redhat.

Monitoring Java Application Performance Using Thermostat

Scott Seighman
Solutions Architect
sseighma@redhat.com

September 29, 2017

AGENDA

OHIO LINUX FEST 2017

BRIEF OVERVIEW

INTRODUCTION TO THERMOSTAT

INSTALLATION & SETUP

HOW TO INSTALL

DEMOS AS WE GO ...

THERMOSTAT IN ACTION

RESOURCES

MORE INFO & LINKS

BRIEF OVERVIEW

WHAT INFO DO I WANT/NEED?

Gather

- CPU Usage
- Memory
- Garbage Collection
- Classes
- JIT Behavior
- I/O Calls
- Threads

Change

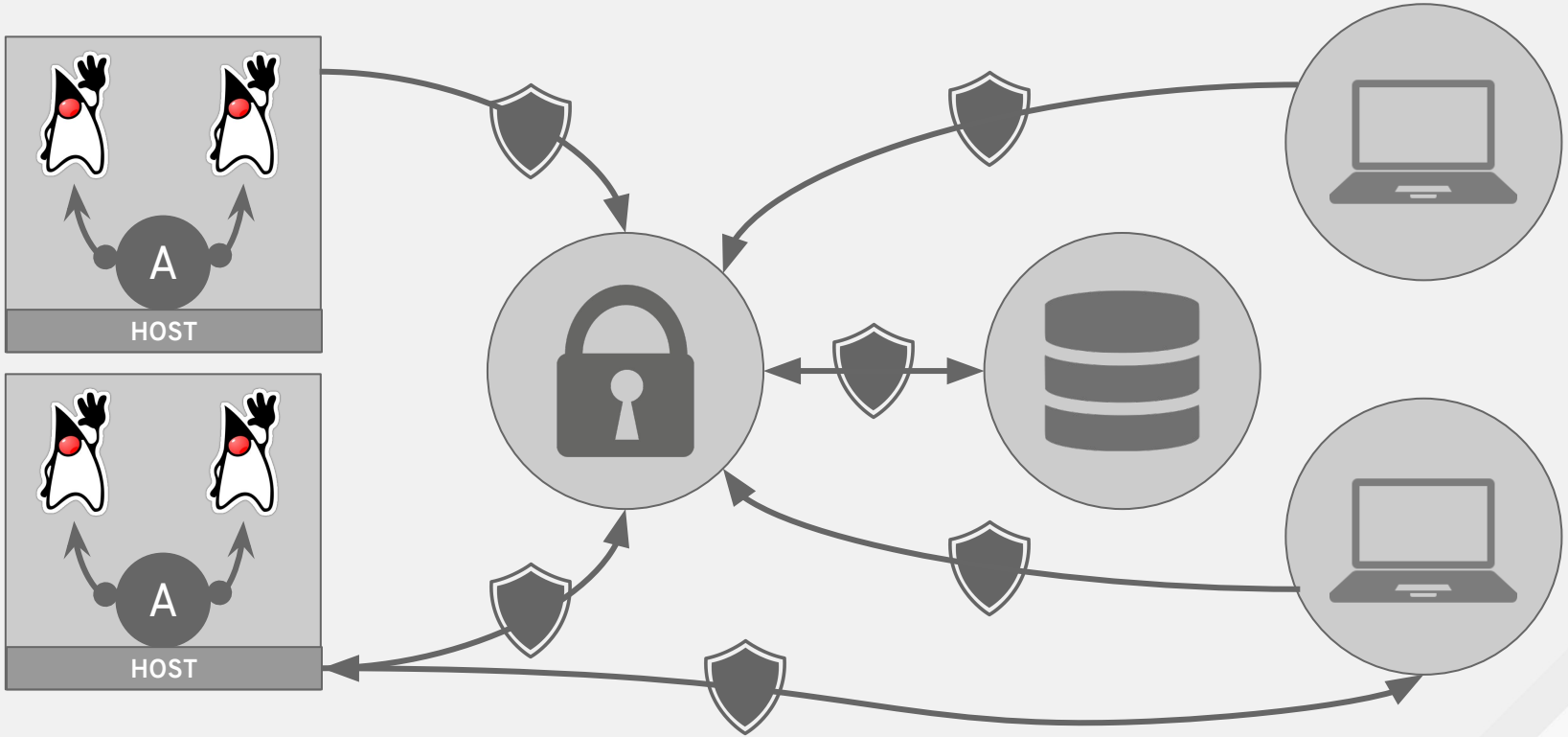
- Heap Dumps
- Invoke Garbage Collector
- Detect Deadlocks
- Inject custom code for on-demand instrumentation

