

MAC0438 - Programação Concorrente - 1s2012

Relatório

Thiago de Gouveia Nunes
Wilson Kazuo Mizutani

May 12, 2012

Contents

1	Introdução	2
2	Solução desenvolvida	2
2.1	Ideia geral	2
2.2	Estendendo a busca em largura	2
2.3	Aplicando concorrência	2
3	Implementação da solução	2
3.1	Breve descrição das classes	2
3.2	Alguns detalhes de implementação	2
3.3	Sobre a barreira simétrica usada	2
4	Resultados	2
4.1	Resultado para o grafo da NSFNet	2
4.2	Comparações de eficiência	2
5	Conclusões	2

1 Introdução

Esse relatório trata das decisões tomadas na implementação desse EP. Também fornece uma saída do programa para a entrada de exemplo fornecida no enunciado, além de explicitar a localização no código da implementação da barreira simétrica usada. Informações sobre como compilar, as dependências necessárias e o modo de uso do programa encontram-se no arquivo LEIAME.

2 Solução desenvolvida

2.1 Ideia geral

A ideia geral da nossa solução se divide em duas partes:

1. Estender o algoritmo de busca em largura (Breadth-First Search, um caso particular do algoritmo de Dijkstra no qual todas as arestas possuem custo 1) para que ele encontre não só o menor caminho, mas sim os n menores caminhos.
2. Refatorar esse algoritmo para usar programação concorrente, de tal maneira que cada thread seja responsável por tentar encontrar um novo caminho e depois sincronizar com as demais, criando assim um processo iterativo.

2.2 Estendendo a busca em largura

2.3 Aplicando concorrência

3 Implementação da solução

3.1 Breve descrição das classes

3.2 Alguns detalhes de implementação

3.3 Sobre a barreira simétrica usada

4 Resultados

4.1 Resultado para o grafo da NSFNet

4.2 Comparações de eficiência

5 Conclusões