Computação Paralela e Distribuída 2012/13

Enunciado do Trabalho Prático sobre *Algorithmic Skeletons*

O conceito de *algorithmic skeleton* foi definido por Murray Cole [1] como sendo

"a programming template which is parameterised by the application programmer with problem-specific customising functions and commonly specified as a higher-order function [...] which can perform recursive computations"

Em geral, *algorithmic skeletons* representam conhecimento adquirido/boas práticas ou eventos recorrentes (e.g. capturados pela observação de um sistema em execução), numa determinada área. Tendo em vista a simplificação de programas no domínio da programação paralela, têm sido desenvolvidas diversas ferramentas e linguagens de programação que permitem uma programação estruturada baseada em *algorithmic skeletons* (por vezes também designados *parallel patterns*). Um resumo destas ferramentas foi compilado em [2].
O objectivo deste trabalho é a utilização de uma destas ferramentas na programação de um problema no domínio das Ciências dos Materiais.

FastFlow

A ferramenta sugerida, denominada *FastFlow* [3], destina-se à programação paralela no contexto de arquitecturas *multi-core*, estando particularmente vocacionada para o desenvolvimento de *streaming applications*. Este tipo de aplicações são estruturadas tendo como base alguma forma de "stream"/fluxo de dados, por exemplo, aplicações video e aplicações para processamento de imagem.
A ferramenta *Fastflow* permite a utilização dos padrões *farm* e *pipeline* que são básicos em computação paralela.

Identificação de Objectos em Imagens

Os especialistas de Engenharia de Materais procuram desenvolver novas formas de produzir materiais compostos, em que o comportamento de um material base (*matriz*) é modificado pela introdução de outro material (*reforços*). Para avaliar a qualidade de um processo de fabrico de um composto são obtidas imagens 3D de uma amostra do material num tomógrafo; cada imagem corresponde a uma matriz 3D armazenada num ficheiro em que cada byte corresponde a um nível de cinzento; esse nível de cinzento representa a densidade do material no ponto correspondente na amostra. Idealmente teríamos uma imagem a preto e branco em que:

- Ao nível de cinzento 255 corresponde o branco: menor densidade, a matriz ou o material original
- Ao nível de cinzento 0 corresponde o preto: maior densidade, os reforços

Na matriz 3D que sai do tomógrafo cada elemento (voxel) pode tomar valores de 0 a 255 correspondendo a níveis de cinzento. É preciso tornar esta imagem monocromática de forma a que os reforços fiquem claramente separados; após este processo pode-se fazer a caracterização da população de reforços.
Supondo que existe uma diferença de densidade significativa entre a matriz e os reforços é significativa, a obtenção da imagem monocromática passa por três etapas:

1) *Construção de um histograma* -- para todas as cores possíveis (do preto ao branco, i.e. do valor 0 ao 255) pretende-se contar quantos voxels na imagem apresentam essa cor. O objectivo é a construção de um histograma de modo a identificar dois valores de tom de cinzento L1 e L2 que melhor permitam distinguir os dois tipos de materiais numa amostra. Assim, o valor L1 representa um aglomerado de voxels com cores mais próximas do preto correspondente a um tipo de material, e o L2 do branco, correspondente a outro tipo de material. O objectivo da obtenção dos dois valores é permitir transformar a imagem original numa outra em que existe um maior contraste entre os dois tipos de material, transformando todos os voxels em voxels apenas com cores preta ou branca, usando os dois algoritmos seguintes.

2) *Algoritmo da "bi-segmentação"* -- este algoritmo é extensivamente usado na identificação de objectos em imagens na área das Ciências dos Materiais, e parte de dois valores L1 e L2, obtidos a partir de uma análise inicial da imagem, tal como descrito em 1). O algoritmo consiste no processamento de todos os voxels da imagem tal que:

se um voxel tiver uma cor inferior ao valor L1, o voxel fica com a cor preta;
senão, se o voxel tiver uma cor de valor maior ou igual a L1 e menor que L2, o voxel fica com a cor cinzenta;
senão, o voxel fica com a cor branca.

3) *Histerese* -- partindo de uma imagem apenas em tons de branco, preto e cinzeto, por exemplo obtida com o algortimo em 2), pretende-se chegar a uma imagem apenas com tons de preto e branco. Assim, para cada voxel cinzento, são avaliados os seus voxels vizinhos. Para simplificar vamos admitir que o voxel x,y,z tem 6 vizinhos (vizinhança de faces):

- O "de cima": (x,y,z-1)
- O "de baixo": (x,y,z+1)
- O "do lado da frente": (x, y+1, z)
- O "do lado de trás": (x, y-1, z)
- O "do lado esquerdo": (x-1, y, z)
- O "do lado direito": (x+1, y, z)

Após analisar os vizinhos do voxel, o algoritmo a efectuar é o seguinte:

- se um voxel cinzento tiver uma maioria de voxels vizinhos de cor preta, o voxel fica com a cor preta;
- senão, so o voxel tiver uma maioria de voxels com a cor branca, o voxel fica com a cor branca;
- senão, o voxel fica com a cor cinzenta.

Este processo é repetido até que não existam mais voxels de cor cinzenta, ou não seja possível reduzir o número de cinzentos.

Quando a algoritmo acima descrito não consegue eliminar mais cinzentos, passa-se a uma estratégia mais radical:

- se um voxel cinzento tiver mais voxels vizinhos de cor preta do que branca, o voxel fica com a cor preta;
- senão, so o voxel cinzento tiver mais vizinhos brancos do que pretos fica com a cor branca;

- senão fica preto ou branco de forma aleatória

Ver detalhes sobre estas várias fases na referência [4].

Objectivo do trabalho

Dada uma imagem em Ciências dos Materiais pretende-se que construa um programa que implemente os algoritmos 1) e 2) em cima, i.e. a construção do histograma da imagem fornecida para determinação do L1 e L2, bem como a transformação da imagem usando o algoritmo da "bi-segmentação". **A terceira etapa, a Histerese, neste trabalho não é implementada**.

O seu programa deve tere em consideração o desempenho, e asim tirar partido da programação paralela possibilitada pelo uso da ferramenta *FastFlow*, apresentada na aula prática.

Data de Entrega

O trabalho deve ser entregue até ao final do dia ***7 de Junho***, 6a. feira, e deve ser enviado por email para:

mcg@fct.unl.pt

O cabeçalho do email deve conter a informação:

trabalho skeletons CPD

e o trabalho deve ser entregue num zip com o nome e número do aluno.

Bibliografia

[1] Murray Cole. 1991. Algorithmic Skeletons: Structured Management of Parallel Computation. MIT Press, Cambridge, MA, USA.

[2] Horacio González-Vélez and Mario Leyton. 2010. A survey of algorithmic skeleton frameworks: high-level structured parallel programming enablers. *Softw. Pract. Exper.* 40, 12 (November 2010), 1135-1160. DOI=10.1002/spe.v40:12 http://dx.doi.org/10.1002/spe.v40:12

[3] FastFlow. http://calvados.di.unipi.it/dokuwiki/doku.php?id=ffnamespace:about

[4] Fernando Birra, Magda Encarnação, Adriano Lopes, Pedro Medeiros, Nuno Oliveira, Bruno Preto, Paulo Quaresm and Alexandre Velhinho. 2013. Tomographic image analysis of reinforcement distribution in composites using a flexible and material's specialist-friendly computational environment. Submetido para publicação, *Journal of Synchrotron Radiation*, IUCr (disponível na página da cadeira no clip, na secção "Outros").