Projeto Laboratórios de Informática III

Fase 2

Projeto desenvolvido por

Diogo Silva (A104183), João Pinto (A104270) e Miguel Barrocas (A104272)

Grupo 75

Licenciatura em Engenharia Informática

Universidade do Minho
Escola de Engenharia

Departamento de Informática

Universidade do Minho

Conteúdo

Introdução

Este projeto está a ser desenvolvido no âmbito da UC de Laboratórios de Informática III do ano 2023/2024, cujos objetivos de aprendizagem estão relacionados com o processo de programação, desenvolvimento de projetos e trabalho em equipa.

Após a conclusão da primeira fase, que consistia em implementar o *parsing* dos dados e o modo *batch*, vamos agora tentar aprimorar requisitos do programa, nomeadamente tornar o programa capaz de realizar todas as dez queries e criação de um modo interativo. O modo interativo consiste numa interface gráfica no terminal que permite fornecer ao programa a diretoria dos ficheiros com os dados e realizar queries introduzidas pelo utilizador.

As estruturas e algoritmos implementados anteriormente também seriam postos ao teste, visto que nesta fase foi introduzido um *dataset* novo, muito maior que o antigo. Assim sendo, quaisquer ineficiências ou ideias mal concebidas iriam ser ainda mais pesadas para um bom funcionamento.

Com o tempo máximo de execução do programa definido em dez minutos, aceitamos o desafio e começamos a analisar as escolhas da primeira fase e formas de as melhorar ou corrigir. Deparamos-mos também com uma ocupação de memória máxima de 5GB, que se acabou por tornar numa das principais dificuldades passadas nesta segunda fase.

Arquitetura e Estruturas

Recapitulando a estrutura base da primeira fase, optamos por armazenar os dados lidos pelo parser em três *hash tables* diferentes, sendo que na primeira fase eram 4, mas decidimos remover a dos *passengers* pois esta antiga forma que tínhamos ocupava muita memória e demorava muito tempo a processar, sendo que da forma renovada ficou mais eficiente e rápido.

Após a apresentação da primeira fase à equipa docente, a nossa visão e estrutura foi bem recebida, mas seguimos uma dica de melhoria que nos foi dada, onde teríamos de transformar os antigos *Catalogs* em um só, de forma a tornar o código mais eficiente e otimizado. Assim o seguimos e trabalhámos nesta segunda fase.

Resumidamente, não houve grandes mudanças em relação a arquitetura da primeira fase pois as pequenas mudanças que houve foram com o intuito de tornar o código mais eficiente e não ocupar tanto espaço.

Otimizações

Após concluirmos todas as queries, sentimos que a performance do programa estava longe de ser ideal. O tempo de execução do *dataset* grande aproximava-se dos 10 minutos de limite no servidor da equipa docente e a memória ultrapassava mesmo os 5GB máximos.

Logo, com o tempo de execução limitado a 10 min, tentamos reduzi-lo ao máximo. Tal apenas seria possível observando as nossas implementações originais e perceber quais eram os seus pontos fracos, alterando-os para soluções mais eficientes. As três otimizações principais foram:

- Criar um catálogo com todos os catálogos o que ajudou na eficiência do programa;

- A alteração dos *parsers* dos ficheiros de forma que a sua execução seja mais eficiente, sendo agora os erros tratados dentro dos mesmos;

- O parse dos passengers agora é tratado de forma diferente, pois este era o que demorava mais e ocupava mais memória. O *passenger* ao ser processado na "*process_passenger*" recebe dois parâmetros *"flight_id"* e *"user_id"* e verifica se os mesmos existem nos catálogos respetivos, depois, dentro dos voos será guardado numa *hashtable* passengers em "*add_flight_passenger*" os passageiros para cada voo respetivo, sendo que nos *users* temos a "*add_user_flight*" que usa uma lista de voos onde temos para cada *user* os voos que o mesmo realizou, evitando assim que mais para a frente se percorra estas listas outra vez se precisarmos destes parâmetros.

Modo Interativo

Quanto ao modo interativo, ficamos um pouco desiludidos pois não conseguimos finalizá-lo.

Enfrentamos alguns problemas a tentar resolver leaks e a tentar reduzir a ocupação de memória e achamos melhor para o futuro do projeto focar em meter o código a funcionar com o novo data_large e deixar o modo Interativo para o fim.

Infelizmente com todas as complicações, não conseguimos meter este modo a trabalhar como queríamos.

Dificuldades Sentidas

Com a chegada do novo e muito maior dataset, chegaram também novos problemas que persistiram durante muito tempo.

Com este dataset o código não corria. Estava a ocupar muita memória e demorava muito tempo a ser executado, ambos estes fatores estavam muito perto do limite estimulado pela equipa docente, a ocupação de memória chegava mesmo a ultrapassar as 5GB delineadas.

Fomos trabalhando na otimização para resolver estes dois problemas, mas estava a descambar tudo porque ao mudar num sítio partia em outro.

Finalmente quando conseguimos chegar a uma nova solução, ficamos sem tempo para as queries pois precisávamos de alterar a forma como chamávamos os argumentos.

Conclusão

Concluindo, esta segunda fase trouxe-nos muitos mais problemas e não conseguimos obter os resultados finais que pretendíamos.

Claramente não nos correu tão bem como a primeira fase. Foi um grande desafio com muitos conceitos novos e de implementação complexa.

Apesar de tudo sentimos que aprendemos muita coisa e gostamos muito de trabalhar com estes conceitos tais como a modularidade e o encapsulamento, que são fundamentais para qualquer engenheiro desta área.