Runtime Infrastructure - Feature #11767

add a monitoring tool in FWD to check for lock contention

08/20/2026 05:05 AM - Constantin Asofiei

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

History

#3 - 08/20/2026 05:08 AM - Constantin Asofiei

In #11764 and #11521, we've found cases of lock contention. Currently, we rely on manually getting thread dumps and analyzing them.

The idea of this task is to add a tool in the FWD server which, when enabled:

- will get thread dump state on a specific period (i.e. 1 second)
- analyze the thread dumps for dead-locks, lock contention, and any other cases where we can find performance problems.

This will help finding issues for load-testing scenarios.

#4 - 08/20/2026 05:41 AM - Şerban Bursuc

FYI, the thread dumps I have given in #11521 besides when specifically requested for TemporaryAccount are all done **automatically** by Lyra.

The way Lyra takes thread dumps is:

1. Monitor each client step. If the step is sent from Lyra but doesn't receive a response from FWD/browser in a set amount of time, then the client is assumed hung and thread dumps are taken. This is currently 8s and is configurable. This is triggered all the time during spawn time, as this is the slowest step in the load tests thus far.
2. When a client fails it automatically takes a thread dump. This is distinct from the above, the above only triggers when there's no expected timeout to wait.

There is an issue with this, because in a 300 client run at spawn time the system is slow, 300 clients will try to take a thread dump which breaks jstack, especially inside Docker, the command usually times out or throws an error. Lyra uses a lock first client wins and adds a global lock - others cannot take a thread dump if there is a lock active and this step is skipped.

For client thread dumps, Lyra looks for the generated log file if available, as it has both the user which is known in the test and the client PID in the name. If the log file paths aren't provided, then it tracks current Java processes and counts the new PIDs and tries to map them to their respective virtual user. This is a highly inconsistent method, as the order can be off. Another issue with the log path is that the log files might not appear in time.

This entire thread dump collection phase is optional and is gated through a CFG file, so it can always be disabled if FWD will officially support this.

#5 - 08/20/2026 08:45 AM - Greg Shah

If we expose this tool via a REST Admin API, that gives us lots of flexibility for how we use it.