consider: can we implement an approach that only needs a single jetty port on the P2J server? In other words, can we implement the reverse-proxy inside the P2J server, such that all client communication is redirected through the same main P2J server jetty port?

#14 - 04/04/2016 11:46 AM - Sergey Ivanovskiy

First we need to update jetty to 9.3.8 and upper since org.eclipse.jetty.proxy.ProxyServlet doesn't exist in the current version jetty-all-9.1.2.v20140210.jar.

#15 - 04/04/2016 11:49 AM - Greg Shah

Go ahead with the upgrade.

#16 - 04/04/2016 01:03 PM - Sergey Ivanovskiy

Committed revision 11093 added jetty-proxy-9.1.2.v20140210.jar from http://archive.eclipse.org/jetty/9.1.2.v20140210/dist/
Planning to update to 9.3.8 later since jetty-all jar isn't included in the jetty distribution and it needs to build it from the corresponding pom.xml by Maven.

#17 - 04/09/2016 10:03 AM - Greg Shah

- % Done changed from 0 to 20

- Target version changed from Milestone 12 to Deployment and Management Improvements

Sergey did quite a bit of research on this approach in #3018-9 (through 89). It is clear that the primary issue will be how to properly implement a transparent reverse proxy for web-sockets, which by default will not work through a transparent proxy.

It is possible to implement a custom proxy that may be able to work, but that effort is still an unknown. It is also possible that Jetty 10 will have this support built in, so working on this now is probably not a good use of our time.

In addition, the customer can potentially install and configure nginx or some other 3rd party solution for this problem. This can be done without any code needed on our part. This means this task is not really needed and is something that would enhance a deployment but is not mandatory. For that reason we are not working on this any further at this time and I am moving this task to the deployment milestone. We'll consider it further during that work.

#18 - 04/09/2016 10:03 AM - Greg Shah

- Assignee deleted (Sergey Ivanovskiy)

#19 - 11/16/2016 01:06 PM - Greg Shah

- Target version changed from Deployment and Management Improvements to Deployment and Management Improvements

#20 - 02/28/2017 04:29 PM - Greg Shah

From my reading of the above links, it seems that the web socket client will potentially be able to work without changes as long as we only use secure mode. Since this is exactly what we do, I hope that the client side is not an issue.

As a next step in this task, we should setup Apache and configure it as a reverse proxy. The tricky part here is that we want to be able to have that reverse proxy run behind a firewall and NAT. For that to work, we would need to multiplex access to a single port so that each client was uniquely mapped. I think it may be possible to use URL rewriting to make this work.

We need to be able to run multiple web clients at the same time. This means that we need to run with port set to 0 so that the web client dynamically allocates its port. We need to multiplex all usage (FWD server's jetty, embedded application web server's jetty and multiple web clients jetty) over port 443, which will make it easy to firewall and NAT (via port mapping at the firewall).

Please setup a working reverse proxy installation and get the client running in reverse proxy mode.

Thoughts?

Questions?

#21 - 03/01/2017 04:50 AM - Sergey Ivanovskiy

Greg, please correct me. What would be reverse proxy settings and functionality for outside usages if we supposed that the reverse proxy was already setup correctly?
The web client login screen is usually accessed by this url, https://localhost:7443/gui. After passing a successful client authentication, the browser client is redirected to new application url https://localhost:7449/index.html?token=... If directory.xml sets a zero web client port, then the web client should use any free port. Thus we have one entry url
https://localhost:7443/gui and many redirected urls https://localhost:freeport/index.html?token=... from these port range on Linux server [32768..60999] (cat /proc/sys/net/ipv4/ip_local_port_range)

#22 - 03/01/2017 06:17 AM - Sergey Ivanovskiy

The question is simply what is the base entry url for this reverse proxy? Is it https://localhost/gui?

#23 - 03/01/2017 06:23 AM - Greg Shah

1. I assume when a reverse proxy is being used, we will not be using localhost. The idea here is that localhost does not need reverse proxying because there are no firewall issues to resolve. For the purposes of this discussion, let's assume that the url is a fqdn like myapp.acme.com.

2. Valid protocols will be https, wss at a minimum. Are there others that we use from the web client that I am forgetting?

3. The host portion of the url will always be with myapp.acme.com.

4. The idea here is that the port actually used to communicate with Apache will be 443. This 443 port must be multiplexed to the proper back end port.

5. The rest of the url after the original / will be the same as the input url but appended to whatever we do for multiplexing.

Can we bidirectionally map something like this?

https://myapp.acme.com/7443/gui to https://myapp.acme.com:7443/gui
https://myapp.acme.com:8443 to https://myapp.acme.com:8443
https://myapp.acme.com/<freeport>/index.html?token=... to https://myapp.acme.com:<freeport>/index.html?token=...

I think that for inbound requests to the Apache server, the rewriting engine can be used to do this. I don't think the reverse proxy support can help us translate the other way, since these will already be in the document on the client side. So our FWD servers and web client would probably need to know about the reverse proxying and thus generate different urls in the documents we send back.

#24 - 03/01/2017 06:28 AM - Greg Shah

Sergey Ivanovskiy wrote:

> The question is simply what is the base entry url for this reverse proxy? Is it https://localhost/gui?

Consider that it is very likely that most customers will be using embedded mode. They probably will not use both virtual desktop mode and embedded mode, but we should plan for both to be working at the same time.

Yes, it is possible that we would want to map the FWD server OR the embedded web app as 443. This should be an option.

In fact, this is something we are doing for demo.goldencode.com when used externally. Our firewall maps demo.goldencode.com:443 to demosrv01:8443 and there are direct mappings for demo.goldencode.com:7443/demosrv01:7443 and demo.goldencode.com:7449/demosrv01:7449. When used internally, demosrv01:443 is not in use and we must go to demosrv01:8443 to make it work. So in this case, we are using port mapping in the firewall, but the idea is the same.

