TRABALHO 0:
PROGRAMAÇÃO PARALELA VIA BMF
FINAL

RAUL H.C. LOPES

1. Preliminares

Neste trabalho você implementará um conjujnto de algoritmos que podem que têm o fato de que podem ser facilmente derivados usando os formalismos de *Bird-Meertens* e *powerlists*. O trabalho consiste de exercícios sobre ordenação paralela e exercícos de cálculo de seqüências.

Algumas dicas importantes sobre a execução do trabalho seguem.

(1) Não deixe de ler este documento por completo antes de iniciar o trabalho.
(2) Siga estritamente as especificações deste documento: qualquer desvio delas pode significar a anulação de um exercício ou de todo o trabalho.
(3) Comece a trabalhar de imediato: o trabalho foi concebido para ser realizado em quatro semanas. Depois disso, você terá outro trabalho e nova prova.

O trabalho pode ser executado em grupos, mas, como estabelecido na seção 4.1, grupos maiores recebem menos créditos.

2. Cálculo de seqüências

Exercício 1. *Implemente o algoritmo para cálculo de todos os primos menores do que um inteiro, apresentado em sala no dia 14/05/06.*

Seu programa compilado terá o nome ***primes****.*

Seu programa receberá um único argumento na linha de comando: o inteiro limite superior dos primeiros a gerar. Ele produzirá os primeiro em stdout*, um por linha.*

Exercício 2. ***Opcional.*** *Apresente testes similares aos do ex. 9.*

Exercício 3. *Implemente um dos algoritmos de Vishkin and Vishkin, veja em Experiments with list ranking for explicit multi-threaded (XMT) instruction parallelism, para cálculo dos* ranking *de uma lista. Escolha entre os algoritmos* No-Cut *e* Cut-6*.*

Seu programa compilado terá o nome ***vishkin****.*

Seu programa receberá um único argumento na linha de comando: o número de nós na lista. Ele receberá o vetor de índices de sucessores de nós de stdin*, um por linha, e exibirá em* stdout*, um por linha, os rankings calculados.*

Exercício 4. *Implemente o algoritmo de Helman-Jájá, veja em Prefix Computations on Symmetric Multiprocessors, para cálculo dos prefixos de uma lista encadeada de inteiros.*

Seu programa compilado terá o nome ***hjaja****.*

Seu programa receberá um único argumento na linha de comando: o número de nós na lista. Ele receberá o vetor de índices de sucessores de nós de stdin*, um por linha, e exibirá em* stdout*, um por linha, os rootings calculados. Note que você estará calculando, via soma de prefixos, a distância de cada nó até o início da lista.*

Exercício 5. ***Opcional.*** *Apresente testes similares aos do ex. 9.*

3. Ordenação

Exercício 6. *Implemente o algoritmo de radix sort paralelo apresentado em sala para ordenar inteiros positivos.*

Seu programa compilado terá o nome ***radix****.*

Seu programa receberá um único argumento da linha de comando: o número de inteiros a ordenar. Ele lerá os inteiros de stdin*, um por linha, e exibirá o resultado da ordenação em* stdout*, um por linha.*

Exercício 7. *Implemente uma versão paralela não recursiva do algoritmo* bitonic sort*.*

Seu programa compilado terá o nome ***bitonic****.*

Seu programa receberá um único argumento da linha de comando: o número de inteiros a ordenar. Ele lerá os inteiros de stdin*, um por linha, e exibirá o resultado da ordenação em* stdout*, um por linha.*

Exercício 8. *Implemente uma versão não recursiva do algoritmo de* merge sort *paralelo de Cole, veja em Parallel merge sort.*

Seu programa compilado terá o nome ***merge****.*

Seu programa receberá um único argumento da linha de comando: o número de inteiros a ordenar. Ele lerá os inteiros de stdin*, um por linha, e exibirá o resultado da ordenação em* stdout*, um por linha.*

Exercício 9. *Apresente quadro de testes comparativo dos algoritmos dos exercícios 6, 7,8. Seu quadro abordará os seguintes testes:*

- *Para um número fixo de inteiros (ao menos,* 2^{20}*), os tempos de execução para 4, 6, 8, 16 processadores.*
- *Para 16 processadores, os tempos de execução para* $2^{16}, 2^{20}, 2^{24}$ *itens.*

Fator	Multiplicador
Grupos de 1 aluno	110/100
Grupos de 2 alunos	100/100
Grupos de 3 alunos	75/100
Grupos de 4 alunos	60/100
Grupos de mais 4 alunos	0
Plágio	0

TABELA 1. Tabela de multiplicadores

4. CRÉDITOS

A definição dos critérios de avaliação A atribuição de nota para o trabalho ocorrerá em duas fases:

(1) Atribuição de créditos pelo trabalho submetido, de acordo com critérios apresentados na sc. 4.1.
(2) Definição do multiplicador de acordo com a tabela 1.

4.1. **Definição do número de créditos.** Todos os exercíos, com a exceção dos opcionais, têm o mesmo peso. Os exercícios opcionais substituem até metade do valor de qualquer exercício não concluído deste trabalho.

4.2. **Definição de multiplicador.** A nota de cada trabalho é dada pelo número de créditos obtidos pelo trabalho multiplicado pelo fator da tabela 1.

Em caso de plágio todos os trabalhos de grupos envolvidos são anulados. Poderá ser enquadrado como plágio qualquer trabalho em que um mais alunos do grupo não tenham conhecimento de qualquer item do trabalho. Esse conhecimento deverá ser demonstrado em prova e/ou enttrevista.

5. A SUBMISSÃO PARA CORREÇÃO

Submeta sua trabalho por e-mail. Corrigirei quaisquer trabalhos submetidos até às 12 horas do dia 11/08/05.

A mensagem de submissão do seu trabalho deverá conter:

- **Subject: pp.tp0**
- *Attachment*: **tp0.tar.bz2**

O corpo da mensagem será desprezado. O arquivo anexdo deverá ser obrigatoriamente denominado

tp0.tar.bz2

e conterá todo os fontes necessários para compilar seu trabalho prático, incluindo fontes em LaTeX, **Haskell**, **Makefile**e arquivo de identificação.

As seguintes regras devem ser rigorosamente seguidas:

- Nenhum compilado deve ser enviado.
- O *tarball* do seu trabalho conterá a seguinte estrutura:
  - arquivo de **id**
    Conterá uma linha inicial com *e-mail* de contato do grupo e, depois, uma linha de identificação para elemento do grupo, contendo número de matrícula e nome completo separados por '**:**' (dois pontos.) Por exemplo:
    ```
    jolero@gmail.com
    99900089:Ze' do lero
    ```
  - arquivo **Makefile**
  - diretório **doc**
    Conterá a descrição e resultados dos testes realizados. Esses resultados podem ser apresentados em formato texto. Nao leio nada que venha em formato proprietário.
- seu *tarball* será aberto e compilado com as seguintes linhas:
  ```
  tar -jxf tp0.tar.gz
  make all
  ```