

CF911: Stack Tracing CFML Requests to Solve Problems (Presentation Notes)

Charlie Arehart, charlie@carehart.org

Definitions:

- **Running Request:** a CFML page request that is currently executing
- **Thread:** Java executes a running request on a thread
 - There are many threads in Java that besides those for running requests (RMI Reaper threads, Object Skimmer threads, various daemons, etc.)
- **Stack Trace:** java-provided feature to identify specific java method being executed at a point in time for a thread (running request)
 - In CF/Railo/BlueDragon, the stack trace also shows what line of CFML source was running, if any was executing at the time of the stack trace
 - Each of the three primary CF server monitors provide a feature to obtain a stack trace
 - There are also other tools that can obtain stack traces, mentioned below
- **Thread Dump:** a list of all running threads and the stack trace of each (at that point in time)
 - Each of the three primary CF server monitors provide a feature to obtain a thread dump
 - There are also other tools that can obtain thread dumps, mentioned below
- **Automatic email of thread dump:** some problems happen when you're not online to use tools
 - So all three primary CF monitors offer a feature to send email notifications when trouble conditions are detected, including a thread dump

Alternative Stack Tracing Tools:

- **CF Enterprise Server Monitor** (built into CF 8/9 Enterprise)
 - Accessible from CF Admin (does not work on CF 8/9 Standard or any edition before CF 8)
 - Double-click a running request. Stack trace section in middle of request detail page
 - Note: you must enable "start monitoring" to first even see running requests
 - And you must enable "Start Profiling" to see stack tracing of running requests
 - Thread dump available via "SnapShot" feature
 - Automatic email of thread dump offered via "Alerts" feature
- **FusionReactor** (commercial CF/JEE Server Monitor, <http://www.fusion-reactor.com/>)
 - Works with any version of CF 6-9, whether Standard or Enterprise, and also any JEE server
 - Click magnifying glass icon to left of running request to obtain color-coded stack trace

- Thread dump available via “Resources>List All Threads” feature
- Automatic email of thread dump offered via “Crash Protection” feature
- More on FR Stack Tracing at <http://www.fusion-reactor.com/fr/featurefocus/stacktrace.cfm>
- **SeeFusion** (commercial CF Server Monitor, <http://www.seefusion.com/>)
 - Works with any version of CF 6-9, whether Standard or Enterprise
 - Click paper pile icon to left of running request to obtain stack trace
 - Thread dump available via “Stack” menu option, then “Trace” feature
 - Automatic email of thread dump offered via “Configuration” menu option, then “Active Monitoring Rules” feature
- **Other stack tracing tools**
 - **SeeStack** (<http://www.seefusion.com/seestack/seestack.cfm>), a free web-based tool to analyzes thread dumps to identify running CF requests within thread dump
 - see “Resources” section below for other stack tracing tools

Other Tools Mentioned:

- **CFSTAT**: command line tool built into CF Standard and CF Enterprise Server deployment
 - Can show queued requests (once all available “simultaneous requests threads” are busy, new requests will be queued, and this tracks how many are queued. None of the monitors track queued requests)
 - Available at [cf]\bin directory
 - Note:
 - Must enable in CF Admin Debugging page
 - Does not function in CF Enterprise Multiserver (multiple instances) or J2EE deployment
 - Can also see same information using CF getmetricdata(“perf_monitor”) function and Windows Performance Monitor/System Monitor
 - with same notes above

Resources

- **Obtaining thread dump in CF without the server monitors above**
 - “Debugging thread dumps and server problems in ColdFusion MX 6.1 and 7.0”
 - http://kb2.adobe.com/cps/183/tn_18339.html
 - Requires starting CF from command line, using command-line ctrl-break to generate thread dump
 - Applies CF 8, 9 as well
 - Particularly valuable for those not running CF Enterprise, who can't use CF Server

Monitor

- again, though, those people can use FusionReactor, SeeFusion
- Technnote also offers additional useful information on interpreting stack traces
- See also “*Obtaining a Thread Dump with ColdFusion or JRun running as a Windows Service*”, blog entry by Brandon Purcell
 - <http://www.bpurcell.org/blog/index.cfm?mode=entry&entry=1062>
- **Using JStack and other java tools to obtain stack traces**
 - Several are built into Java, and have improved from Java 1.4 to 1.5 and 1.6
 - These require installing Java SDK (separate from the JVM built into CF)
 - Some will work only if CF is running under same account as the Java tools being executed
 - some will also work only if CF is started from command line, unless CF is configured for remote
 - More info:
 - “*JVM Monitoring Tools For ColdFusion*”, blog entry by James Netherton
 - <http://www.jamesnetherton.com/blog/2008/05/16/JVM-Monitoring-Tools-For-ColdFusion/>
 - “*Using JDK tools to analyze ColdFusion hangs*”, blog entry by Jochem Van Dietem
 - <http://jochem.vandieten.net/2008/12/03/using-jdk-tools-to-analyze-coldfusion-hangs/>
 - “*Finding memory leaks in your ColdFusion JVM*”, blog entry by Brian Ghidinelli
 - <http://www.ghidinelli.com/2009/07/16/finding-memory-leaks-coldfusion-jvm>
 - “*Using hprof with Coldfusion to troubleshoot slow code issues*”, blog entry by Chris Peterson
 - <http://www.alagad.com/blog/post.cfm/using-hprof-with-coldfusion-to-troubleshoot-slow-code-issues>
 - “*Troubleshooting ColdFusion Performance: Analysis Part II*”, blog entry by Chris Peterson
 - <http://www.alagad.com/blog/post.cfm/troubleshooting-coldfusion-performance-analysis-part-ii>
 - VisualVM, visual tool integrating several commandline JDK tools
 - <https://visualvm.dev.java.net/>
 - to name just a few
- **CFWatcher 2.1**: Another tool offering stack tracing of CF requests
 - <http://www.1smartsolution.com/>