



Hotpatcha din Java-6-applikation

Markus Wahl

IBS JavaSolutions

Javablogg: **<http://jsolutions.se>**



Hotpatcha din Java-6-applikation

- Tekniken bakom jConsole: Suns Attach API, `com.sun.tools.attach`
 - ansluta en egen agent
- Kombinera med Reflection-API och Instrumentation-API
 - ändra programkoden
 - inspektera datastrukturerna
- D v s skriver en agent som
 - ändrar byte-koden
 - i laddade och t o m instansierade klasser



Hotpatcha din Java-6-applikation

- Olika syften
 - Exempelvis: förenkla loggningssyntaxen

D v s istället för att skriva:

```
if( logger.isDebugEnabled() ) {  
    logger.debug( "The large object is " + largeObject.toString() );  
}
```

Kan man m h a Instrumentation skriva:

```
logger.debug( "The large object is " + largeObject.toString() );
```

- Krystat exempel, men tekniken finns



Hotpatcha din Java-6-applikation

- Hotpacha = att utföra instrumentation från en Java-agent
- Java-agent körs i ett befintligt JVM för att
 - genskjuta klassladdningen
 - upprepa klassladdningen
 - modifiera byte-kod
- `java.lang.instrument` saknar programmatiskt stöd
 - byter endast ut byte-kod
- Agenten: 2 instrumentation-val:
 - skicka med preparerad byte-kod
 - skapa byte-kod i runtime (ex Javassist eller ASM)



Hotpatcha din Java-6-applikation

- Demonstration 1
 - Agent + Reflection
 - Agent + Instrumentation m h a förkompilerad klassfils-ersättare
 - Bitvis svårt: inre klasser → egna klassfiler
 - Agentjarfil blir skrivskyddad



Hotpatcha din Java-6-applikation

- Att tänka på
 - Klassfilens schema ändras inte
 - Metoder avbryts inte
 - Metodernas innehåll ändras
 - Oförändrad access → datastrukturer kan ändras
 - Attribut kan inte tillföras
 - Suns Attach-API saknar "remote"-funktion



Hotpatcha din Java-6-applikation

- Att tänka på
 - Aningen krångligt → Hotpatchning är en sista utväg
 - Appservrar/webservrar har ju hot deploy
 - Hotpatching vs debugging: olika syften



Hotpatcha din Java-6-applikation

- Javassist
 - programmatisk styrd byte-kodsförändring
 - behovet inte lära sig byte-kod
 - Använder vokabulären från Java-språket
 - Agent med Javassist kan bli mer dynamisk
- Fler alternativ, ex ASM



Hotpatcha din Java-6-applikation

- Demonstration 2
 - Agent + Instrumentation m h a Javassist



Länkar

- Javablogg på svenska – <http://jsolutions.se>
- Attach + Instrumentation – <http://www.fasterj.com/articles/hotpatch1.shtml>
- Attach + Instrumentation – http://blogs.sun.com/sundararajan/entry/using_mustang_s_attach_api
- Instrumentation – <http://www.javalobby.org/java/forums/t19309.html>
- Attach API – <http://java.sun.com/javase/6/docs/jdk/api/attach/spec/index.html?com/sun/tools/attach/package-tree.html>
- Instrumentation API – <http://java.sun.com/javase/6/docs/api/java/lang/instrument/package-summary.html>
- Javassist – <http://www.csg.is.titech.ac.jp/~chiba/javassist/>
- Loggingsexemplet – <http://surguy.net/articles/removing-log-messages.xml>
- ASM – <http://asm.objectweb.org/>