Threading and the Java Agent

Related pages:

- Constructor Mode Thread Tracing
- Executor Mode Thread Tracing (Experimental)

Introduction

A pivotal part of the functionality of the Java agent is tracing activity within the JVM such that all the activity (both internal and external to the JVM) that results from servicing any given inbound request can be associated with the request itself. In simple cases, each request is processed by one Java thread, making it only necessary to track the processing of that single thread, in a pattern widely known as 'thread per request'.

This was the predominant case for JEE applications prior to the introduction of version 3.0 of the servlet specification (introduced in Java EE 6). In systems where a significant amount of time is spent waiting for external resources, the thread per request model can be seen as inefficient since many application threads spend most of their lifetime blocked awaiting external responses.

This observation motivated the asynchronous servlet capabilities provided as part of servlet 3 and also motivates many of the increasingly prevalent reactive frameworks that are in common usage (for example, Play, Akka, Vert.x and Spring Reactor). To correctly correlate transactions end to end in applications using this architectural approach, the agent must follow transactions from thread to thread to enable it to measure and link all the elements of work performed on behalf of each request, irrespective of which thread that work is executed on.

The agent provides two mechanisms for thread tracking, known as 'Constructor Mode' and 'Executor Mode' (a new *Experimental Feature* added in Java agent release 4.5.11)

The following pages describe these two alternative approaches to correlation of cross-thread transaction activity:

- Constructor Mode Thread Tracing
  This is the mode that the agent has used for several years. It instruments Threads, Runnables and Callables based on their creation.
- Executor Mode Thread Tracing (Experimental)
  This new experimental tracking mode is based on instrumenting executor frameworks which are commonly used by the majority of applications and frameworks. This mode is expected to make agent operation more efficient, especially in cases where heavy use of threading is in place, for example, in the presence of the above mentioned reactive frameworks.

Choosing the Thread Correlation Mode

The thread correlation mode used by the agent is selected using the `async-instrumentation-strategy` property within the `app-agent-config.xml` configuration file. To use the new (*Experimental*), executor-based instrumentation, modify `app-agent-config.xml` by changing the value `async-instrumentation-strategy` property from `constructor` to `executor` before starting the agent. You can also set the instrumentation strategy by setting the `appdynamics.async.instrumentation.strategy` system property on the JVM command line. For example: `-Dappdynamics.async.instrumentation.strategy=executor`

The system property takes precedence over the configuration file. The thread instrumentation mode that is used cannot be changed at runtime. Threading behavior is tracked using Constructor Mode by default.