INTO THE BOX 2024

THE NEW ERA OF MODERN DEVELOPMENT

BLUE ROOM

Demonstrating Monitoring Solutions for CF and Lucee (and BoxLang!)

PRESENTED BY

Charlie Arehart

Ortus Solutions
A Software Revolution

“Wha’ happened?”

- Nearly all of us face challenges at some time with CF, Lucee, etc. whether on:
  - Our dev environment (whether running on commandbox or not), or
  - a server, or a cluster of servers, or
  - vm’s, or
  - containers, etc.
- Regardless of platform or architecture, this can be challenging
  - Is the problem at the OS level? Or some component (cpu, memory, disk, network)
  - Is it in CF or Lucee? Some setting? Our code? A bug?
  - Is it about our traffic? Whether expected or not, unusual or not?

We need SOMETHING to observe

- There are a range of alternatives to watch what's going on
  - At each of those levels
  - Whether built-in or as can be added (some free, some paid)
- My 2021 ITB preso covered those in more detail
  - What the options are, which are included/how to add them, considerations
  - Common causes of problems, more
- In this talk, I just want to demonstrate many of the alternatives
  - You can go view the other preso for the details (nearly all are unchanged)
  - *carehart.org/presentations/#mon_consider*

Me.about()

- I focus on CF/Lucee server troubleshooting, as an independent consultant
  - Assist organizations of all sizes, experience levels
  - Work remotely: safe, secure, easy via **shared desktop** (zoom, meet, teams, etc.)
  - **Solve most problems in less than an hour**, teaching you also as we go
  - **Satisfaction guaranteed**
  - More on rates, approach, online calendar, etc at *carehart.org/consulting*
- But to be clear, I'm not selling anything in this session! ☺
  - Just sharing my experience, and that of others
- Presentation online at *carehart.org/presentations*

Topics

- CF/Lucee monitoring features
  - Simple things built-in
  - Alternative monitoring tools
- Java monitoring features
  - Java command line tools, Ui's, and more (built-in or freely added)
  - Alternative monitoring tools (APM's)
- OS monitoring features
  - Simple—and more capable—built-in tools
  - Alternative monitoring tools
- Other monitoring tools to consider

A comment…

- Different people may view some tools/features here as being for:
  - Troubleshooting
  - Debugging
  - Observability (o11y)
- Keep this point in mind when searching for info on them
  - One person's "debugging" is another's "troubleshooting"
  - And one person's "monitoring" is another's "observability"
  - In this talk, I'm not going to dwell on the distinctions
    - And not covering step debugging at all (deserves its own talk)

I'll default showing most of my demos…

- …in Windows
  - I know the Mac and Linux folks will grumble
  - But you're used to translating or finding things
    - Indeed many things you find are WRITTEN more for *nix folks
  - Windows folks often are left feeling like this:
    - https://www.youtube.com/embed/TSse5b3wDe4

Questions we should be able to answer

- Is CF/Lucee/boxlang running?
- What are key resource usage metrics (cpu, memory, etc.)?
- How many requests are running/have recently run?
  - What are their key characteristics?
  - Are any hung? Slow? ***WHY?***
- Same with respect to queries, cfhttp calls, etc.
- Are errors happening? At what rate? Why happening?
- And more

Simple things built-into CF/Lucee

- Lucee and CF both offer logs of course
  - CF offers a metrics.log, if enabled
- And both offer request debug output (as enabled in CF or Lucee Admin)
  - Better than nothing, but not suited to production (or viewing requests of others)
- Lucee Admin shows a bit of monitoring
  - Heap use within Lucee instance
  - Cpu within Lucee instance
  - Count of requests (running and queued), threads
  - Counts of memory scopes used
  - Count of datasource connections
- Demo…

Simple things built-into CF/Lucee (cont.)

- Consider also:
  - *getmetricdata()* function (CF and Lucee)
  - cfstat cli script in CF (cfusion/bin directory)
  - Lucee *getmemoryusage()* function, about JVM heap memory
  - Also Lucee *get*memory()* functions, about system memory
- CF outputs metrics to Windows Performance Monitor
- Of course, these work with CF/Lucee however they are deployed

Extending CF/Lucee for monitoring

- Lucee Performance Analyzer extension (*github.com/zspitzer/lucee-performance-analyzer*)
  - Requires debugging be enabled in Lucee Admin
- Demo…
- CF or Lucee:
  - *github.com/foundeo/cfmetrics*
- ColdBox Debugger (*forgebox.io/view/cbdebugger*)

- And there are more-complete monitoring solutions for CF/Lucee…

FusionReactor

- Powerful tool with many capabilities
  - Can work entirely within JVM (“on-prem”)
    - or also pushing data to cloud (“FR Cloud”), separately priced
  - Easily added into FusionReactor: *forgebox.io/view/commandbox-fusionreactor*
- Could do entire talk on FR, and indeed I have done many
  - Just last month: *carehart.org/presentations/#cf_fr_opt*
  - My series: *youtube.com/playlist?list=PLG2EHzEbhy09al6Lx0JlIUcpbA8UrwPuo*
- Works with CF or Lucee (or BoxLang!)
  - Commercial, with available developer license (*fusion-reactor.com*)
- Quick demo…

