

Creating your own Debugging Tool with JDK Serviceability Technologies

Andrei Pangin
Odnoklassniki



About us

- Andrei Pangin



Lead developer



Specialist

- Odnoklassniki (ok.ru)

- Top 20 site (SimilarWeb)
- 50M DAU
- 1000 Gb/s
- 8000 servers
- 99% Java code

Troubleshooting

Bugs

Performance

Memory

JVM crash

Debugger

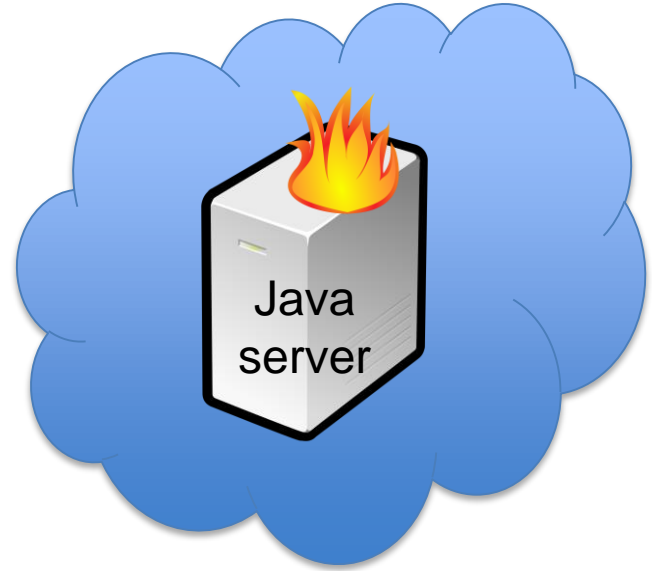
Profiler

Memory
analyzer

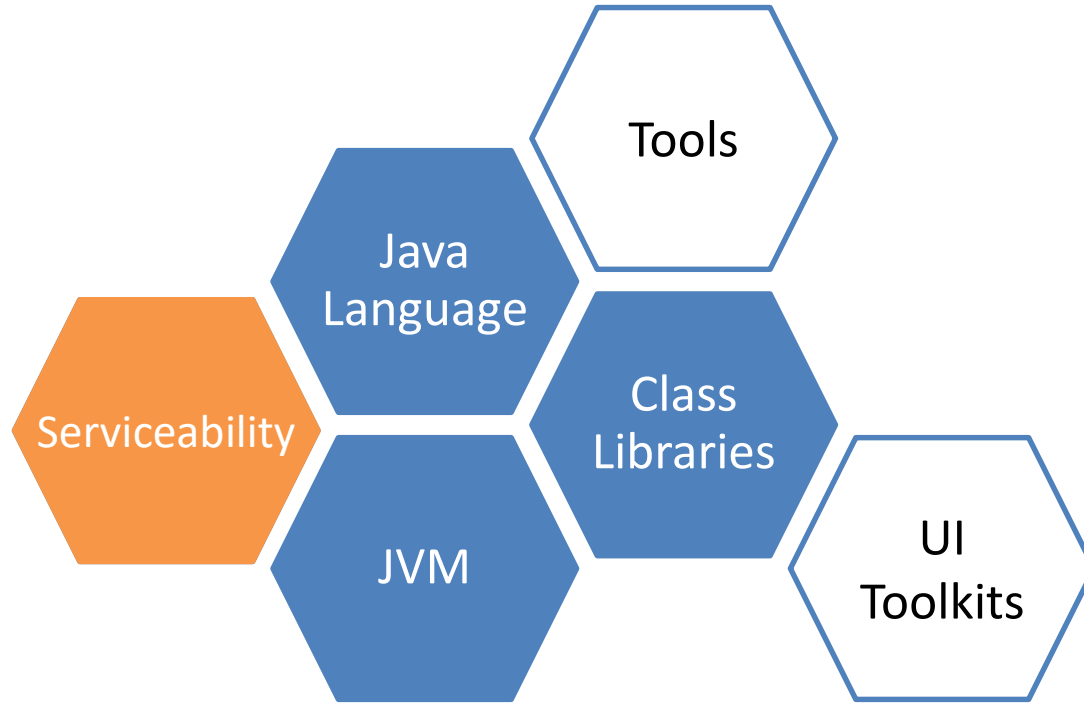
Crash dump

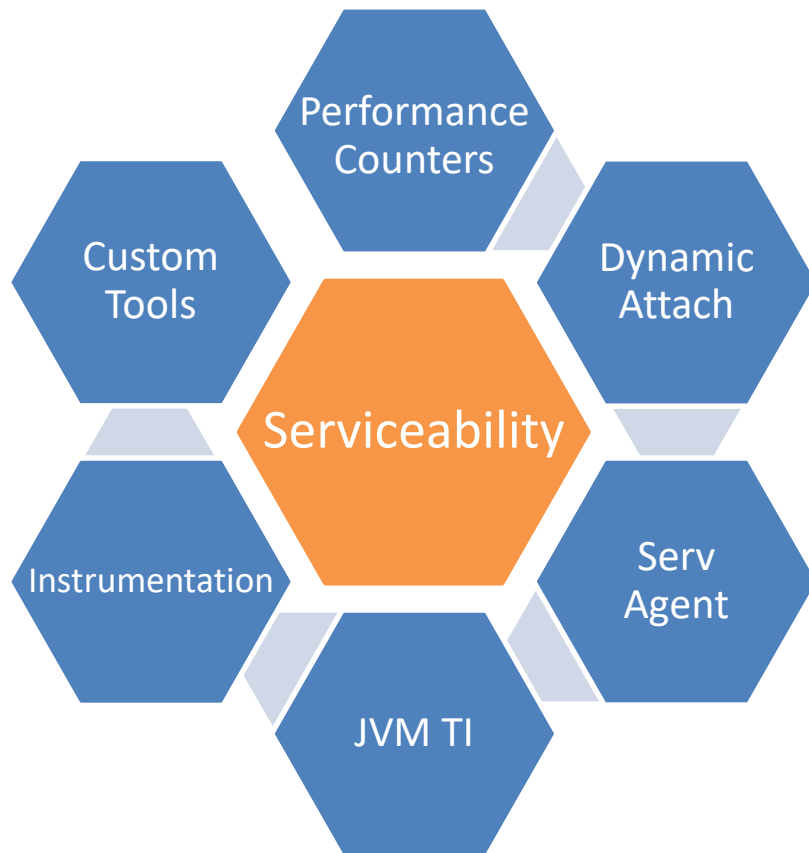


SSH



Java Platform





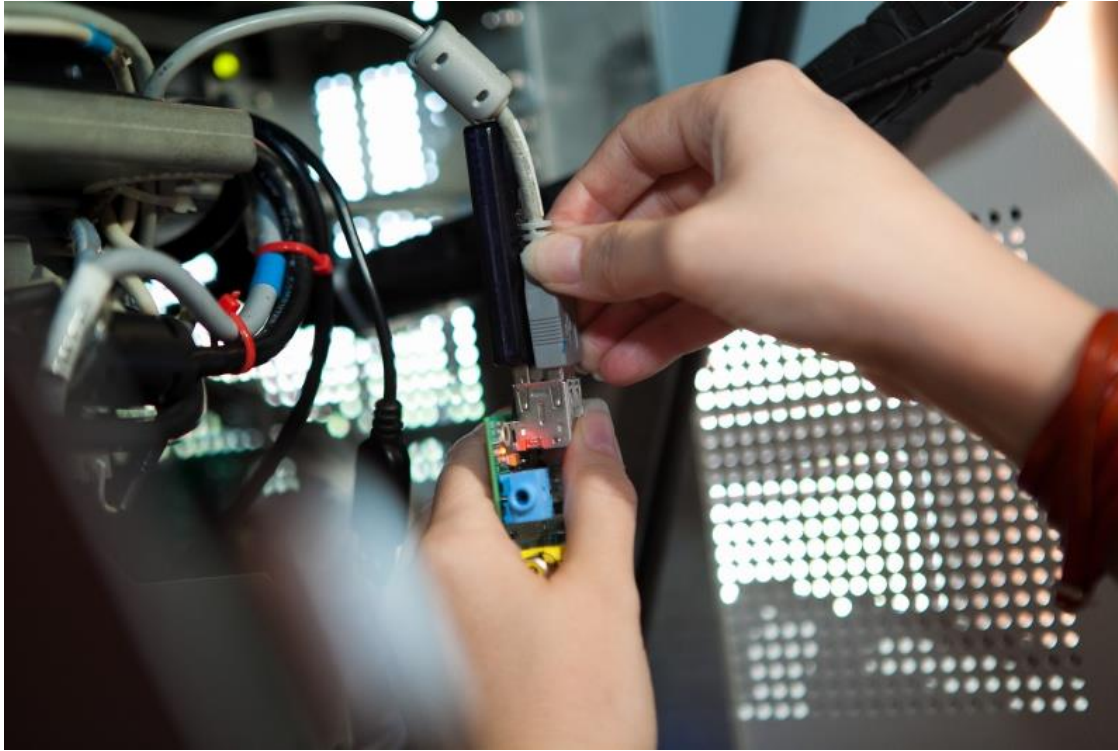
Performance Counters



Performance Counters

- 250+ variables
- Exported to /tmp/hsperfdata_user
