



I bet you have a memory leak

Nikita Salnikov-Tarnovski
@iNikem

Plumbr

- Estonian startup trying to solve performance problems
- We started with memory leaks and solved them :)
- Ongoing effort, more problems to be solved

Memory leak

- A situation where some objects are not used by application any more, but GC fails to recognize them as unused
- The notion of “object is not used by application any more” is totally, absolutely, 100% application specific!

<http://plumbr.eu/blog/what-is-a-memory-leak>

ClassLoader leak

- The most common leak in web applications
- After application undeploy classes are not longer needed
- If even 1 class is left behind, he keeps all his friends
- After some reddeploys, you crash

Should you care?

- PermGen will die soon
 - Metaspace does not eliminate the problem
- No redeploy, no problem
 - Yeah, but...

Scout rule

Leave this world a little better than you found it.

(c) Robert Baden-Powell
founder of scouting movement

The problem is...

- Most of the classloader leaks you will ever encounter are not your fault
- Double-edge sword of reuse and modular development
- You have no idea what do you use in your application

Demo

- We are eating our own dog food
- Test our client-facing application with our agent
- Quite small web application
- CRUD with JPA, Spring, JSP, WebSockets
- Running on Jetty

Solving performance problems is hard. We don't think it needs to be.

Plumbr

@JavaPlumbr/@iNikem