

# Arquiteturas Paralelas e Distribuídas

Bacharelado em Ciência da Computação

Prof. Dr. Eduardo Takeo Ueda

2º Trabalho

## Mochila binária com MPI

Data de entrega limite: 15 de junho de 2015

O problema da mochila binária (*0-1 knapsack*) é um problema clássico de otimização. O cenário do problema é o seguinte: considere uma mochila que suporta um peso máximo  $\mathbf{C}$  e considere também um conjunto de  $\mathbf{N}$  objetos, cada um com um peso e um valor. O objetivo do problema é escolher um subconjunto de objetos a carregar na mochila, de forma a preenchê-la com o maior valor possível, sem ultrapassar o peso máximo.

Uma forma exaustiva de resolver este problema seria testando todas as combinações possíveis. Na prática, geralmente utiliza-se alguma técnica para abreviar a computação. No programa sequencial disponível em <http://goo.gl/dqrezK> (e também no BlackBoard), o problema da mochila é resolvido usando uma matriz  $\mathbf{N} \times \mathbf{C}$ , onde cada elemento  $(i, j)$  armazena o valor máximo na mochila para um peso  $j$ , usando objetos de 0 a  $i$ . Os elementos da matriz vão sendo calculados de forma gradual, com novos elementos aproveitando valores dos elementos anteriores.

O principal objetivo deste trabalho é aplicar MPI para paralelizar o programa sequencial fornecido, buscando obter o máximo ganho de desempenho possível.

Baixe o pacote **knapsack.tar.gz**. Descompacte ele e execute o programa sequencial para o problema default:

```
tar xzvf knapsack.tar.gz
cd knapsack
make knap
./knap
```

Execute o programa para 100.000 objetos e capacidade (peso) máxima 5.000, como descrito abaixo.

```
./knap -b 100000 -c 5000
```

Observe atentamente o pacote fornecido e perceba que:

- (i) Os objetos são livros;
- (ii) O programa **knap.c** é instrumentado para medição de tempos;
- (iii) Existem diferentes opções para geração de pesos e valores dos objetos ;
- (iv) Existem arquivos que são desnecessários para este trabalho.

Estude cuidadosamente o código **knap.c** com o intuito de entender a solução sequencial. Depois implemente uma versão paralela aplicando recursos do MPI (MPICH). Não esqueça de certificar-se que ambas as implementações (sequencial e paralela) produzem o mesmo resultado.

### Recomendações importantes

1. O trabalho pode ser feito em dupla ou individualmente.
2. O código desenvolvido (em linguagem C) deve compilar com o GCC (GNU Compiler Collection) e MPI (MPICH), além de executar no ambiente Linux (Ubuntu 14.10).
3. Você deve submeter no BlackBoard um arquivo .rar (ou .zip), contendo o código-fonte da implementação com MPI e um arquivo README (com instruções de como compilar, executar e utilizar o programa), e identificado com a concatenação do seu primeiro nome com seu sobrenome. Por exemplo, meu nome é Eduardo Takeo Ueda, então devo submeter um arquivo chamado EduardoUeda.rar (ou .zip).
4. Não esqueça de deixar comentários explicando seu código-fonte, pois isso será levado em consideração na correção do trabalho.
5. Não serão aceitos trabalhos atrasados e nem entregues por e-mail, apenas pelo sistema BlackBoard. Então, não deixe para fazer o trabalho na última hora, como na véspera da data de entrega.

**Bom trabalho!**