Projeto de Laboratórios de Informática 3
Grupo 26

José Pinto (A81317) Luís Correia (A81141) Pedro Barbosa (A82068)

12 de Junho de 2018

Resumo

Neste projeto da disciplina de Laboratórios de Informática 3 (LI3) do Mestrado Integrado em Engenharia Informática da Universidade do Minho foi-nos proposto desenvolver um sistema em Java capaz de processar ficheiros XML que armazenam informações utilizadas pelo Stack Overflow, podendo depois executar um conjunto de querys de forma eficiente.

Conteúdo

1 Introdução

Neste relatório vamos primeiramente falar das principais mudanças efetuadas em relação ao projeto realizado em C, seguido de uma breve explicação de como foi feito o parsing dos documentos XML, acabando com uma breve opinião sobre a realização do trabalho.

2 Principais mudanças

2.1 Posts

No projeto em C decidimos optar por guardar todos os posts numa lista ligada ordenada do mais recente para o mais antigo. Para podermos encontrar um post na lista e percorrê-la a partir desse post utilizámos uma hash table que associava um post à sua posição na lista. Para ser possível percorrer a lista a partir uma data também utilizamos uma hash table que associava um mês à posição na lista do último post deste. Em Java não é possível utilizar pointers a apontar para uma posição da lista (pelo menos com os conhecimentos obtidos na cadeira de POO) no entanto conseguimos manter as mesmas funcionalidades que em C. Para tal decidimos optar por recorrer ao uso de um ArrayList em vez de uma lista ligada. Desta forma, em vez de guardar nas tabelas de hash (HashMaps) um apontador, é guardado o índice do post nesse array. Optámos por substituir a lista ligada por um array pois este método é dependente do tempo de acesso a um dado elemento o que torna um array a escolha óbvia. Desta forma, a estrutura em C e a em Java

são funcionalmente equivalentes. Enquanto que em C, para ser possível percorrer a lista de posts externamente, utilizamos um tipo abstrato de dados que representava um apontador para a lista e bastava devolver o apontador para o próximo elemento, em Java optámos por uma estratégia diferente. Criamos um método get_next() que dado um post retorna o próximo post. Para tal este recorre ao hashMap para obter o seu índice na lista e simplesmente retorna o próximo elemento.

2.2 Restantes classes

A conversão das restantes estruturas de dados/classes para Java foi realizada sem grandes dificuldades. As alterações feitas são devidas às típicas diferenças entre as duas linguagens de programação e achamos que nenhuma destas requer uma menção específica.

2.3 Interrogações

Devido a conseguirmos transformar a estrutura de dados de C para Java sem qualquer alteração funcional, todas as interrogações foram resolvidas com os mesmos algoritmos(descritos no relatório anterior). Apenas foram feitas pequenas alterações para permitir o uso de exceções e de estruturas de dados auxiliares diferentes.

3 Parsing dos documentos XML

Uma das falhas do nosso projeto em C foi a impossibilidade de lidar com um dump com ficheiros XML de tamanho significativo (devido à utilização de um parser pouco adequado). Para colmatar essa falha utilizamos o parser SAX (Simple API for XML). Assim é possível processar ficheiros significativos e esse processamento também é realizado significativamente mais rápido do que o conseguido em C.

4 C vs Java (opinião sobre a realização do trabalho)

Em relação ao projeto realizado em C, foi mais fácil garantir a modulação e o encapsulamento no projeto realizado em Java. Foi simplesmente seguir as práticas de encapsulamento standard de Java (variáveis private, clones, gets...) e tirar partido da hierarquia de classes (Por exemplo as classes Question e Answer são subclasses de Post). Também houve utilização de outras funcionalidades do Java como exceções e Comparadores. Apesar de todos estes benefícios o encontrámos algumas barreiras no que toca a apontadores(como mencionado anteriormente) e a manipulação de memória.