

## 18 - Atividade 3

Super Computação 2018/2

Igor Montagner, Luciano Soares

**Entrega:** 29/10

Vamos trabalhar hoje com dados complexos usando MPI. Mais especificamente, vamos fazer processamentos de imagens distribuídos usando MPI.

**Exercício:** Os arquivos *image.c/h* disponibilizam uma interface para leitura e escrita de imagens. Aprenda a usá-los criando um programa *copy.cpp* que recebe dois argumentos na linha de comando: o caminho para uma imagem fonte e seu novo nome.

### Limiar adaptativo - exercício 1

O limiar adaptativo converte uma imagem em níveis de cinza para uma imagem binária (0 para preto, 255 para branco) de acordo com a seguinte regra:

1. Para cada ponto da imagem, calcule a média de uma região  $3 \times 3$  ao seu redor.
2. Se o valor do pixel for maior que a média, sete ele como branco na imagem de saída
3. Se não sete ele como preto na imagem de saída.

**Exercício preliminar:** como você faria a troca de mensagens entre os processos? Mostre abaixo usando um diagrama.

**Exercício:** calcule o limiar adaptativo de uma imagem usando MPI levando em conta os seguintes pontos:

1. somente o processo mestre (rank 0) poderá ler a imagem do disco. Os processos restantes deverão receber a imagem via passagem de mensagens.
2. você deverá dividir a imagem em vários pedaços, de modo que cada processo processe uma parte da imagem.
3. tome cuidado com as bordas das suas divisões.

## Processamentos em paralelo - exercício 2

O arquivo *edge.c* possui uma implementação de um filtro de bordas. Gostaríamos agora de criar um programa *MPI* que, dada uma lista de imagens passadas pela linha de comando, aplica este filtro em cada uma delas, salvando o resultado com o nome original mais “\_edges”.

A restrição do exercício acima continua valendo: somente o processo mestre (rank 0) pode ler/escrever as imagens do/para o disco.

**Exercício preliminar:** como você faria a troca de mensagens entre os processos? Mostre abaixo usando um diagrama.

**Exercício:** resolva este problema usando passagem de mensagens síncronas.

**Exercício:** resolva este problema usando passagem de mensagens assíncrona. Onde foi possível obter ganhos de desempenho?