

Rapport d'objets dupliqués

Korantin Auguste & Maxime Arthaud

21 janvier 2014

1 Introduction

À première vue, le sujet nous paraissait relativement intéressant : concevoir un système gérant l'état d'objets est plutôt intéressant, et il y avait l'air d'y avoir pas mal de challenge au niveau de la concurrence et des problèmes de deadlock, etc...

2 Phase 1

Le code est venu plutôt naturellement à partir du squelette fourni.

Nous avons rapidement abouti à une version fonctionnelle, mais après développement d'un « fuzzer » pour torturer un peu notre système, plein de bugs/deadlock sont apparus. Même si la plupart ont pu être résolus assez rapidement, certains apparaissaient dans des cas très vicieux et ont été particulièrement ardues à résoudre.

3 Phase 2

Nous avons commencé par envisager l'utilisation de l'API « reflect » de Java pour créer une classe proxy qui serait capable d'exposer les fonctions originales de manières transparentes, mais avons fini par abandonné devant certains problèmes très difficiles à surmonter (problèmes de type, principalement).

Nous nous sommes donc rabattus sur l'utilisation de l'API « reflect », mais seulement pour analyser les classes et faire ensuite de la génération de code bête et méchante, qui a l'avantage d'être très simple. Mais c'est sale.

4 Phase 3

La phase 3 a consisté principalement à rajouter une méthode pour sérialiser des `SharedObject`, ce qui n'a pas posé de difficulté particulière, même si des tests ont pu montrer qu'il y avait quelques corrections annexes à effectuer pour que tout fonctionne bien.

5 Conclusion

Le projet n'a pas été toujours évident, principalement à cause de la gestion de la concurrence dans tous les cas tordus qui peuvent arriver en montant en charge. Au final, nous pensons toutefois avoir quelque chose de très robuste !

Le reste n'a pas vraiment posé problème. Ah si : on aime pas le Java, mais encore sur ce projet, ça allait (pas comme le JEE, horreur !).