SeeFusion

- Came out about same time as FR mid-2000's
  - Offers many basic/key monitoring features
  - Not as capable as FR, but has a new advantage…
- Is no longer commercial: **now open source**, *github.com/seefusion/seefusion*
- Works with both CF and Lucee
- Must implement either as Tomcat valve or servlet filter (vs javaagent)
  - Must "wrap" datasources to monitor SQL/JDBC
- CFMeetup session on SeeFusion: *youtube.com/watch?v=QKpTtpCc0qE*

PMT

- Available free in CF2018 and above
  - Adobe's replacement for the CF Server Monitor in CF8-2016
- Offers many capabilities like FR (and each has things the other doesn't)
  - 20-min intro video: *youtube.com/watch?v=ycydtKHi_AI*
- While the tool is powerful, the take up in the community has been slow
  - Due partly to its architecture: PMT and PMT Datastore are apps separate from CF
  - Can be run on same machine as CF or another (even separate from each other)
  - Tip: Can use FR to monitor PMT and Datastore, as they are java apps!
- Quick demo of PMT…
- (One thing that PMT and FR can do—uniquely--is track cfthread processing)

- But sometimes CF/Lucee tools can't be added or aren't sufficient
  - Of course, CF and Lucee (and BoxLang) run on Java
  - And there are several Java tools we can look to…

Java/JVM command line tools

- Several commands, each having many subfeatures, in JDK versions of Java
  - Java 8 and earlier offered option of JRE or JDK, later java versions are JDK only
  - Tools located in Java's bin directory (such as the Java used to run CF/Lucee/BoxLang)
- For local development instances, tools should just work with them (even running as service)
  - As long as user running jvm tool is same as user running CF/Lucee/BoxLang
  - On Windows, if CF/Lucee runs as service under "LocalSystem" account, run cmdline as admin
- What if instance is not local, or user running tools is not same as user running CF/Lucee/etc?
  - Then you need to specify JMXremote args in startup of CF/Lucee/BoxLang
  - Plenty of resources online cover doing that, for CF and Lucee (and java in general)
  - See also optional related tool called jstatd
- These jvm tools work with CF/Lucee/BoxLang regardless of how deployed

Java command line tools (cont.)

- They’re run by pointing to the OS process id of the running java/CF/Lucee/etc
  - Can try to find with *jcmd* (one of those java commands), with no args
    - That will list those java processes YOU started
    - (If CF/Lucee/etc is running as a service (Windows or *nix), won’t show those)
- Can use OS commands to find pid of running processes
  - Windows: task manager (look for coldfusion.exe, not coldfusionsvc.exe), or
    - *tasklist* | *findstr coldfusion* (CF typical install)
    - *tasklist* | *findstr tomcat* (Lucee typical install)
    - *tasklist* | *findstr java* (Commandbox running CF/Lucee/BoxLang)
  - Linux: *pgrep java* (CF and Lucee always show in Linux as “java” processes)

Java jcmd

- Can do many things with this tool
  - Can get list of features by using *jcmd <processid>*
- Some examples (assuming CF/Lucee/etc is process id 123)
  - *jcmd 123 VM.version* (args are case-sensitive)
  - *jcmd 123 VM.uptime* (how long jvm/CF/Lucee/etc has been up)
  - *jcmd 123 GC.heap_info* (memory space info)
  - *jcmd 123 GC.run* (request garbage collection)
  - And more …

Obtaining thread dumps/stack traces

- Thread dumps identify what running requests (and other threads) are doing
- FusionReactor, PMT, and SeeFusion offer means to create them
- jcmd offers another way to obtain them
  - *jcmd <pid> Thread.print* (request thread dump)
- As does:
  - *jstack <pid>*
- And we'll see other visual jvm tools that can create/view thread dumps
- Demos…

- For now, on to other java cli tools…

Other Java CLI monitoring tools

- *jdb, jinfo, jmap, jps, jstack, jstat*
  - I'll leave you to explore those as may interest you
- *jhat* and hprof were removed with Java 9
  - Capabilities now in other tools already mentioned
- Again, *jstatd* allows allowing remote calls into a host running a JVM
  - Including setting it up via ssh
  - Will leave that for you to explore if interested
- For now, let's wrap up with one powerful (and rather new) jvm CLI tool...

