

Concorrência e Paralelismo 2016-17 - HW 4 - Transactional Memory

Aluno: Gonçalo Sousa Mendes, n.º 42082, Turno p.3
Prof: João Lourenço
Dept. de Informática
FCT/NOVA

24 de Dezembro de 2016

1 Introdução

Neste último trabalho tinha-se de aplicar a solução de memória transacional, dada nas aulas teóricas, ao problema dado no projeto.

2 Implementação

A implementação desta solução foi bastante simples, sendo apenas necessário identificar os locais onde os acessos teriam de ser protegidos, isto é, onde é que tinha de haver exclusão mútua e colocar a anotação *@atomic* antes dos métodos. Este métodos foram o de inserção, remoção e leitura.

Foram testados todos os algoritmos para conseguir se conseguir descobrir qual o melhor para este problema, sendo o *tl2inplace* o que apresentou mais operações por segundo. Faz-se ainda o reparo que todos os algoritmos passam os testes de validação.

3 Conclusão

A primeira coisa que se consegue perceber com esta solução é a simplicidade da implementação, mesmo quando esta fornece garantias fortes das propriedades ACID, o que com uma implementação baseada em *locks* é difícil de se conseguir. No entanto, esta solução apresenta uma grande desvantagem quando comparada com a solução encontrada no projeto, usando *locks* de granularidade média, tem uma eficiência bastante mais baixa.

Esta é a principal preocupação quando se quer encontrar uma maneira de tornar um programa sequencial num programa concorrente, perceber bem quais as propriedades que queremos preservar, as invariantes do programa, avaliar as soluções existentes e fazer o balanço entre a simplicidade da solução contra a eficiência que ela mesma tem.