- Free to read
- -XX:-UsePerfData
- jstatd

Dynamic Attach



Dynamic Attach

- Cooperative IPC
- Commands:
 - dump threads, dump heap, manage flags
 - run GC, rotate logs, start JFR
 - load agent lib
- `-XX:+DisableAttachMechanism`

Serviceability Agent

- JVM internals → Java API
- sa-jdi.jar



Serviceability Agent

- Reads process memory / coredump
- Reflection-like API
- No cooperation from target JVM
- jsadebugd

JVM Tool Interface

- Standard
- Native API

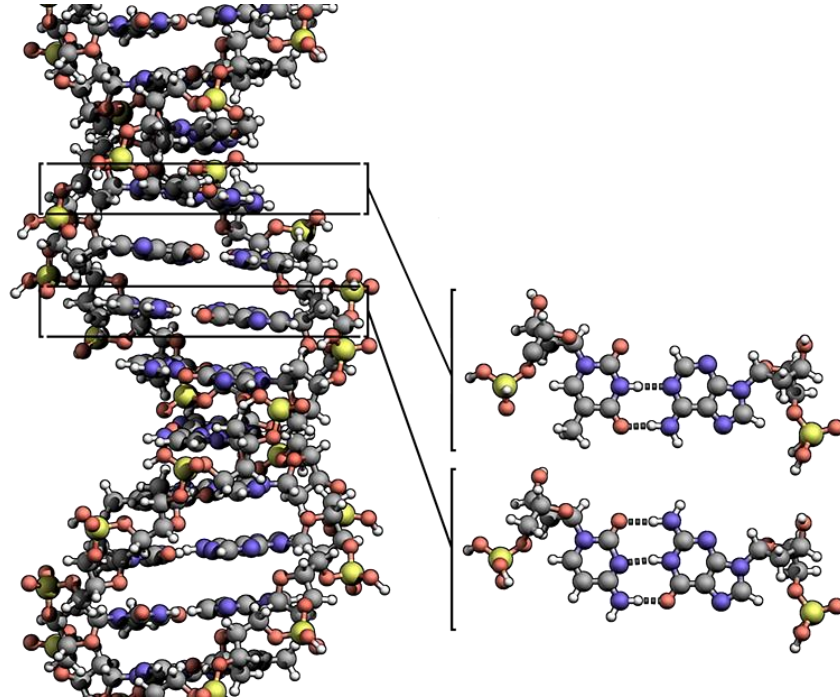


JVM Tool Interface

- Native API for debuggers
 - threads, stack frames, local variables
 - heap walking
 - breakpoints, watchpoints
 - event callbacks
- Capability-based
- `java -agentpath`

Instrumentation API

- Standard Java API
 - `java.lang.instrument`
- Patch bytecode
 - `redefineClasses`
- `-javaagent` or dynamic attach





Documentation

- Serviceability in HotSpot
 - <http://openjdk.java.net/groups/hotspot/docs/Serviceability.html>
- Dynamic Attach
 - <https://docs.oracle.com/javase/8/docs/jdk/api/attach/spec/>
- Serviceability Agent
 - <https://docs.oracle.com/javase/8/docs/serviceabilityagent/>
- JVMTI
 - <http://docs.oracle.com/javase/8/docs/platform/jvmti/jvmti.html>
 - Demos and samples: [\\${JDK}/demo/jvmti](#)

Thank You

@AndreiPangin
andrey.pangin@corp.mail.ru

<https://github.com/odnoklassniki>