JFR (Java flight recorder)

- JFR is a feature that's been around for a while, can be very powerful
  - Was for a time an Oracle commercial feature, but is now free
- Can create jfr recordings two ways
  - Use *jfr* tool to launch a JVM/app or use *jcmd <pid> JFR.start*
    - Can pass it args to control WHAT it records, which is where power comes in
    - Available args also to control saving/viewing JFR info at consle
  - Will see GUI tools to create/view JFR recordings soon

- Indeed, while CLI tools can be powerful, many prefer GUI tools for many monitoring tasks
  - And there are built-in or easily obtained GUI jvm tools…

jconsole

- First of 3 available java GUI tools (least capable of the 3)
  - But better than nothing, since it's built-in to Java
  - Like CLI tools, found in JDK's *bin* folder
- Demo…

- If on headless Linux or debugging a remote machine, use JMXRemoting
- More about JMX: Java management extensions
  - Both a way the JVM exposes metric data to tools, and an API to dig into more
  - Viewing JMX info supported by this tool and next two, and also by FusionReactor

visualvm , formerly *jvisualvm*

- Has also gone through transition of licensing over years, now free/open source
  - *visualvm.github.io*
- Can do many things in its UI, while watching a JVM process (CF/Lucee/etc.)
  - obtain thread dumps, heap dumps/analyze them
  - profile memory and cpu use (across all threads in the process)
  - view JMX info
  - **read and graph JFR recordings(!)**
  - and more
- Has community of plugins offering still more
- Demo…

Java Mission Control (JMC)

- Has also gone through transition of licensing over years, now free/open source
  - *wiki.openjdk.org/display/jmc/Main*
  - *docs.oracle.com/en/java/java-components/jdk-mission-control/*
- Can do many of same things as visualvm
  - Including taking thread dumps, processing JFR recordings, showing JMX info
  - Seems to REQUIRE enabling of JMX (port being exposed by CF/Lucee/etc)
- Download:
  - JMC 8: *jdk.java.net/jmc/*
  - JMC 9: *oracle.com/java/technologies/jdk-mission-control.html*
  - JMC 9 requires Java 17, JMC 8.1 requires Java 11

Related Java monitoring tools

- I've mentioned how JVM (and some CF tools) can create thread dumps
- Check out helpful free online tool to analyze them:
  - *fastthread.io*
- And free sister online tools:
- *heaphero.io*
- *gceasy.io*
- *ycrash.io*

- Still more java monitoring tools
  - See my list at *cf411.com/javamon*
- But such built-in Java monitoring tools have more capable cousins…

Java APMs

- There's a whole world of Java Application Performance Monitoring (and "observability") tools, which can be added to CF/Lucee/etc
  - Across all platforms
  - Some free, some commercial
- Like other Java tools, often these APMs don't easily track page requests
  - And they often present low-level java info, hard to relate back to CF/Lucee/etc
- No time to demo them in this talk
  - See my list at *cf411.com/apm*

- Note that some Java tools and APMs include "system" monitoring features
  - Indeed, often problems in our apps really reflect larger system problems
  - Several kinds of tools focus on that…

Simple OS-level tools

- Windows
  - Task Manager (see *Processes* and *Details*, sortable; and *Performance*)
  - *tasklist* (command)
  - Bonus: *systeminfo* | *findstr "Boot time"*
- Linux
  - *top* (can sort using “f” key)
    - Consider installable variants: *htop*, *atop, vtop, gtop*, etc.
  - *ps -aux* (try also *-auxf*)
  - Bonus: *uptime*
- MacOS
  - Activity Monitor (GUI)
  - Also Linux (Unix) command line tools above

More capable OS tools, built-in

- Windows
  - Resource Monitor
  - Performance Monitor (tip: switch “graph type” to “report”)
- Linux
  - *sar* (collect system activity over time)
  - *netstat* and *ss -tua* (network stats)
  - Consider adding
    - *iostat* (cpu load, disk activity), *iotop* (still more disk i/o detail)
    - *iftop* (bandwidth usage on an interface by host)

System monitoring tools, as add-ons

- As with APMs, a whole world of OS monitoring tools available
  - Across all platforms
  - Again, some free, some commercial
- See my list at *cf411.com/sysmon*

- Finally, some problems are specific to other aspects of processing…

Other monitoring tools

- There are also tools for monitoring your:
  - Web server
  - Database
  - Network
  - IO
  - And more
- I track tools like these and still more at *cf411.com/dbmon*
- And cloud platforms (AWS, Azure, GCP) offer monitoring for their components

Finding more

- I’ve held off offering links to more on most of these things
  - Most can be found readily via web searching
  - Do beware that some resources are old (and may no longer be accurate)
- Again, the PDF of this talk is already online on my site
  - And the recordings of these ITB sessions will be offered online soon
- I’ve also pointed to related presentations I’ve done (usually also recorded)
- Certainly feel free to reach out to me if you have trouble finding a resource

Summary

- Problems will happen with your CF/Lucee instances
  - Don’t regard them as a black box, or throw darts trying to “fix” things
- We’ve seen there are built-in tools, or tools easily added, to better understand:
  - CF/Lucee monitoring capabilities and options
  - Java monitoring features and options
  - OS monitoring features and options
  - Monitoring other components/aspects of architecture
- I hope you feel more empowered to dig into such problems on your own
  - If you hit snags/want guiding hand, am available for remote, screenshare consulting
  - Can work in as short as 15-min intervals; solve most problems in less than an hour

INTO THE BOX 2024

THANK YOU TO OUR SPONSORS

Ortus Solutions
A Software Revolution

TeraTech
The ColdFusion Experts
Develop | Secure | Optimize

media3
secure ▪ managed ▪ cloud

Lucid Outsourcing Solutions
Intelligent Transparent

COMPUTER KNOW HOW