Runtime Infrastructure - Feature #4406

server-side REST execution without appserver agents

11/12/2019 12:31 PM - Greg Shah

Status:	New		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:			**Estimated time:**	0.00 hour
Target version:				
billable:	No		**version:**	
vendor_id:	GCD			

Description

Related issues:	
Related to Base Language - Feature #4065: server-side processing of client pl...	**WIP**
Related to Base Language - Feature #3254: add support for running 4GL on mult...	**WIP**
Related to User Interface - Feature #3931: single sign-on for virtual desktop...	**Closed**
Related to User Interface - Feature #4912: move UI portions of the web client...	**New**
Related to Runtime Infrastructure - Feature #5170: add support for cloud-base...	**New**
Related to Runtime Infrastructure - Feature #6646: short circuit legacy REST ...	**New**

History

#1 - 11/12/2019 12:37 PM - Greg Shah

The current design of our appserver support, including how we execute REST requests, is to forward requests to a pool of agent sessions. Each agent requires a FWD client and any platform specific processing will occur in that client JVM (which runs in batch mode).

The idea is to allow these requests to occur directly on the calling thread instead of forwarding the request to the agent. Doing this will greatly reduce the overhead of the current approach and should scale better.

The downside is that we have to consider security more carefully, handle swapping session context on the calling thread and we must limit/eliminate client side usage. Or we can enable the client-side usage to be executed in the server, but again it requires some security considerations to be handled.

#2 - 11/12/2019 12:38 PM - Greg Shah

- Related to Feature #4065: server-side processing of client platform dependencies added

#3 - 01/09/2020 05:04 PM - Greg Shah

- Related to Feature #3254: add support for running 4GL on multiple threads in a single session added

#4 - 04/09/2020 06:46 AM - Greg Shah

I think that the solution should be made generic enough that it can work with non-REST cases as well. I think this means allowing any server-side invocation of 4GL code to be done in place, without a client process associated with the session. This would enable usage for appserver calls.

I presume this may also require a way to quickly/easily establish a security context and start a legacy "session" (initialize the transaction support and so forth).

#5 - 04/09/2020 07:02 AM - Greg Shah

- Related to Feature #3931: single sign-on for virtual desktop mode added

#7 - 09/24/2020 10:00 AM - Greg Shah

- Related to Feature #4912: move UI portions of the web client to the server-side added

#8 - 03/02/2021 10:18 AM - Greg Shah

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

#9 - 06/02/2022 10:50 AM - Greg Shah

From email by Constantin:

> I'm sorry, but there was a misunderstanding on my part for [#4406](#) - I read that as 'execution without agent's FWD client', to remove the FWD client from the agent (as there is no need for a FWD client if there is nothing to execute on it).
>
> After reading that correctly, things are a little more tricky, as making the REST worker thread to act like an agent is really dependent on truly client-less agent support. I haven't really thought about how to implement it.

#10 - 06/02/2022 10:53 AM - Greg Shah

Eliminating the client is one step on the path to the complete solution here. It is mostly enabled by [#4065](#), but we would need to change how we startup such client-less agents. That extra work would be in this task.

Later, we could take the next step to implement the "in-place" execution of legacy code from the REST handling thread. It is not clear to me what the savings in performance will be, I'm interested in your opinion here.

#11 - 08/01/2022 08:04 AM - Greg Shah

- Related to Feature #6646: short circuit legacy REST program invocation added