Relatório 2º projecto ASA 2020/2021

Grupo: al090

Aluno(s): Sara Marques (93342)

Descrição do Problema: Este projeto tem como objetivo minimizar o custo total de execução de um programa constituído por um conjunto de processos considerando que existem dois processadores disponíveis. Para isto, tem se em conta os custos de execução de cada processo em cada processador e os custos de comunicação entre os vários processos, se estiverem em processadores diferentes.

Descrição da Solução: A solução proposta foi desenvolvida em C++. O problema é representado como um grafo dirigido onde os dois processadores (X, Y) e os n processos $(\{p_1,...,p_n\})$ são os vértices. Se considerarmos o processador X como vértice *source* e o processador Y como vértice *sink,* temos n arestas do vértice X para cada processo (cujas capacidades correspondem aos custos de execução $X_1,...,X_n$, respetivamente) e n arestas de cada processo para o vértice Y (cujas capacidades correspondem aos custos de execução $Y_1,...,Y_n$, respetivamente). Adicionalmente, temos para quaisquer dois processos p_ie p_j, tal que o custo de comunicação entre eles é maior que zero uma aresta de p_i para p_j e outra de p_j para p_i (sendo que ambas as capacidades corresponde ao custo de comunicação c_{ij}). Desta forma, conseguimos modelar o problema como um problema de fluxo máximo e encontrar o corte mínimo e, calculando a sua capacidade, temos a solução para o problema. No código desenvolvido este grafo é representado por uma matriz de adjacências de dimensão $(n+2) * (n+2)$.

Como temos a garantia de que a soma dos custos de execução era da ordem de grandeza do número de processos, n, o valor do fluxo máximo calculado será também limitado por $O(n)$. Assim, utilizou-se o algoritmo de Edmonds-Karp, uma vez que podemos majorar a sua complexidade temporal por uma expressão que depende linearmente do valor máximo de fluxo, f^*, que é $O(E\ |f^*|)$. Para valor inicial de fluxo máximo usou-se, assim, $(n+2)^2$, uma vez que é o mínimo valor que podemos ter a certeza que não é inferior ao valor do fluxo máximo calculado. Para representar a rede residual durante a execução do algoritmo usou-se também uma matriz de adjacências. Para além disto foi tido em conta a existências de arcos de refluxo e assim, por cada caminho de aumento encontrado, após calculado o fluxo nesse caminho atualizava-se as capacidades dos arcos e dos arcos de refluxo nesse caminho, reduzindo a primeira e aumentando a segunda com o valor do fluxo. Isto é feito acedendo às posições inversas na matriz da rede residual.

Análise Teórica

Seja V o número de vértices e E o número de arcos na rede de fluxos. No contexto do problema $V = n + 2$ $(V \epsilon O(n))$, em que n é o número de processos, e $E = 2 * n + 2 * k$ $(E \epsilon O(n + k))$, em que k é o número de custos de comunicação entre dois processos. Temos as seguintes complexidades:

- Leitura dos dados de entrada: depende linearmente do número de processos e do número de custo de comunicação, ou seja, $O(n + k)$ que pode ser escrito como $O(E)$;
- Criação da matriz de adjacências para a rede de fluxos e para a rede residual: depende do número de processos, ou seja, $O(n^2)$ ou $O(V^2)$;
- Aplicação do algoritmo BFS para encontrar os caminhos de aumento: depende do número de processos, uma vez que se está a utilizar matriz de adjacências, logo é $O(V^2)$;

Relatório 2º projecto ASA 2020/2021

Grupo: al090

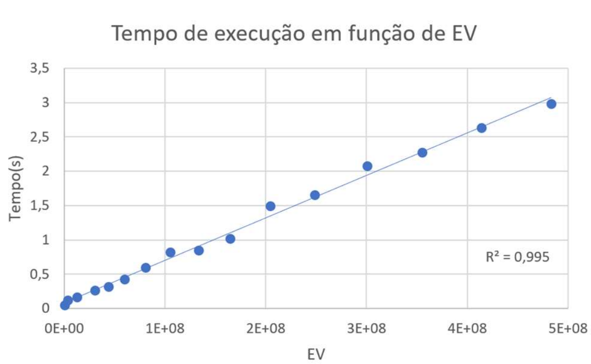
Aluno(s): Sara Marques (93342)

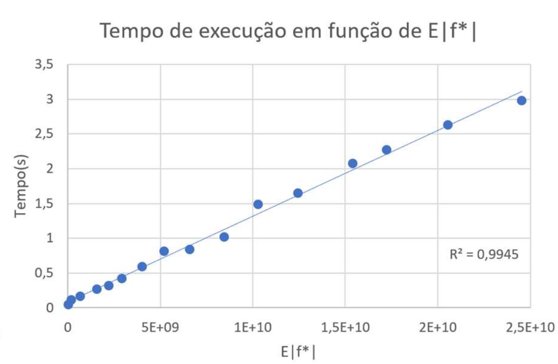
- Aplicação do algoritmos de Edmonds-Karp: pode ter como limites assimptóticos $O(VE^2)$ e $O(E\ |f^*|)$, este último devido ao facto do algoritmo ser uma especificação do algoritmo de Ford-Fulkerson. O valor do fluxo máximo, f^*, é majorado por n, ou por V. Assim, podemos garantir que $O(E\ |f^*|)\ \leq\ O(EV)\ \leq\ O(VE^2)$, e a complexidade da solução em função dos parâmetros do problema é $O(EV)$. Se considerarmos a expressão de complexidade que depende do fluxo máximo, $O(E\ |f^*|)$ temos um limite assimptótico ainda mais apertado.

Complexidade global da solução: $O(EV)$, ou escrita diretamente em função dos parâmetros do problema $O(n(n+k))$.

Avaliação Experimental dos Resultados

Para confirmar a complexidade da solução obtida teoricamente, $O(EV)$, realizaram-se vários testes gerando redes de fluxo com número de arcos vezes número de vértices variável e mediu-se o tempo de execução de cada. Adicionalmente, analisou-se também o comportamento da solução proposta em função do valor de fluxo máximo calculado vezes o número de arcos, ou seja, a majoração pelo limite mais apertado, $O(E\ |f^*|)$. Estes testes foram realizados num computador com processador Intel® Core™ i7 1.10GHz e com 16GB de memória RAM. Os dados obtidos encontram-se nos dois gráficos abaixo.





Como se pode observar pelos gráficos, o tempo de execução cresce linearmente com o aumento de EV no gráfico da esquerda e de $E|f^*|$ no gráfico da direita, tendo ambos um valor de R^2 muito próximo de 1, o que comprova a análise teórica prevista.

Referências

- https://en.wikipedia.org/wiki/Edmonds%E2%80%93Karp_algorithm