

Trabalho Prático 6

Este trabalho prático tem por objetivo exercitar questões ligadas a paralelização de algoritmos. Sendo assim, o aluno deverá paralelizar, utilizando primitivas da biblioteca *pthread*, o algoritmo criado para solucionar o problema abordado no tp0. É necessário que a eficiência da paralelização implementada seja avaliada.

1 Definição do Problema

Este trabalho prático consiste em projetar um algoritmo paralelo para ser executado em uma máquina paralela. O algoritmo deve solucionar o problema clássico do Casamento Estável (*Stable Marriage Problem*, SMP), objeto do trabalho prático 0. Apenas lembrando, o SMP consiste em encontrar um *matching* estável entre dois conjuntos distintos de elementos. Os conjuntos têm o mesmo tamanho e cada elemento possui uma lista de preferências sobre o outro conjunto. Para maiores informações, consulte a especificação do tp0.

Para esse trabalho, **DEVEM** ser exploradas duas oportunidades de paralelização: dados e tarefas. A documentação do trabalho **DEVE** definir e discutir as oportunidades de paralelização observadas, bem como apresentar a avaliação do algoritmo em relação a diferentes configurações de instâncias do problema para um número de *threads* variado. Logo, a documentação final do trabalho deve apresentar:

- Descrição detalhada das estratégias de paralelismo utilizadas.
- Detalhamento dos tipos de dados utilizados.
- Listagem das funções implementadas.
- Definição da estratégia para gerar variadas configurações de entrada do problema.
- Análise experimental e discussão dos resultados sobre os cenários testados.

DEVE ser realizada uma comparação entre a versão sequencial (TP0) e as versões paralelizadas (TP6) para diferentes configurações de entrada, com comparação do valor obtido para o *speedup*. Além disso, **DEVE** ser analisado o comportamento da solução paralelizada em função do número de *threads*. Dessa forma, é possível descobrir empiricamente o comportamento assintótico do *speedup* do seu algoritmo paralelo.

1.1 Entrada e Saída

O arquivo executável deve ser chamado de *tp6* e deve receber como parâmetro o arquivo de entrada *input.txt*, o arquivo de saída *output.txt* e o número de threads *<num-threads>*, conforme demonstrado a seguir:

```
./tp6 -i input.txt -o output.txt -t <num-threads>
```

O arquivo *input.txt* deve conter as seguintes informações, na ordem indicada abaixo:

1. número de instâncias do problema;
2. tamanho n dos conjuntos considerados;
3. n linhas com as preferências de cada homem;
4. n linhas com as preferências de cada mulher.

Cada indivíduo é representado por um identificador único (*id*), por sexo. Cada *id* é um valor inteiro entre 1 e n . Sendo assim, cada linha possui todos os *ids* do sexo oposto ordenados pela preferência do usuário correspondente.

Exemplo de entrada para 8 indivíduos (4 homens - *ids* de 1 a 4, e 4 mulheres - *ids* de 1 a 4):

```
1 // número de instâncias.
4 // número n de indivíduos em cada conjunto.
2 4 1 3 // homem de id = 1 prefere mulher de id = 2, seguido da mulher de id = 4, e assim por diante.
4 1 2 3
2 3 4 1
2 3 4 1
2 3 4 1 // mulher de id = 1 prefere homem de id = 2, seguido do homem de id = 3, e assim por diante.
3 4 2 1
3 2 1 4
3 2 4 1
```

A saída do programa deve ser impressa em um arquivo *output.txt* e deve conter as seguintes informações, na ordem indicada abaixo:

1. indicação dos casamentos finais de todos os indivíduos;
2. valor da satisfabilidade* geral dos indivíduos;
3. valor da satisfabilidade masculina;
4. valor da satisfabilidade feminina;

***Satisfabilidade** é uma métrica expressa como a média das posições que os maridos e/ou esposas ocupam na lista de preferências de seus respectivos parceiros. Isto é, se uma mulher se casa com seu pretendente preferido, ela contribui em 1 para a satisfabilidade. No entanto, caso ela se case com a terceira opção de sua lista de preferências, sua contribuição será de 3. Sendo assim, quanto menor o valor dessa métrica, maior a satisfação dos indivíduos considerados.

O valor das satisfabilidades devem ser apresentados com precisão de 3 casas decimais, como no exemplo de saída abaixo.

Exemplo de saída genérico (arquivo *output.txt*):

```
2 4 // homem de id = 2 se casou com mulher de id = 4.
1 1
3 2
4 3
2.250 // satisfabilidade geral.
1.750 // satisfabilidade masculina.
2.750 // satisfabilidade feminina.
```

Entrada e saída padrão devem seguir rigorosamente o formato descrito. Instâncias distintas para o problema devem ser geradas pelo próprio aluno para testar e avaliar seu algoritmo.

2 O que deve ser entregue:

2.1 Documentação: deve abranger pelo menos os seguintes pontos

- Introdução do problema apresentado.
- Modelagem e solução do problema.
- Complexidade de tempo e espaço.
- Principais decisões de implementação.
- Análise experimental.
- A documentação **não** pode exceder 10 páginas.

2.2 Código:

- O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios de graduação do DCC;
- Deve ser **obrigatoriamente** escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras **não** serão aceitos);
- As estruturas de dados devem ser alocadas dinamicamente e o código deve ser modularizado (ou seja, dividido em múltiplos arquivos fonte e fazendo uso de arquivos cabeçalho - .h);
- O utilitário Make deve ser utilizado para compilar o programa (**o arquivo de makefile deve ser submetido juntamente com o código fonte**);
- A saída deve ser impressa seguindo estritamente o formato da especificação, caso contrário o resultado será considerado errado;
- O arquivo executável deve ser chamado de **tp6**. Não serão aceitos outros nomes de executáveis além do mencionado;
- Faça seu código de forma legível.

2.3 Entrega:

- Data de entrega: 03/07/2012
- Submissão: a documentação e o código do trabalho devem ser submetidos ao *minha.ufmg*. Para isso, compacte os dois (formato *tp6_NomeSobrenome.zip*) e faça a submissão. Teste seu arquivo compactado antes de enviá-lo.
- Apenas a documentação deve ser entregue impressa na secretaria do DCC. Não coloque nos escaninhos dos professores, entregue para a secretária para que sua documentação seja colocada no envelope de AEDS3. A documentação impressa pode ser entregue no dia útil seguinte da submissão digital. **Trabalhos que não tiverem a documentação entregue na secretaria, dentro do prazo de entrega, receberão nota 0.**
- Será postada uma planilha no Moodle sobre a entrevista do trabalho, leia-a e siga as orientações para o agendamento da sua entrevista.
- Será adotado média harmônica entre a pontuação obtida na execução e na documentação do TP, o que implica em valor zero caso alguma das partes não seja apresentada.
- A política para desconto por atraso de entrega do trabalho prático considera a fórmula:

$$\frac{2^d - 1}{0.32} \%$$

onde d é o atraso em dias úteis. Note que após 5 dias úteis, o trabalho não pode ser mais entregue.

Observações

- O formato de submissão **DEVE** ser respeitado: *tp6_NomeSobrenome.zip*. Por favor, **NÃO** submetam trabalhos em outros formatos (.rar, .tar, .tar.gz). Certifique-se que seu nome completo foi incluído no nome do arquivo.
- **EXCLUA** o arquivo executável do arquivo compactado a ser submetido no *minha.ufmg*.
- **INCLUA** seu email na capa da documentação.

Referências

Parallel Computing Theory and Practice. M. J. Quinn. McGraw-Hill, 1994.