CONSIDERING MONITORING SOLUTIONS FOR CF AND LUCEE

Charlie Arehart
Independent Consultant/Server Troubleshooter
CArehart.org (@carehart)

Updated Apr 25, 2022
A DAY IN THE LIFE
CHALLENGE: CHOOSING THE RIGHT TOOLS
WHAT'S NOT THE RIGHT TOOL?
WHO AM I?
COMMON PERF PROBLEMS OBSERVED

- High CPU
- High memory
- Hanging requests
- Site(s) unavailable
- CF/Lucee Crashing
- CF/Lucee slow to stop/start
- And more
CAUSES THAT MANY SUSPECT

- Code inefficiency
- SQL inefficiency
- GC inefficiency
- Need of JVM “tuning”
- CF itself is slow
- CF/Lucee needs to be restarted
- And more
LESS-EXPECTED CAUSES

- Config issues
- External calls hanging
- DB connection pool hanging
- Unexpected traffic volume/nature
- Sudden, excessive GCs
- Web server connector issues
- CF engine / JVM bugs
- And more
LESS-NOTICED PROBLEMS

- CFThreads performing poorly
- CF/Lucee crashing frequently
- End-user slowness, not on server
- Crazy high session count
- Client var repo purging
- Unexpected low or no traffic
- And more
CAUSES OUTSIDE OF CF/LUCEE

- DBs called
- APIs called
- Other processes on server
- Disk or Network Issues
- Problem in VM host, other guests
- And more
THE CHALLENGE OF SCALE
FOUR CLASSES OF TOOLS TO HELP

- CF/Lucee Troubleshooting Features
- CF/Lucee Monitoring Tools
- Java Troubleshooting Approaches
- Application Performance Monitors (APMs)
- Other System Monitors
CF/LUCEE TROUBLESHOOTING FEATURES
CF/Lucee Monitoring Tools

- CF Server Monitor (CF8-2016)
- SeeFusion
- CF PMT (CF2018 & up)
- FusionReactor
THEN THERE ARE JAVA TOOLS…

- Built-in or freely obtained GUI tools
- JMX
- Tomcat Manager
- 3rd party tools
- Java APMs
JAVA TROUBLESHOOTING APPROACHES

- Stack Traces
- Thread dumps
- Profiling
- Heap dumps
BUILT-IN JAVA TOOLS

- jcmd
- jmap
- jvisualvm / visualvm
- Java Flight Recorder (JFR)
- jstack
- jconsole
- Java Mission Control / jmc
Again, jstack, visualvm, etc.
FR, SeeFusion, CF PMT
Some APMs (requires reaching into running JVM)
Taking thread dumps
- Again: VisualVM, Java Mission Control, Tomcat Manager, jcmd, jstack
- And within most APM tools (as well as FR, SF, CFSM/PMT), and other ways

Analyzing thread dumps (free, online services/tools)
- fastthread.io
- jstack.review
- spotify.github.io/threaddump-analyzer/
- github.com/irockel/tda

More help: baeldung.com/java-analyze-thread-dumps
These monitor time spent within or across threads, by java method

Available in VisualVM and Java Mission Control, as well as:

- Some APMs (as well as FR, SF, CFSM/PMT)
- Yourkit: yourkit.com
- Jprofiler: ej-technologies.com
- Xrebel: xrebel.com
- Netbeans: netbeans.apache.org

More help: baeldung.com/java-profilers

PROFILING TOOLS
Taking heap dumps
- VisualVM, Java Mission Control, Tomcat Manager
- jcmd, jmap, jmx, HeapDumpOnOutOfMemoryError
- And most APM tools (including FR, PMT)
- Lucee also has a Heap Dump extension

Analyzing them
- Eclipse memory analyzer tool (eclipse.org/mat)
- heaphero.io
- jxray.com

More info: dzone.com/articles/how-to-capture-java-heap-dumps-7-options
Forcing GCs: jvisualVM, jcmd, most APM tools (and FR, SF, CFSM/PMT)

Obtaining GC logs: -XX:+PrintGCDetails and related args

Tools to analyze GC logs
  - gceasy.io
  - gcplot.com
  - github.com/chewiebug/GCViewer
  - And others
JMX TOOLS

- jmx: java mgt extensions
  - Many java apps, including Tomcat, expose metrics via jmx
- Tools to view them include:
  - jvisualVM, Tomcat Manager
  - jmc, jconsole
  - FusionReactor
  - Some APM tools
  - jolokia.org
JAVA APMS: MANY, MANY

- Appdynamics
- Dynatrace
- NewRelic
- Datadog
- Glowroot
- Solarwinds AppOptics
- Sumo Logic

- Stackify (prefix and retrace)
- JavaMelody
- Perfino
- Sematext
- MoSKito
- Scouter
- and still others!
SOME NON-APM JAVA TOOLS OF NOTE

- yCrash (Java Root Cause Analyzer)
- Overops (app observability/root cause analysis)
- Akamas (autonomous JVM performance optimization)
OTHER FEATURES TO WATCH FOR

- FR and some APMs export their data to cloud monitoring/telemetry
  - Vital when managing many instances, containers
- FR/SF/PMT as well as some APM’s also offer:
  - Alerting about trouble
  - Error tracking
- And FR/PMT and some APM’s also offer:
  - Tracking end-user response time
  - Tracking sessions
  - Reporting
OTHER BENEFITS OF MONITORING

- Use to track baselines, then changes over time, such as due to:
  - Code changes
  - CF/Lucee admin
  - JVM config changes
  - Changing volume/nature of traffic/load
FINALLY, IMPORTANT TO MONITOR OTHER PARTS OF ENVIRONMENT (CF411.COM/MON)
We’ve seen many problems, approaches, tools
Still, even basic tools (task mgr/top, cfstat/perfmon, etc.) better than nothing
But often need to see inside CF/CFML processing: requests, queries, more
And Java tools can help (or not), but you maybe learned of new ones
Get and use SOME diagnostics. Don’t just “restart” CF/Lucee! 😊
For more, on this topic or other troubleshooting:
  - charlie@carehart.org
  - @carehart (on twitter, linkedin, github, facebook, etc.)