OSGi Performance Bloopers

Martin Lippert

 © 2009 Martin Lippert; made available under the EPL v1.0

Titanic

Camera reflected in door

Star Wars

Stormtrooper bangs head on door

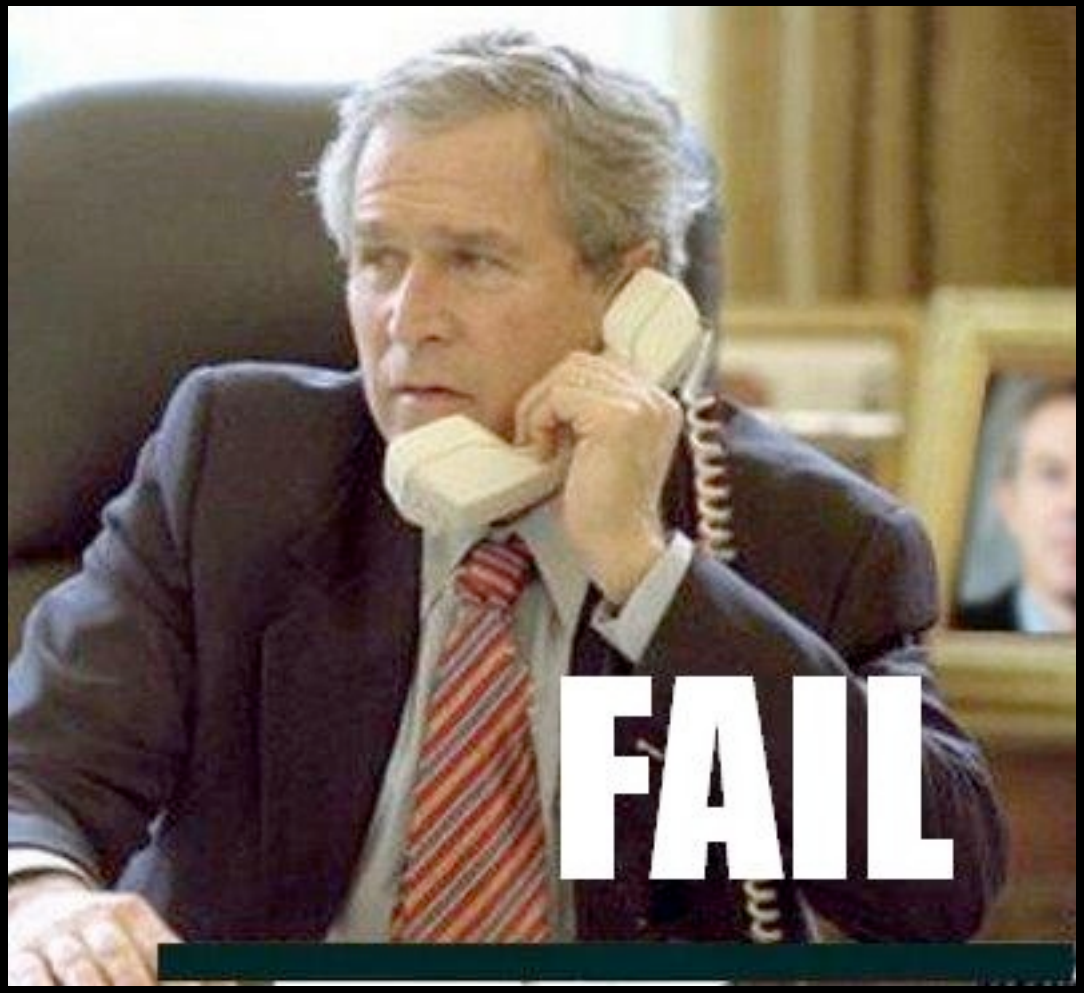
PotC: Curse of the Black Pearl

Crew member visible

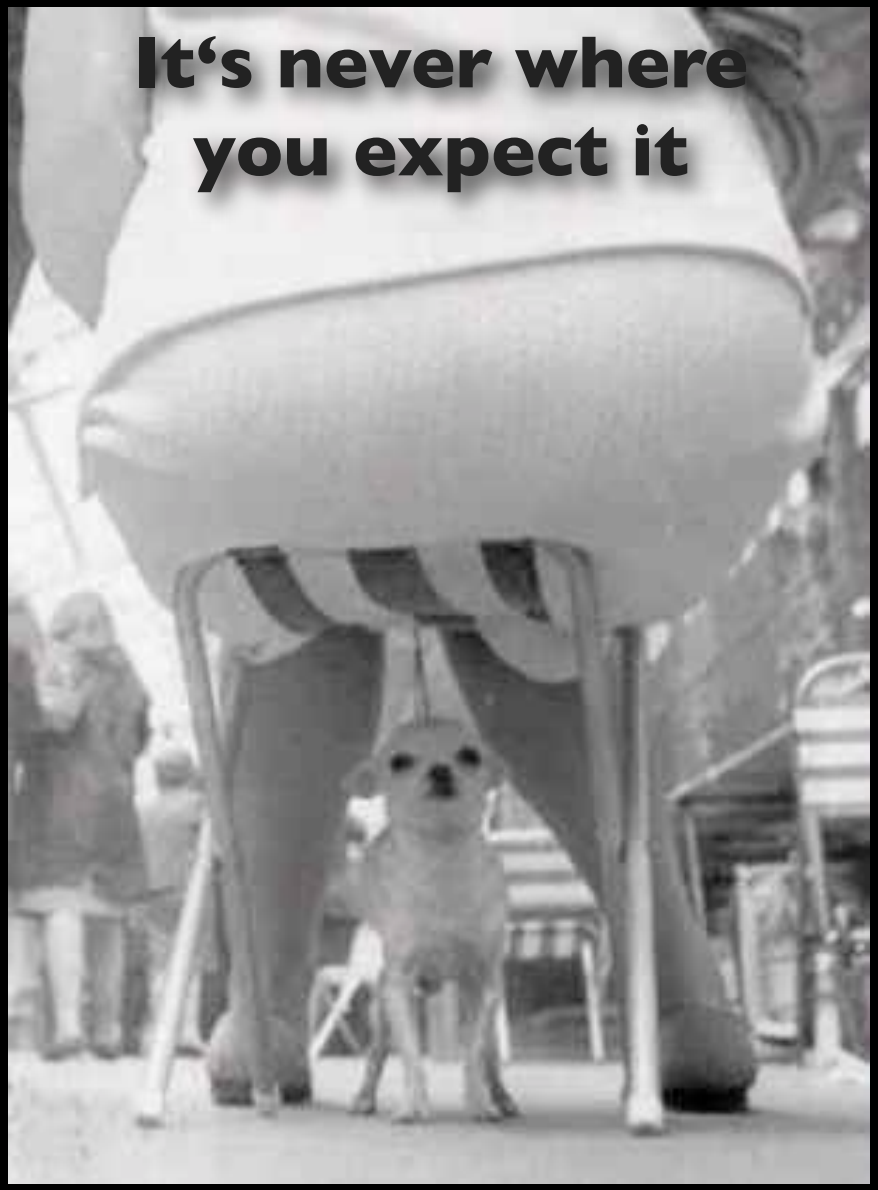
What is this all about?

„Take the fat man off your app and it will run faster“

Find out where the problem is...



failblog.com



failblog.com

Measure !!!
VACUUM

In most cases your performance problems are not related to OSGi at all...

Instead

poor database access

scalability of algorithms

wrong usage of HashMaps

Strings

...

But now, lets focus on

OSGi

Startup

A solution...?!?

Proposal mock-up – not an actual program

from Chris Laffras talk on Eclipse performance

Don‘t turn on everything...!!!

Bundle activation can be expensive

Avoid bundle activation
if possible

Use instead
extension registry
lazy activation
no activation

Do work asynchronously

How?

Lazy initializations
Threading (but complex)
Extender pattern

Take care what others add

failblog.com

Extenders can cause „interesting“ behavior at startup

Example:

Spring Dynamic Modules

Spring DM

creates application context (on bundle activation)
sync or async

sounds good, eh?

But...

XML parsing is expensive
proxy bean creation is expensive
bean creation can be expensive

if you do this for hundreds on bundles,
you will wait for a long time

Avoid bean + proxy creation

„lazy-init=true“

Lazy activation can also be dangerous

bundle is activated
spring context is created
class is loaded from another bundle

the provider bundle is now activated (lazy)
runs itself activator
creates spring context

Don‘t clean too often

Equinox has the „-clean“ option

Don‘t use it!!!

It cleans all the OSGi framework caches at startup

modules, abstractions, APIs

sometimes results in
calling the same
method over and
over again

Conclusions

1. Measure
2. Optimize

3. Goto 1.

Thank you for your attention

Martin Lippert:
martin.lippert@it-agile.de

 © 2009 Martin Lippert; made available under the EPL v1.0