Injection de code source et couverture de code en temps réel

Auteurs :
Maxime CLEMENT
Jordan PIORUN

16 Décembre 2015

Table des matières

Introduction

La couverture de code est une métrique représentant le pourcentage de nombre de ligne de code exécuté. Cette métrique est utiliser lors l'exécution de suites de tests afin de mesurer le nombre de ligne de code couvertes par ces suites.

Certaines méthodes de développement comme le TDD[1] vont garantir une bonne couverture de par le fait que les tests sont écrit avant le code. Une question peut alors se poser, la totalité du code couvert est il bien exécuté en production ? Il est probable qu'une ligne de code exécuté lors d'une suite de tests, ne soit jamais exécuté dans un environnement de production.

Le but est montrer que calculer la couverture de code sur un programme exécuté dans un environnement de production est possible, de plus, ce calcul pourrait être réaliser en temps réel pour ne pas avoir à stopper l'exécution du code en production afin d'obtenir cette fameuse métrique.

Pour atteindre notre objectif, nous avons utiliser Spoon, une librairie Java permettant de faire de la transformation de code source Java. L'idée est que le programme transformé puisse d'auto-instrumenter afin de notifier à l'utilisateur sa couverture à un moment t en temps réel.

Nous avons évaluer notre approche sur plusieurs critères. Tous d'abord la couverture de code calculer par notre outils comparé à d'autre déjà existant, ensuite le coût supplémentaire nécessaires en mémoire afin de pouvoir réaliser ce calcul en temps réel. Et pour finir l'impact sur le temps d'exécution du programme instrumenté par rapport à l'original.

[1] Test-Driven Development

Réalisation

1.1 Approche

Creer une section Principes [1]

1.2 Architecture et design

1.3 Technologie

Évaluation

2.1 Pertinence

2.2 Temps

2.3 Mémoire

2.4 Complexité

Bilan personnel

Conclusion

Bibliography

[1] Ira. D. Baxter. Branch coverage for arbitrary languages made easy. 2002.