

Unit-testing concurrent mode

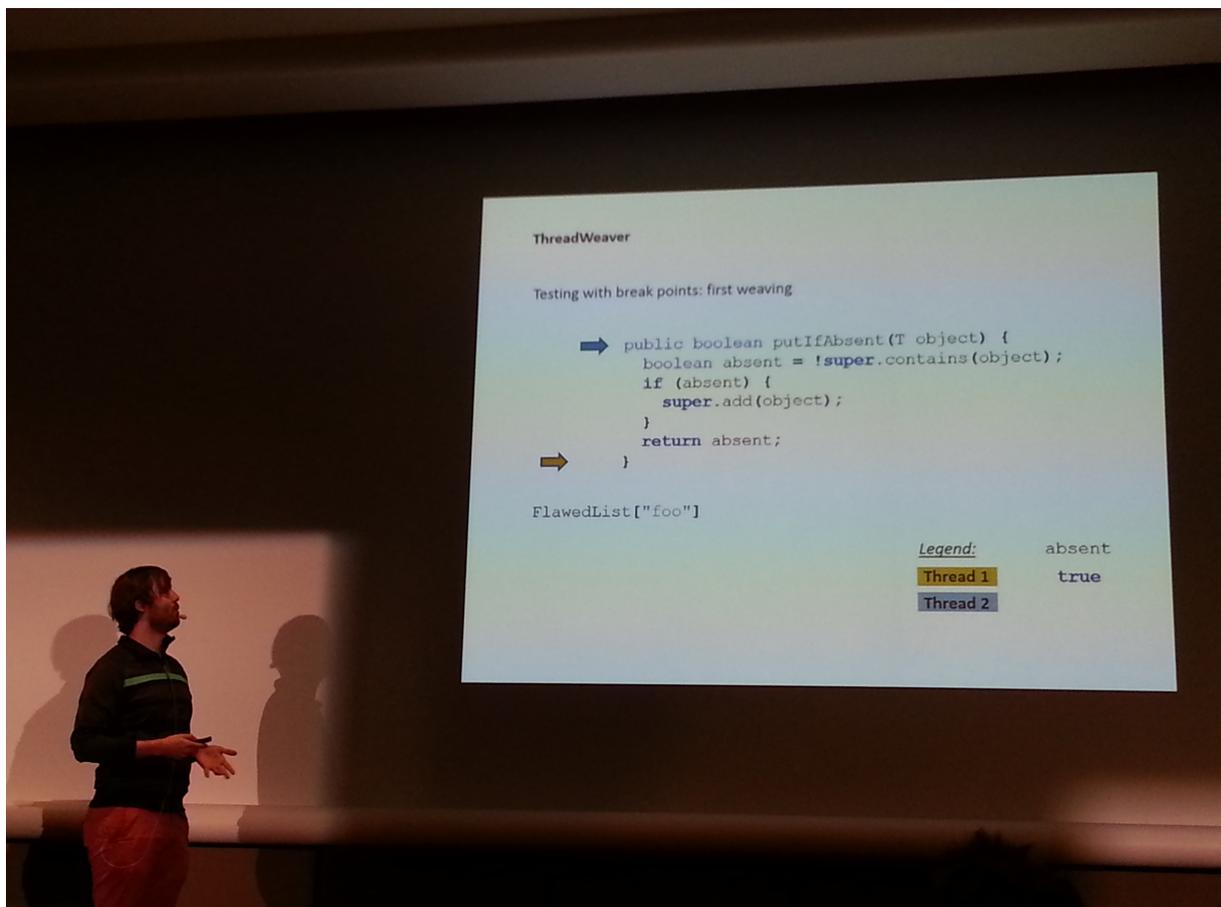
Date : 9 avril 2015

Format : Quickie

Speakers : Rafael Winterhalter, Bouvet

Beaucoup de personnes trouvent difficile de tester le code concurrent.
Donne un exemple de code non thread-safe (racing condition) : méthode putIfAbsent

1. Le test JUnit fonctionne.
2. Dans TestNG, l'annotation @Test prend en paramètre la taille du pool de threads et le nombre d'invocations. Il faut lancer 1000x le test pour que le test échoue. Ce n'est pas satisfaisant.
3. Utilisation de TestNG combiné avec un point d'arrêt => cela fonctionne, mais nécessite d'une intervention humaine ...
4. L'outil **ThreadWeaver** est l'objet de ce Quickie : l'exécution de 2 threads va s'entrelacée. ThreadWeaver positionne des breakpoints. Permet de reproduire systématiquement ce type de problèmes.



ThreadWeaver propose plusieurs annotations : @Threaded<...>

Il utilise l'instrumentation de byte-code. Il réécrit la classe de manière transparente pour le développeur. Possibilité de décompiler le code instrumenté.

Attention au modèle mémoire de Java. ThreadWeaver passera à côté.

5. Utilisation de jcstress de OpenJDK : concurrency stress testing concurrent mode

Annotations : @JCStressTest, @Actor, @Arbiter, @Outcome

Attention, ce n'est pas un outil de tests unitaires. Qui plus est, le résultat peut différer d'une architecture à une autre (ARM vs x86).