#25 - 03/01/2017 08:18 AM - Sergey Ivanovskiy

A reverse proxy should add these headers to requests:

```
X-Forwarded-For
    The IP address of the client.
X-Forwarded-Host
    The original host requested by the client in the Host HTTP request header.
X-Forwarded-Server
    The hostname of the proxy server.
```

It seems that these directives should do a reverse mapping

```
ProxyPass "/7443/gui"  "https://myapp.acme.com:7443/gui"
ProxyPassReverse "/7443/gui"  "https://myapp.acme.com:7443/gui"
```

the problem is if ProxyPass redirects requests for /7443/gui to https://myapp.acme.com:7443/gui, but the corresponding redirect responses has unknown Location header in this form
https://myapp.acme.com:<freeport>/index.html?token=... and ProxyPassReverse that is responsible to adjust Location, Content-Location and URI headers on HTTP redirect responses
should adjust Location https://myapp.acme.com:<freeport>/index.html?token=... to https://myapp.acme.com/<freeport>/index.html?token=... Now I am investigating the way how to do this rewrite. May be rewrite rules can help here.
Also these two modules: mod_proxy and mod_proxy_wstunnel should be enabled.

#26 - 03/01/2017 08:27 AM - Greg Shah

FYI, we will continue to configure the FWD servers/clients using either the self-signed certificates OR real public certificates (see #3240). The idea is that we are not going to use the Apache server to "terminate" the SSL session (and then use insecure sockets to communicate to FWD). In the modern world, secure sockets need to be used everywhere, between all components.

The Apache configuration is about being a gateway, not about serving actual content. Thus I am hoping that it does **not** need to be configured with those same certificates. But if it does, then that is OK.

Keep the configuration as simple as possible and please create (and post here) a full set of the configuration files that one would have to apply to get this running.

I also would like you to document the exact commands needed to install/setup Apache (including the commands to enable the modules/sites...). Someone reading this task should be able to replicate your work without doing anything more complicated than following your instructions.

#27 - 03/01/2017 02:23 PM - Sergey Ivanovskiy

I couldn't redirect https://localhost/gui to https://localhost:7443/gui

```
The proxy server could not handle the request GET /gui.

Reason: Error during SSL Handshake with remote server
```

in error logs

```
[Wed Mar 01 22:16:03.461151 2017] [proxy:error] [pid 19751] (502)Unknown error 502: [client 127.0.0.1:50706] A
H01084: pass request body failed to 127.0.0.1:7443 (127.0.0.1)
[Wed Mar 01 22:16:03.461252 2017] [proxy:error] [pid 19751] [client 127.0.0.1:50706] AH00898: Error during SSL
 Handshake with remote server returned by /gui
[Wed Mar 01 22:16:03.461263 2017] [proxy_http:error] [pid 19751] [client 127.0.0.1:50706] AH01097: pass reques
t body failed to 127.0.0.1:7443 (127.0.0.1) from 127.0.0.1 ()
[Wed Mar 01 22:16:17.700626 2017] [proxy:error] [pid 19751] (502)Unknown error 502: [client 127.0.0.1:50712] A
H01084: pass request body failed to 127.0.0.1:7443 (127.0.0.1)
[Wed Mar 01 22:16:17.700690 2017] [proxy:error] [pid 19751] [client 127.0.0.1:50712] AH00898: Error during SSL
 Handshake with remote server returned by /gui
[Wed Mar 01 22:16:17.700699 2017] [proxy_http:error] [pid 19751] [client 127.0.0.1:50712] AH01097: pass reques
t body failed to 127.0.0.1:7443 (127.0.0.1) from 127.0.0.1 ()
```

May be it requires that Apache reverse proxy uses the same certificate key and file, but it should work for different ssl setups too.

#28 - 03/01/2017 03:47 PM - Sergey Ivanovskiy

Resolved this issue by providing these options

```
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
```

#29 - 03/02/2017 03:13 AM - Sergey Ivanovskiy

The web client login page was changed to use ajax in order to fix the embedded client to work with IE. I didn't find another solution. The problem was that if the embedded page gets a redirect response, then IE is failed to follow this link and displays certificate's errors warnings. Thus we have a programmatic redirect to this application page https://myapp.acme.com:<freeport>/index.html?token=... with normal code 200 instead of HTTP redirect with code 302, 301. It weakens application security since it gives a knowledge how this given string is encoded by HTTPS although this token=... is used only once. I have two ways to roll back The web client login page changes in order to use HTTP redirect. In that case Header edit directive should help to rewrite a redirect url, but in this case IE embedded web client shouldn't work and this issue should be reopened. The other way is to use REFERER header and to rewrite this https://myapp.acme.com:freeport/index.html?token=... to https://myapp.acme.com/freeport/index.html?token=... by ajax client.
Greg, do you have objections to follow the first or the second way?

#30 - 03/02/2017 04:10 AM - Sergey Ivanovskiy

Also we can change redirect url to something like https://${X-Forwarded-Server}/freeport/index.html?token=... if X-Forwarded-Server header is present.

#31 - 03/02/2017 05:21 AM - Sergey Ivanovskiy

I decided to use X-Forwarded-Server and X-Forwarded-Proto to rewrite web client uri if a reverse proxy is present.

#32 - 03/02/2017 06:41 AM - Sergey Ivanovskiy

Another trouble is that it seems that Apache has a bug (https://bz.apache.org/bugzilla/show_bug.cgi?id=46665) that prevents these usages:

```
ProxyPassMatch ^(/.*\.page)$ ajp://hostname:8009$1
```

is not valid

```
ProxyPassMatch ^/(.*\.page)$ ajp://hostname:8009/$1
```

is valid.
In our case we have

```
ProxyPassMatch ^/([0-9]+)/(index\.html.*)$  https://localhost:$1/$2
```

that is not valid.