THERMOSTAT OVERVIEW

FEATURES

- Serviceability & Monitoring for OpenJDK
- Single Machine
- Multiple Hosts/JVMs
- Up & Down the Stack
- Historical Information
- Command Line & GUI

THERMOSTAT ARCHITECTURE

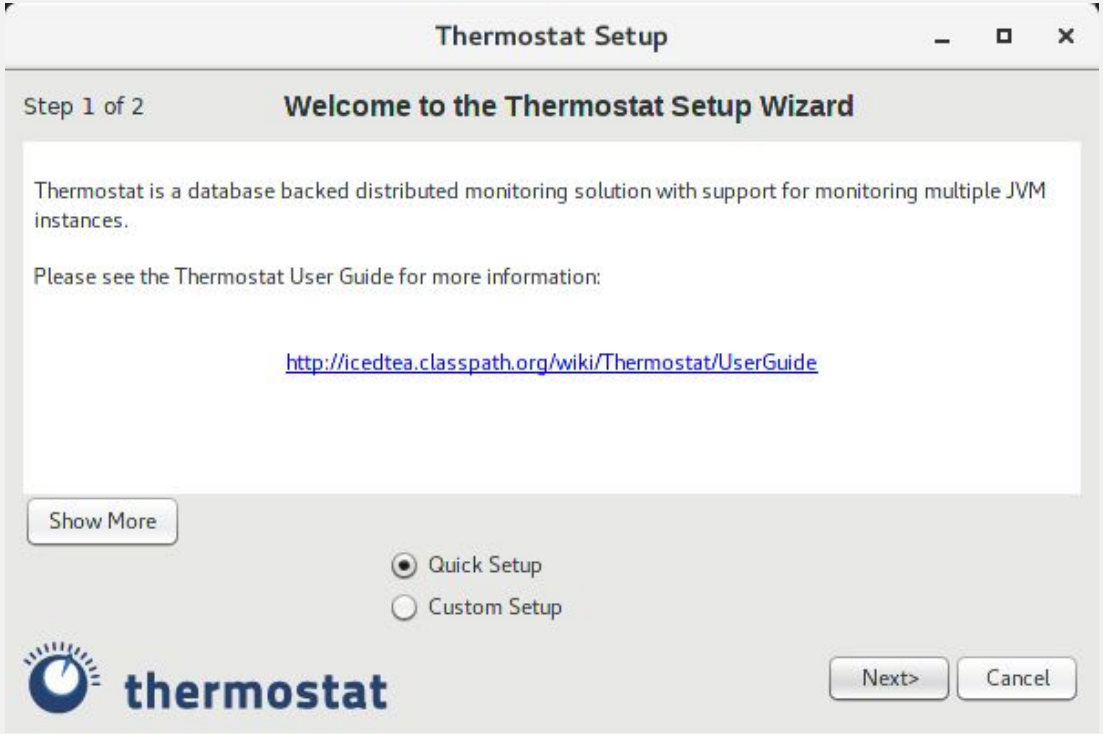


INSTALLATION & SETUP

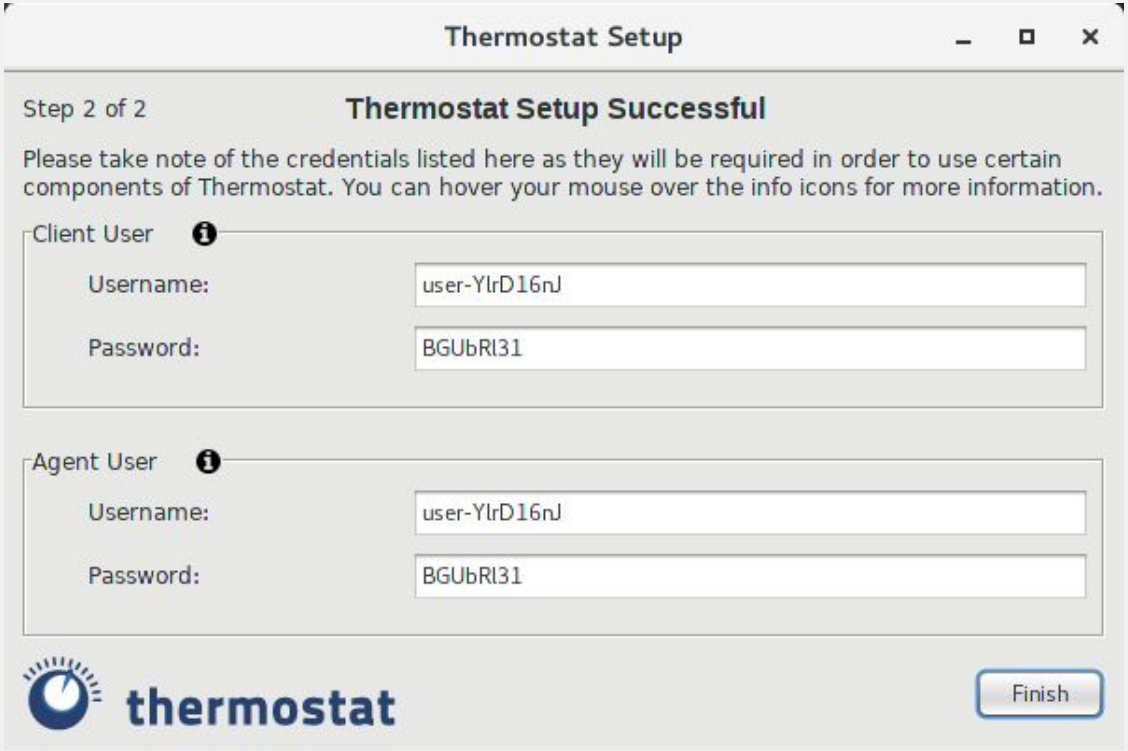
SETUP THERMOSTAT

```
$ thermostat setup
starting storage server...
server listening on ip: mongodb://127.0.0.1:27518
log file is here: /home/sseighma/.thermostat-1.6/logs/db.log
pid: 20685
server shutdown complete: /home/sseighma/.thermostat-1.6/data/db
log file is here: /home/sseighma/.thermostat-1.6/logs/db.log
```

SETUP THERMOSTAT



SETUP THERMOSTAT



RUNNING THERMOSTAT

```
$ thermostat local
starting storage server...
server listening on ip: mongodb://127.0.0.1:27518
log file is here: /home/sseighma/.thermostat-1.6/logs/db.log
pid: 23666
```

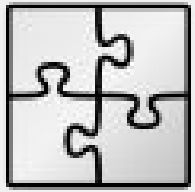
PLUGINS

DEMOS

THERMOSTAT PLUGINS

EXTENDING THERMOSTAT

- Customize Agent and/or Client
- Collect, Record, Analyze your own metrics
- Integrate your own tools
- Most built in functionality is actually plugins
- Documented API



OTHER FEATURES

BYTEMAN INTEGRATION

- Bytecode Manipulation Tool
- Change the operation of Java app either at load time or while the app is running without the need to rewrite or recompile the original program
