TOMCAT MONITORING ALTERNATIVES: PROS AND CONS

Charlie Arehart
Independent Consultant/Server Troubleshooter
(CArehart.org, @carehart)
A DAY IN THE LIFE
WHO MIGHT YOU BE?
CHOOSING YOUR TOOLS
WHO AM I?
What are some of the common troubleshooting **approaches** you have used?

- Not asking about **tools**, for now
COMMON TROUBLESHOOTING APPROACHES

- Logs (Tomcat, GC, etc)
- Thread dumps/Stack traces
- Heap dumps
- Core dumps
- JMX
- Profiling
- Snapshots
What are some of the typical tools you have used?
TYPICAL TOOLS

- Built-in JVM tools
  - Thread dump/viewing, heap dump/viewing
  - Memory space monitoring
  - GC timing/triggering
  - Thread profiling
  - JMX monitoring, manipulation
  - And more

- Tomcat Manager
  - Can help with some of same, and more

- 3rd party JVM tools
  - Often focused on one of these aspects

- Log analysis tools

- APM tools
What are some of the common problems you have faced?
Common Problems?

- High CPU
- Hanging requests
- External calls hanging
- And more
- High memory
- DB calls hanging
- Tomcat Crashing
LESS COMMON PROBLEMS

Problems when no traffic
End-user time slow, not on server
Unexpected traffic
Tomcat crashing frequently/unexpectedly
Web server issues
Web connector issues
And more
SYSTEM RESOURCE ISSUES

- CPU, Memory, Disk, Network
  - On Tomcat box
  - On boxes of any related services (db, cache, etc.)
- Special issues with VMs
  - Issues on the VM host
  - Issues with other VMs within host
THE CHALLENGE OF SCALE

Charlie Arehart (CArehart.org, @carehart)
Not trying to scare or overwhelm you

Just saying there’s a place for various tools
  - And not all tools are best suited to solving all problems

May seem obvious
  - But again my experience suggests otherwise

SCARED AWAY? NOT MY GOAL

Charlie Arehart (CArehart.org, @carehart)
DEMO TIME
There are so many tools in the java and tomcat monitoring space
  - Can’t show them all
- Let’s look at a bit more about most of them
  - In same order originally introduced before

NO TIME TO DEMO ALL POSSIBLE TOOLS
Manager available but **not enabled** by default

- Take care to consider **security concerns** with enabling it

**Monitoring-oriented features**

- Show server status, observe sessions, help with memory leaks, trigger thread dumps,

**Available REST, UI, and ANT automation**

MORE ON JVM TOOLS

- Tools provided in JDK only (not JRE)
  - Some only as of a certain JVM version (added/removed)
  - Often most easily used if JMX remoting enabled
  - On Windows, may need to “run as admin” for some to work
- jvisualvm / visualvm
  - Removed from JDK as of Java 9, now at https://visualvm.github.io
  - https://docs.oracle.com/javase/8/docs/technotes/guides/visualvm/
- Java Mission Control / jmc
  - Was originally only available to licensees, then open sourced
  - More: http://hirt.se/blog/
  - New builds: http://jdk.java.net/jmc/
  - https://docs.oracle.com/javase/9/tools/jmc.htm
- ...

Charlie Arehart (CArehart.org, @carehart)
MORE ON JVM TOOLS

- As well as ...
- jcmd
  - [https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jcmd.html](https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jcmd.html)
- jconsole
  - [https://docs.oracle.com/javase/8/docs/technotes/guides/management/jconsole.html](https://docs.oracle.com/javase/8/docs/technotes/guides/management/jconsole.html)
- jstack
  - [https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jstack.html](https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jstack.html)
- jmap
  - [https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jmap.html](https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jmap.html)
- jstat
  - [https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jstat.html](https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jstat.html)
VisualVM
Java Mission Control
Yourkit: https://www.yourkit.com/
Jprofiler: https://www.ej-technologies.com/
Xrebel: https://zeroturnaround.com/software/xrebel/
Netbeans: https://profiler.netbeans.org
http://www.spyglassstools.com/products/spyglass-remote-profiler/
And others, including APMs

JVM PROFILERS
Taking thread dumps
- VisualVM, Java Mission Control
- Tomcat Manager
- Jcmd, jstack, kill -3, windows (ctrl+break), threadmxbean, apm tools

Analyzing them
- TDA: https://github.com/irockel/tda
- Samurai: https://github.com/yusuke/samurai
- http://fastthread.io/
- And more, including APMs
Taking heap dumps
- VisualVM, Java Mission Control
- Tomcat Manager
- jcmd, jmap, jmx, HeapDumpOnOutOfMemoryError, apm tools

Analyzing them
- Eclipse memory analyzer tool (MAT)
  - [http://www.eclipse.org/mat/](http://www.eclipse.org/mat/)
- jhat
  - [https://docs.oracle.com/javase/7/docs/technotes/tools/share/jhat.html](https://docs.oracle.com/javase/7/docs/technotes/tools/share/jhat.html)
- [http://heaphero.io](http://heaphero.io)
- [https://plumbri.io](https://plumbri.io)
- [https://go.nastel.com/autopilot-heap-detector](https://go.nastel.com/autopilot-heap-detector)

And more, including APMs
- Tomcat Manager jmxproxy
- Jvisualvm, jmc, jconsole
- Jmxsh: https://github.com/davr/jmxsh
- Jolokia: https://jolokia.org/
- Jmx exporter: https://github.com/prometheus/jmx_exporter
- More: https://docs.oracle.com/javase/8/docs/technotes/guides/jmx/tutorial/tutorialTOC.html
OTHER TOOLS

- Tools to analyze GC logs
  - http://gceasy.io/
  - https://github.com/chewiebug/GCViewer/wiki

- Tools to manage/store/search any logs
  - Elastic stack / ELK (elasticache/logstach/kibana)
    - logz.io
  - Fluentd (Fluentd.org)
  - splunk.com, loggly.com, sumologic.com, humio.com, etc

AVAILABLE APMS

- Appdynamics
- Dynatrace
- NewRelic APM
- Java Melody
- Plumbr
- Appedo
- Perfino

- Sematext
- Stackify (prefix and retrace)
- Moskito
- Hawkular
- Nastel
- FusionReactor
- And more
OTHER POINTS

- Most APMs implemented via javaagent
  - Some tools can auto-attach, or auto-register
- Some require instrumentation of code, but not all do
- Most JMX tools can control frequency of polling
- Safe to monitor production?
  - Some tools are, absolutely. Others can have impact
  - Need to assess each for your self
OTHER APM FEATURES TO WATCH FOR

- Tracking things over time, value of alerting
- Tracking queries within requests
  - Tracking external server call time
  - Some APMs offer this ability
- Tracking errors
- Monitoring from central repository off-server
- …
TRACKING SERVER RESPONSE TIME SUCCESS/Failure OVER TIME
- Some APMs offer daily/weekly/monthly reports

TRACKING END USER RESPONSE TIME
- Some APMs offer js code to add to your app
- Then network time/browser render time are tracked in the APM

TRACKING SESSIONS BUILD UP
- Perhaps due to spiders/bots
- Ways to watch it, ways to address it

OTHER APM FEATURES TO WATCH FOR (CONT.)
CONCLUSION

- Again, profiling, heap dumps, thread dumps and GC logs are one way to go.
- Need also to see what requests are running, have run.
- Will often benefit from tracking over time, alerting, reports.
- Explore APMs and other alternative solutions to find best fit.
- If I can answer any questions about my talk, please contact me:
  - charlie@carehart.org
  - @carehart (twitter, linkedin, github, facebook, etc.)