#33 - 03/02/2017 06:46 AM - Greg Shah

Sergey Ivanovskiy wrote:

> I decided to use X-Forwarded-Server and X-Forwarded-Proto to rewrite web client uri if a reverse proxy is present.

OK.

#34 - 03/02/2017 06:57 AM - Sergey Ivanovskiy

- File rewrite_uri_diff.txt added

Please review the current diff that builds https://${X-Forwarded-Server}/freeport/index.html?token=... uri if X-Forwarded-Server is present.
Now I have difficulties how to configure a reverse proxy settings to use https://127.0.0.1/7449/index.html?token=7f0e9fb5bff384af9782bb55ba95607d.

#35 - 03/02/2017 07:01 AM - Sergey Ivanovskiy

Now, I am not sure that it is a good practice to open a forward from https://127.0.0.1/7449/index.html?token=7f0e9fb5bff384af9782bb55ba95607d to https://127.0.0.1:7449/index.html?token=7f0e9fb5bff384af9782bb55ba95607d?

#36 - 03/02/2017 07:05 AM - Greg Shah

Some thoughts:

1. We don't want to assume that reverse proxy is always being used. We need to support both direct and reverse-proxy methods. Preferably, we would dynamically detect which mode to use based on how the client makes its request.

2. The reverse proxy mode will break the same origin policy. Because we are pushing the port into the path portion of the url, the origin of all of our jetty instances will be the same. This has the potential to open security holes. Please consider the ramifications and document them here.

3. The current approach won't work for remote spawners (spawned clients on a different internal system than the FWD server). You don't have to solve this problem right now, I just want to document it here.

#37 - 03/02/2017 07:11 AM - Greg Shah

Sergey Ivanovskiy wrote:

> Now, I am not sure that it is a good practice to open a forward from
> https://127.0.0.1/7449/index.html?token=7f0e9fb5bff384af9782bb55ba95607d to
> https://127.0.0.1:7449/index.html?token=7f0e9fb5bff384af9782bb55ba95607d?

I agree this is probably not a good idea. We don't need to forward the IP address portion anyway, so why open the security and possible redirection problems that come with it?

#38 - 03/02/2017 07:43 AM - Sergey Ivanovskiy

The problem is that this rule ProxyPassMatch ^/([0-9]+)/(index\.html.*)$ https://localhost:$1/$2 can't be validated by Apache server since it opens a way to forward https://proxyhostname/target-port/index.html to https://backendhostname:target-port/index.html, but this port can be used for internal service and then this service can be loaded through our reverse proxy.
But this ProxyPassMatch ^/([0-9]+)/(index\.html.*)$ https://localhost/$2?param=$1 is valid and ProxyPassMatch ^/([0-9]+)/(index\.html.*)$ https://localhost:7449/$2 is valid too.
Thus we need to find another ways to multiplex requests, may be it requires a custom solution.

#39 - 03/02/2017 09:07 AM - Sergey Ivanovskiy

This is an interesting thread about "Support proxies with WebSocketClient" https://github.com/eclipse/jetty.project/issues/117. Please look at its last question "the usage of ProxyServlet for a reverse proxy of WebSockets".

#40 - 03/02/2017 09:36 AM - Greg Shah

In summary: "It is not yet supported in jetty.".

#41 - 03/02/2017 09:58 AM - Sergey Ivanovskiy

Greg, does it make sense to support a range of free IP addresses in directory.xml. For an example, the web client gets any available port from this range 32768 .. 32868

```
<node class="container" name="webClient">
   <node class="integer" name="port">
        <node-attribute name="from" value="32768"/>
        <node-attribute name="to" value="32868"/>
   </node>
</node>
```

and then defines reverse proxy configuration for all these ports.
It seems that index.html should take into account the reverse proxy since it contains these links

```
      <link rel="stylesheet" type="text/css" href="/dojo-toolkit/dijit/themes/claro/claro.css">
      <!-- load dojo -->
      <script src="/dojo-toolkit/dojo/dojo.js" data-dojo-config="async: true"></script>

      <script type="text/javascript" src="/common/p2j.js"></script>
      <script type="text/javascript" src="/common/p2j.logger.js"></script>
      <script type="text/javascript" src="/common/p2j.sound.js"></script>
      <script type="text/javascript" src="/client/p2j.strokes.js"></script>
      <script type="text/javascript" src="/client/p2j.canvas_renderer.js"></script>
      <script type="text/javascript" src="/client/p2j.virtual_desktop.js"></script>
      <script type="text/javascript" src="/client/p2j.mouse.js"></script>
      <script type="text/javascript" src="/client/p2j.screen.js"></script>
      <script type="text/javascript" src="/common/p2j.socket.js"></script>
      <script type="text/javascript" src="/common/p2j.keymap.js"></script>
      <script type="text/javascript" src="/common/p2j.keyboard.js"></script>
      <script type="text/javascript" src="/client/p2j.clipboard_helpers.js"></script>
      <script type="text/javascript" src="/common/p2j.clipboard.js"></script>
      <script type="text/javascript" src="/common/p2j.fonts.js"></script>
      <script type="text/javascript" src="/common/p2j.remote.js"></script>
```

that are incorrect in this schema and should be remapped under /32768/.... in order to get them using these rules

```
ProxyPass /gui  https://localhost:7443/gui
ProxyPassReverse /gui https://localhost:7443/gui
ProxyPassMatch ^/7449/(index\.html.*)$  https://localhost:7449/$1
ProxyPassReverse ^/7449/(index\.html.*)$ https://localhost:7449/index.html
```

Greg, should I continue this way?

#42 - 03/02/2017 10:13 AM - Greg Shah

> does it make sense to support a range of free IP addresses in directory.xml

Yes. That is the idea in #3236.

> and then defines reverse proxy configuration for all these ports.

You mean to do this in Apache?

> that are incorrect in this schema and should be remapped under /32768/.... in order to get them using these rules

Do I understand correctly that every possible URI must be mapped explicitly?

If so, then this has two implications:

1. Every time we change our code to have any additional URIs, we must change the Apache proxy config.

2. It would stop us from doing anything to implement a dynamic URI. This will be needed to provide a mechanism that allows applications to generate content and load it in the browser. We already have customer requirements to do this. An example is when the application generates a CSV or Excel spreadsheet and needs to allow the user to open it. Today, they directly launch Excel and access a local file. In the web client, we need to change this to create the resource as a URI and allow the user's browser to load it.

Hopefully, font and bitmap resources would be immune to this because those are handled internal to our websocket implementation. But I worry that the other requirements are a problem.

#43 - 03/02/2017 10:41 AM - Sergey Ivanovskiy

Yes, you are correct, I meant explicit mappings. At that moment we consider this schema
1) https://proxy/gui <-> https://backend:7443/gui
2) https://proxy/7449/index.html <-> https://backend:7449/index.html

3) https://proxy/7449/* <-> https://backend:7449/$1
In this schema all resources have this root https://proxy/, but in order to be mapped correctly, they should be from https://proxy/7449.
Now I have no workable Apache configuration that can be used to work even with an explicitly defined web client port 7449.
Should I continue to get this configuration working properly?

#44 - 03/02/2017 11:11 AM - Greg Shah

> Should I continue to get this configuration working properly?

Do you have work to do for #3222? If so, it has the priority.

Do you have other ideas of a better approach?

#45 - 03/02/2017 01:32 PM - Sergey Ivanovskiy

Ok, planning to investigate a better approach. I used some Google help how to apply html_module that rewrites html links and got this workable configuration

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost
                ServerName acme
                ServerAlias acme
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile    /etc/apache2/ssl/certs/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/private/apache.key

        <FilesMatch "\.(cgi|shtml|phtml|php)$">
                SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
                SSLOptions +StdEnvVars
        </Directory>

SSLProxyEngine On
ProxyRequests Off
ProxyReceiveBufferSize 4096
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
RewriteEngine On
RewriteCond %{HTTP:Connection} Upgrade [NC]
RewriteCond %{HTTP:Upgrade} websocket [NC]
RewriteRule /7449/(.*) wss://127.0.0.1:7449/$1 [P,L]

ProxyPreserveHost On
ProxyAddHeaders On

ProxyPass /gui  https://localhost:7443/gui
ProxyPassReverse /gui https://localhost:7443/gui

ProxyPass /7449  https://localhost:7449
```

```
ProxyPassReverse /7449 https://localhost:7449
ProxyPass /7449/common  https://localhost:7449/common
ProxyPassReverse /7449/common https://localhost:7449/common
ProxyPass /7449/client  https://localhost:7449/client
ProxyPassReverse /7449/client https://localhost:7449/client

ProxyPass /7449/ajax  wss://127.0.0.1:7449/ajax
ProxyPassReverse /7449/ajax wss://127.0.0.1:7449/ajax

<Location /7449/>
   ProxyPassReverse /
   SetOutputFilter  proxy-html
   ProxyHTMLLinks  source src
ProxyHTMLLinks  a          href
ProxyHTMLLinks  area       href
ProxyHTMLLinks  link       href
ProxyHTMLLinks  img        src longdesc usemap
ProxyHTMLLinks  object     classid codebase data usemap
ProxyHTMLLinks  q          cite
ProxyHTMLLinks  blockquote cite
ProxyHTMLLinks  ins        cite
ProxyHTMLLinks  del        cite
ProxyHTMLLinks  form       action
ProxyHTMLLinks  input      src usemap
ProxyHTMLLinks  head       profile
ProxyHTMLLinks  base       href
ProxyHTMLLinks  script     src for
   ProxyHTMLURLMap https://localhost:7449 /7449/
   ProxyHTMLURLMap / /7449/
   ProxyHTMLURLMap  /7449/ /7449/
   RequestHeader    unset  Accept-Encoding
</Location>

    </VirtualHost>
</IfModule>
```

but it needs to fix wss web socket path in index.html

#46 - 03/02/2017 01:39 PM - Sergey Ivanovskiy

It doesnt work without this patch rewrite_uri_diff.txt from the note 34 (explanations note 29) and it requires to fix this

```
                'socket'    : {
                                'url' : 'wss://' + window.location.host + "/7449" + '${context}/ajax',
                                'maxBinaryMessage' : ${maxBinaryMessage},
                                'maxIdleTime'      : ${maxIdleTime},
                                'watchdogTimeout'  : ${watchdogTimeout}
                                },
```

in the template page index.html. We can get rid off this html_module Apache module if we implement a root prefix for index.html template page.

#47 - 03/02/2017 01:56 PM - Sergey Ivanovskiy

Obviously, we can see the current approach to set web socket path is not good.