- Can even be used to modify Java code which forms part of the Java virtual machine
  - Classes such as String, Thread etc.
- Capable of injecting inline Java code into almost any location reachable during execution of a Java method

SHENANDOAH VISUALIZER

- Shenandoah is an ultra-low pause time garbage collector that reduces GC pause times by performing more garbage collection work concurrently with the running Java program
- CMS and G1 both perform concurrent marking of live objects, Shenandoah adds concurrent compaction, which means it's pause times are no longer proportional to the size of the heap
- Garbage collecting a 100 GB heap or a 2 GB heap has the same predictable pause behavior

RESOURCES

DEVELOPMENT

WHAT'S NEXT

- Adhering to standard API versioning
- API breaking changes → Thermostat 2.0 / NG
- Non-breaking changes/bugfixes → Thermostat 1.6
- Integration with more tools
- More automated learning/detection

THERMOSTAT-NG

MICROSERVICES

- Thermostat.Next
- Microservices based

TRY THERMOSTAT

- Red Hat Enterprise Linux 6 and 7
  - Red Hat Software Collections
  - `yum install thermostat16-thermostat`
- Fedora 21+
  - `yum install thermostat`
- Sources
  - http://icedtea.classpath.org/download/thermostat/

CONTRIBUTING

Starting point: http://icedtea.classpath.org/thermostat

- Mailing lists
- IRC
- Bug tracker

- Features
- Report Bugs
- Code

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos