Ano Lectivo 2015/2016

Concorrência e Paralelismo

Projecto 2

Docente: João Lourenço

Alunos: Nuno Mendonça Nº 41623

David Gago Nº 41710

Introdution

Este segundo trabalho prático encontra-se integrado na cadeira de Concorrencia e Paralelismo do 4º ano de M.I. em Engenheria Informática.

O propósito deste trabalho é tornar um programa que funciona apenas com um processador (100% operacional) para um que funciona com vários processadores sem erros de concorrencia, pois ao pegar no código fornecido e usar mais de um processador o programa não executa correctamente.

O programa simula um repositorio de artigos cientificos. Cada artigo tem um título, um conjunto de autores e um conjunto de keywords. A aplicação tem 3 mapas para representar a memória e cada mapa guarda respectivamente artigos, autores e keywords ou palavras-chave.

Approach

O método de resolução deste trabalho passa por usar locks de modo a fazer com que o programa execute correctamente com multi-threads. Estes locks estão localizados nas funções de adição, remoção e “get” encontradas na classe Repository.

Existe então um array de locks para o mapa dos Artigos e 2 mapas de Locks para respectivamente Keywords e Autores.

Para implementar estes locks basta que quando uma função não encontra nenhum problema, por exemplo, quando tenta adicionar um elemento e ele já existe, ou tentar remover um elemento que não existe, se dê lock todas as posições que venham a ser afectadas por essa função, não permitindo assim que outras funções façam alterações sobre essas posições.

Validation

Já existiam funções de validação disponibilizadas no codigo fornecido. No entanto adicionamos mais 4 validações para testar os nossos resultados.

Duas das validações adicionadas consistem numa especie de replica das que ja estavam disponibilizadas que consistiam em percorrer todos os artigos guardados e confirmar se os autores e palavras chaves desses artigos existiam. Então as novas validações criadas consistem em percorrer todos os autores e keywords e ver se os respectivos artigos que foram escritos pelos autores ou têm essas palavras chaves existiam guardados.

As outras duas validações adicionadas consistem em verificar se todo o numero de adições no sistema bem sucedidos supera o numero de remoções bem sucedidos no sistema e verificiar se a soma das das funções realizadas bem sucedidas é igual a totalidade das funções bem sucedidas.

Evaluation

Para testar a nossa resolução, efectuamos vários testes. Ao longo da implementação dos locks fomos sempre obtendo resultados negativos, ou seja, a aplicação nunca finaliza sem parar nalguma das validações.

Por outro lado, para testar as validações, corremos o programa com multiplos processadores, apagando momentariamente as funções de validação ja fornecidas. O programa nunca conseguiu finalizar, provando assim que as nossas validações estavam correctas.

Tudo somado, deu para concluir que se o programa terminava correctamente com vários processadores que a nossa implemtanção dos locks permita superar todas as validações implementadas fazendo assim com que o comportamento do programa fosse o mais correcto possivel.

Conclusions

Podemos então concluir que este projecto ajuda conclusivamente a entender melhor a materia lecionada nas aulas práticas. Num curso como Engenharia Informatica é sempre necessário uma forte componente prática geralmente mais importante que a teorica que permite intriorizar e familiziar com os conceitos lecionados.

É de referir que o todos os objectivos foram atingindos com a conclusão deste projecto.

Acknowledgments

Agradecemos ao docente João Lorenço e a todos os alunos que colocaram duvidas no grupo que indirectamente nos ajudaram na resolução/tiraram algumas duvidas