```
                'socket'    : {
                                'url' : 'wss://' + window.location.host + '${context}/ajax',
                                'maxBinaryMessage' : ${maxBinaryMessage},
                                'maxIdleTime'      : ${maxIdleTime},
                                'watchdogTimeout'  : ${watchdogTimeout}
                                },
```

#48 - 03/23/2017 01:34 PM - Greg Shah

- Related to Feature #3236: implement port range support for the web client's embedded web server added

#49 - 06/15/2017 03:00 PM - Sergey Ivanovskiy

Created new branch 2683a for this task.

#50 - 06/15/2017 03:08 PM - Sergey Ivanovskiy

The plan is to add X-Forwarded-Host support by web server to span web client correctly in the presence of the reverse proxy and to implement the root prefix for each resource path in this template index.html.

#51 - 06/19/2017 08:04 AM - Greg Shah

Please make sure that we can run in reverse proxy mode AND/OR in direct access mode (our current approach). Many installations won't use the reverse proxy, some may only use reverse proxy and some may use both modes at the same time.

If we can implement so that both modes work on the same server at the same time, that would be optimal.

#52 - 06/19/2017 08:58 AM - Sergey Ivanovskiy

Understand this requirement, I think it can be implemented easily using that the reverse proxy can be recognized by additional HTTP headers. If X-Forwarded-Host is present in the POST request, then it is a proxies request, otherwise it is handled as a direct request.

#53 - 06/20/2017 08:52 AM - Greg Shah

Please note that it is NOT a requirement to implement an Nginx configuration. If the Apache approach works well, that is good enough.

I just mentioned Nginx because if the reverse proxy is easier to configure or works better with Nginx, then we can consider it.

Since we are in a time crunch, we cannot afford to spend unnecessary time.

#54 - 06/20/2017 05:23 PM - Sergey Ivanovskiy

It seems that Apache can be configured using map files that simplifies its configuration file to this one

```
ProxyRequests Off
SSLProxyEngine On
ProxyReceiveBufferSize 4096
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
RewriteEngine On

RewriteMap  clients-to-backends   "txt:/etc/apache2/map.clients-to-backends"

RewriteCond %{HTTP:Connection} Upgrade [NC]
RewriteCond %{HTTP:Upgrade} websocket [NC]
RewriteRule /server/([^/]+)/(.*)  wss://${clients-to-backends:$1}/$2 [P,L]
RewriteRule /server/([^/]+)/(.*) https://${clients-to-backends:$1}/$2 [P,L]

ProxyPreserveHost On
ProxyAddHeaders On

ProxyPass /gui  https://localhost:7443/gui
ProxyPassReverse /gui https://localhost:7443/gui
```

and this text file map.clients-to-backends defines port names to backends mapping. I tested with this range

```
7449 localhost:7449
7450 localhost:7450
7451 localhost:7451
7452 localhost:7452
7453 localhost:7453
7454 localhost:7454
```

#55 - 06/20/2017 05:36 PM - Greg Shah

That is great!

#56 - 06/21/2017 02:52 PM - Sergey Ivanovskiy

- File 2683a_current.txt added

Greg, please look at the current 2683a, I tried to explore the idea from https://proj.goldencode.com/issues/3236#note-3. In this diff the range of ports is supposed to be from 7449 and up, tested these changes with the note 54 Apache configuration. I will proceed this construction if you approve that it is a correct way.

#57 - 06/21/2017 02:54 PM - Sergey Ivanovskiy

It is important that the web client is not started with the remote debug mode, because in this case it is not possible to create different web clients that use the same remote debug port.

#58 - 06/21/2017 03:01 PM - Sergey Ivanovskiy

- File deleted (2683a_current.txt)

#59 - 06/21/2017 03:01 PM - Sergey Ivanovskiy

- File 2683a_current.txt added

#60 - 07/03/2017 04:17 AM - Greg Shah

Please merge 2683a to trunk.

#61 - 07/03/2017 05:53 AM - Sergey Ivanovskiy

The task branch 2683a was merged into the trunc as rev. 11154 and archived.

#62 - 07/03/2017 05:59 AM - Greg Shah

- Assignee set to Sergey Ivanovskiy

- % Done changed from 20 to 90

#63 - 07/04/2017 09:42 AM - Sergey Ivanovskiy

Private and public root CA keys are stored in root-ca-pk.store protected by store password and key entry password. Let it be a string "storePassword" and key entry password be a "keyEntryPassword". Then the following procedure helps to get these private and public keys for the root CA.

In order to export private key we need to convert JKS store format into PKCS12 format by keytool and then we retrieve private key in unencrypted format using openssl command line tool.

This command produces a warning indicating that PKCS12 format doesn't support different protected passwords for the store and key entries.

```
keytool -importkeystore -alias hotel-root -srckeystore root-ca-pk.store -destkeystore apache.pkcs12 -srcstorepass "storePassword" -deststorepass "storePassword" -srcstoretype JKS -deststoretype PKCS12 -srckeypass "keyEntryPassword" -destkeypass "keyEntryPassword"
```

Then after the root store root-ca-pk.store has been converted into PKCS12 format apache.pkcs12, the private key apache.key can be retrieved by this command

```
openssl pkcs12 -in apache.pkcs12 -nodes -nocerts -out apache.key -passin pass:"storePassword"
```

But the public key apache.crt can be got by keytool without converting the root store into PKCS12 format. Thus we have these two different ways to

get the target public key

```
openssl pkcs12 -in apache.pkcs12 -nokeys -out apache.crt -passin pass:"storePassword"
```

or

```
keytool -exportcert -alias hotel-root -file apache.crt -storepass "storePassword" -keystore root-ca-pk.store -storetype JKS -providerName SUN -rfc
```

Finally, these private apache.key and public apache.crt keys can be used to setup for Apache ssl configurations.

```
<IfModule mod_ssl.c>
   <VirtualHost _default_:443>
.............................
        SSLEngine on
        SSLCertificateFile    /etc/apache2/ssl/certs/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/private/apache.key
.............................
   </VirtualHost>
</IfModule>
```

#64 - 07/10/2017 09:11 AM - Sergey Ivanovskiy

For external certificates apache.crt issued by well-known authorities it is required to add SSLCertificateChainFile option that includes the CA intermediate certificate file chain.crt. This intermediate CA certificate was used to sign the delivered (exported by the known authority) server's certificate apache.crt

```
SSLEngine on
SSLCertificateFile    /etc/apache2/ssl/certs/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/private/apache.key
SSLCertificateChainFile /etc/apache2/ssl/certs/chain.crt
```

#65 - 03/21/2019 07:52 AM - Greg Shah

- Status changed from WIP to Closed

- % Done changed from 90 to 100

#66 - 03/09/2020 03:14 PM - Sergey Ivanovskiy

- File broker_manager.patch added

I tested this Apache Reverse proxy configuration with this branch 4335a using Hotel Gui application. The current code has the gap when BrokerManger.getOrderedNumber(String host) is used. It requires to set up a broker configuration node by adding

```
<node class="container" name="brokers">
  <node class="broker" name="broker1">
    <node-attribute name="account" value="OS-user"/>
  </node>
</node>
```

to directory.xml under server/default in order that hosts.txt file will be read and initialized.
This file hosts.txt enumerates FWD backends and maps a backend IP4 address to a unique number.
It was introduced in the task #3287.
When checking the Hotel GUI application with the Apache Reverse proxy settings I found two issues:
1) The js web client sometimes hungs and doesn't proceed the business logic but it looks like the js web client loads required js modules successfully and starts communication with the web client.

But this issue is not reproduced with the next connections to the web server.

2) When the web client is set to use reverse proxy configuration, then it is not possible to connect to the web client directly without the Apache web proxy.
It seems this task should be resolved somehow.

I propose to initialize the hosts map before reading the broker manager configurations. Please review this diff.

#67 - 03/09/2020 03:16 PM - Sergey Ivanovskiy

- Related to Feature #3287: local web client installation added

#68 - 03/09/2020 03:28 PM - Greg Shah

I don't have a problem with the patch, but Constantin should review.

> It requires to set up a broker configuration node by adding

Is there a reason that this is useful? I think we would want a sensible default that doesn't need any configuration.

> 1) The js web client sometimes hungs and doesn't proceed the business logic but it looks like the js web client loads required js modules successfully and starts communication with the web client.

Is the BrokerManager patch related to this?

> When the web client is set to use reverse proxy configuration, then it is not possible to connect to the web client directly without the Apache web proxy.

It seems this task should be resolved somehow.

I agree.

These changes cannot go into 4335a since we are trying to get that branch through testing. We need another branch.

Constantin: I think we should consider 4231b as the next branch. Adrian has been working to resolve regressions.

#69 - 03/09/2020 03:41 PM - Sergey Ivanovskiy

No, the broker manager patch is related to the case when the broker manager is not used but clients to ports map is required. We need the well-defined clients to port map in order to be used with the Apache reverse proxy configuration. The js web client hanging issue is unrelated. I am investigating its root causes.

#70 - 03/09/2020 04:04 PM - Sergey Ivanovskiy

The hanging issue has this thread context:
1) KeyReader thread is in this state

```
Name: KeyReader
State: WAITING on java.lang.Object@24fdf9e9
Total blocked: 299  Total waited: 300

Stack trace:
java.lang.Object.wait(Native Method)
java.lang.Object.wait(Object.java:502)
com.goldencode.p2j.ui.client.driver.web.WebClientProtocol.readKey(WebClientProtocol.java:809)
com.goldencode.p2j.ui.client.gui.driver.web.GuiWebDriver.readKey(GuiWebDriver.java:821)
com.goldencode.p2j.ui.client.OutputManager.readKey(OutputManager.java:771)
com.goldencode.p2j.ui.client.TypeAhead$KeyReader.run(TypeAhead.java:645)
com.goldencode.p2j.security.ContextAwareThread.run(ContextAwareThread.java:133)
com.goldencode.p2j.security.AssociatedThread.run(AssociatedThread.java:79)
```

2) main thread

```
Name: main
State: WAITING on com.goldencode.p2j.ui.client.TypeAhead@1ba2fd8e
Total blocked: 238  Total waited: 173

Stack trace:
java.lang.Object.wait(Native Method)
com.goldencode.p2j.ui.client.TypeAhead.getKeystroke(TypeAhead.java:443)
com.goldencode.p2j.ui.chui.ThinClient.waitForEvent(ThinClient.java:15836)
com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:13417)
com.goldencode.p2j.ui.chui.ThinClient.lambda$waitForWorker$60(ThinClient.java:12986)
com.goldencode.p2j.ui.chui.ThinClient$$Lambda$70/1431710377.get(Unknown Source)
com.goldencode.p2j.ui.chui.ThinClient.lambda$doInteractive$74(ThinClient.java:18008)
com.goldencode.p2j.ui.chui.ThinClient$$Lambda$71/1348916831.run(Unknown Source)
com.goldencode.p2j.ui.chui.ThinClient.doInteractive(ThinClient.java:17984)
com.goldencode.p2j.ui.chui.ThinClient.doInteractive(ThinClient.java:18008)
com.goldencode.p2j.ui.chui.ThinClient.waitForWorker(ThinClient.java:12985)
com.goldencode.p2j.ui.chui.ThinClient.waitFor(ThinClient.java:12928)
com.goldencode.p2j.ui.chui.ThinClient.promptFor(ThinClient.java:11646)
com.goldencode.p2j.ui.ClientExportsMethodAccess.invoke(Unknown Source)
com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:156)
com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:755)
com.goldencode.p2j.net.Conversation.block(Conversation.java:412)
com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:348)
com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1201)
com.goldencode.p2j.net.Queue.transact(Queue.java:672)
com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
com.sun.proxy.$Proxy14.standardEntry(Unknown Source)
com.goldencode.p2j.main.ClientCore.start(ClientCore.java:374)
com.goldencode.p2j.main.ClientCore.start(ClientCore.java:165)
com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:250)
com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
```

3) The server Conversation thread is at this state

```
Name: Conversation [0000000F:bogus]
State: WAITING on java.lang.Object@c61305
Total blocked: 24  Total waited: 25

Stack trace:
java.lang.Object.wait(Native Method)
java.lang.Object.wait(Object.java:502)
com.goldencode.p2j.net.Conversation.block(Conversation.java:391)
com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:348)
com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1201)
com.goldencode.p2j.net.Queue.transact(Queue.java:672)
com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
com.sun.proxy.$Proxy18.isChui(Unknown Source)
com.goldencode.p2j.ui.LogicalTerminal.init(LogicalTerminal.java:1302)
com.goldencode.p2j.security.ContextLocal.initializeValue(ContextLocal.java:655)
com.goldencode.p2j.security.ContextLocal.get(ContextLocal.java:494)
com.goldencode.p2j.security.ContextLocal.get(ContextLocal.java:430)
com.goldencode.p2j.ui.LogicalTerminal.locate(LogicalTerminal.java:11414)
com.goldencode.p2j.ui.LogicalTerminal.setClientMetrics(LogicalTerminal.java:12561)
com.goldencode.p2j.ui.LogicalTerminalMethodAccess.invoke(Unknown Source)
com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:156)
com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:755)
com.goldencode.p2j.net.Conversation.block(Conversation.java:412)
com.goldencode.p2j.net.Conversation.run(Conversation.java:232)
java.lang.Thread.run(Thread.java:748)
```

The Conversation thread is waiting the client in ErrorManager.setSystemAlertBoxes(!client.isChui()); line 1302 LogicalTerminal.

#71 - 03/09/2020 04:31 PM - Sergey Ivanovskiy

It seems that this hanging issue is reproduced only when I tried to connect https://localhost/gui and the Apache proxy server is redirected to this page https://192.168.1.37/server/client1005/index.html since the forwarded host is given by 192.168.1.37 and when I used this path https://192.168.1.37/gui, then this issue is not reproduced. Please look at https://proj.goldencode.com/projects/p2j/wiki/Reverse_Proxy.
The hanging js web client has this last message MSG_SET_UPLOAD_FILE_SIZE_LIMITS according to the js web console log and then has only series of ping/pong messages.

#72 - 03/09/2020 04:59 PM - Sergey Ivanovskiy

It is interesting if we should remove logs here in ThinClient.initializePost because it would block the web client?

```
      // initialize drag and drop engine
      if (!om.isChui())
      {
         ((GuiDriver<?, ?>)tc.tk.getInstanceDriver()).initDragAndDrop(tc.server);
      }

      dynamicLayoutEnabled = !tc.isChui() && DynamicLayoutConfig.getInstance().isDynamicLayoutEnabled();

      // register external widget implementations
      ClientConfigManager ccm = ClientConfigManager.getInstance();

      // Note that this call must not happen during client configuration manager init. On the client the
      // descriptors must be loaded from the server and client config manager is itself part of the
      // network protocol (it is needed during @Protocol.attachChanges@).
      ccm.registerExternalWidgetConfigs();

      WidgetDescriptorHelper.getDescriptors().stream().forEach(descr ->
      {
         Class<?> cfgClass = descr.getConfigClass();
         Class<?> implClass = descr.getImplClass();

         if (cfgClass != null && implClass != null)
         {
            om.getFactory().addWidgetClass(cfgClass, implClass, implClass);
            descr.initializeClient(session);
            LOG.log(Level.INFO, "Registered widget implementation: " + cfgClass.getName() + " " +
                                implClass.getName() + ".");
         }
      });
   }
```

#73 - 03/09/2020 05:38 PM - Sergey Ivanovskiy

Yes, it looks like the old issue documented in WebClientProtocol

```
* IMPORTANT: Do not use logging anywhere inside this class. When an authentication plugin
 * is running during session establishment with the P2J server using logging no messages are
 * written into the log and the application is blocked.
```

I removed the logging and it seems the hanging issue becomes unreproduceable.

```
=== modified file 'src/com/goldencode/p2j/ui/ConfigManager.java'
--- src/com/goldencode/p2j/ui/ConfigManager.java    2020-03-05 01:30:25 +0000
+++ src/com/goldencode/p2j/ui/ConfigManager.java    2020-03-09 21:09:11 +0000
@@ -247,8 +247,8 @@
          if (clz != null)
          {
             addWidgetConfiguration(clz);
-            LOG.log(Level.INFO,
-                    "Registered widget configuration class '" + clz.getName() + "'.");
+//            LOG.log(Level.INFO,
+//                    "Registered widget configuration class '" + clz.getName() + "'.");
          }
       });
    }

=== modified file 'src/com/goldencode/p2j/ui/chui/ThinClient.java'
--- src/com/goldencode/p2j/ui/chui/ThinClient.java    2020-02-25 13:35:40 +0000
+++ src/com/goldencode/p2j/ui/chui/ThinClient.java    2020-03-09 21:07:44 +0000
@@ -3411,8 +3411,8 @@
          {
             om.getFactory().addWidgetClass(cfgClass, implClass, implClass);
             descr.initializeClient(session);
-            LOG.log(Level.INFO, "Registered widget implementation: " + cfgClass.getName() + " " +
-                                implClass.getName() + ".");
+//            LOG.log(Level.INFO, "Registered widget implementation: " + cfgClass.getName() + " " +
+//                                implClass.getName() + ".");
          }
       });
    }
```

#74 - 03/09/2020 06:23 PM - Constantin Asofiei

Greg Shah wrote:

> I don't have a problem with the patch, but Constantin should review.

I'm OK with the patch.

> Constantin: I think we should consider 4231b as the next branch. Adrian has been working to resolve regressions.

I agree.

#75 - 03/09/2020 06:24 PM - Constantin Asofiei

Sergey Ivanovskiy wrote:

> Yes, it looks like the old issue documented in WebClientProtocol

OK, please add javadocs/warnings to these classes, too.

#76 - 03/10/2020 08:42 AM - Sergey Ivanovskiy

I can't use 4231b for #2683-73 changes because these code exist only in 4335a. These changes are only to remove logs and don't require to restart regression testing.

#77 - 03/10/2020 09:10 AM - Sergey Ivanovskiy

Sergey Ivanovskiy wrote:

> I can't use 4231b for #2683-73 changes because these code exist only in 4335a. These changes are only to remove logs and don't require to restart regression testing.

I would like to clarify that the hanging issue in 4335a that I observed with the Apache reverse proxy use case could affect the embedded mode too.

#78 - 03/11/2020 02:18 PM - Greg Shah

Although it should be safe, I prefer to avoid changes in 4335a. Please put your changes into 4231b after it is rebased from the trunk revision that has 4335a (rev 11345, I hope).

I agree with Constantin, the changes will need comments like this in both methods:

```
// WARNING: Do NOT put logging in this method!  It is used during the early phases
// of the web client session initialization.  Any logging during the authentication
// processing of the session init will cause a hang.  Don't log here!
```

The changes that cause the hang are related to the new external widget support.

#79 - 03/11/2020 02:25 PM - Sergey Ivanovskiy

Greg, please could you provide me with information about external widget support? Are there ready examples?

#80 - 03/11/2020 02:54 PM - Greg Shah

We don't have documentation yet. External widget support is only needed in special cases where the code cannot be placed directly in FWD. An example would be a customer-specific widget. The work to provide the necessary "hooks" was done in branch 3821b, some discussions are in #3822 and you can see a sample application in #3820-305. Anyway, removing the logging is safe to do.

#81 - 03/17/2020 07:51 AM - Sergey Ivanovskiy

Constantin Asofiei wrote:

> Sergey Ivanovskiy wrote:
>
> > Yes, it looks like the old issue documented in WebClientProtocol
>
> OK, please add javadocs/warnings to these classes, too.

Please review the committed revision 11372 (4231b).

#82 - 03/17/2020 08:37 AM - Greg Shah

Code Review Task Branch 4231b Revision 11372

Overall, it is a nicer implementation to have the HostsManager.

1. The HostsManager should probably be a singleton. The current code is effectively a singleton but only because BrokerManager implements it that way. It seems better to hide such details inside the HostsManager. For example, we must remember that the BrokerManager must always be initialized before the the WebClientsManager. This is an unnecessary dependency.

2. Is there a reason that the HOSTS_FILE_PATH is inside BrokerManager? It seems better to be inside HostsManager.

3. Is there a reason that we should use a separate hosts file instead of putting the mapping in the directory? The directory is the more obvious place for this. It seems better to be there than to have a "one off" configuration file.

4. The comments added to ThinClient should be // style since they are inline with code.

5. I expected the logging to be disabled in the ConfigManager but it was unchanged.

#83 - 03/17/2020 01:02 PM - Sergey Ivanovskiy

Committed revision 11374 (4231b) fixed 1), 2), 4) and 5). The hosts file is used for mapping unique pairs of host and port to the client's names. It came from #3287. Its implementation looks simple but it is required to support this file separately. Should I change it to use the directory?

#84 - 03/17/2020 01:45 PM - Greg Shah

> Should I change it to use the directory?

Is there any advantage to leaving it as a separate file?

#85 - 03/17/2020 02:29 PM - Sergey Ivanovskiy

This hosts file is related only to the network configuration. It is used only to generate mapping on the fly for new connected broker client. I don't know which of the directory nodes can collect these data:
IP41 1
IP42 2
.....
IP4N N
Actually these data are not related to any user or broker #3287-63.

#86 - 03/17/2020 03:22 PM - Sergey Ivanovskiy

I rechecked if Jetty implements the reverse proxy support for wss connections and found that https://webtide.com/http2-with-haproxy-and-jetty/ advised to use haproxy (https://www.haproxy.com/blog/websockets-load-balancing-with-haproxy/) This thread https://discourse.haproxy.org/t/using-reverse-proxy-with-secured-web-sockets-wss/2917 has some useful information that I didn't check. It seems that it is possible to setup haproxy as a reverse proxy for FWD different network use cases. Should we check it?

#87 - 03/17/2020 03:31 PM - Greg Shah

> It is used only to generate mapping on the fly for new connected broker client.

Is it used for anything external to the FWD server?

If yes, then why do we want it as a separate file?

> Actually these data are not related to any user or broker #3287-63.

It is not clear to me what #3287-63 has to do with this hosts file.

> It seems that it is possible to setup haproxy as a reverse proxy for FWD different network use cases. Should we check it?

Sorry, I did not have time to read all the links. What is the advantage in comparison to using Apache?

#88 - 03/17/2020 03:44 PM - Sergey Ivanovskiy

> > It is used only to generate mapping on the fly for new connected broker client.
>
> Is it used for anything external to the FWD server?

No

If yes, then why do we want it as a separate file?

Actually these data are not related to any user or broker #3287-63.

It is not clear to me what #3287-63 has to do with this hosts file.

Yes, it doesn't use it. I would like to understand under which of the directory node these information can be saved?

It seems that it is possible to setup haproxy as a reverse proxy for FWD different network use cases. Should we check it?

Sorry, I did not have time to read all the links. What is the advantage in comparison to using Apache?

Understand. I have no such information. I tested only the Apache reverse proxy. Jetty doesn't have documentations how to setup jetty embedded reverse proxy with websocket support.

#89 - 06/20/2020 12:02 PM - Greg Shah

Task branch 4231b has been merged to trunk as revision 11347.

#91 - 03/02/2021 10:12 AM - Greg Shah

- Related to Feature #5170: add support for cloud-based load balancing and WAF added

Files

rewrite_uri_diff.txt	2.76 KB	03/02/2017	Sergey Ivanovskiy
2683a_current.txt	23 KB	06/21/2017	Sergey Ivanovskiy
broker_manager.patch	3.12 KB	03/09/2020	Sergey Ivanovskiy