

# PCP - Trabalho 2

Matheus Ambrozio & Renan Almeida

October 24, 2018

O trabalho foi desenvolvido conforme a especificação. A três diferentes versões do código podem ser encontradas na pasta `/src` em `/problem-01`, `/problem-02/version-01`, e `/problem-02/version-02`.

A função usada para teste está definida em `/src/weird.c`. Utilizamos a função `weird1` para os testes. Calculamos a área usando o intervalo  $[-3, 3]$ . A função matemática correspondente é:

$$f1(x) = (x - 1) * (x + 3)^2 * (x - 2)^3 * (x + 1)^4 \quad (1)$$

## Medições

Realizamos os testes nos nós n00-n07, cada nó com 4 núcleos e possibilidade de conter no máximo 4 processos. As medições são geradas através da execução do script `tester.lua`.

As medições foram feitas usando a função `MPI_Wtime`. Abaixo, temos a tabela com os resultados em segundos. A coluna na esquerda (1, 2, 4, 8, 16) indica o número usado para `-n`. A linha superior indica a qual das versões do trabalho pertence a medição. No caso da primeira versão do segundo problema, o número de sub-intervalos (32, 128, 512, 1024) está indicado entre parenteses.

	<b>P1</b>	<b>P2-V1 (32)</b>	<b>P2-V1 (128)</b>	<b>P2-V1 (512)</b>	<b>P2-V1 (1024)</b>	<b>P2-V2</b>
<b>1</b>	4.628	-	-	-	-	-
<b>2</b>	3.376	5.116	5.295	5.311	4.576	35.040
<b>4</b>	2.626	1.825	1.631	1.877	1.840	14.191
<b>8</b>	2.025	1.166	0.860	0.770	0.803	59.562
<b>16</b>	-	1.028	0.697	0.532	0.468	154.004

A execução mais rápida foi o **P2-V1 (1024)**, terminando em 0.468 segundos, que é a que mais divide o intervalo à ser calculado, tornando os trabalhos de cada nó mais fáceis. Vemos que os piores tempos de execução são do programa utilizando a solução do problema 2 versão 2, onde há mais troca de mensagens entre os nós.

# Desenvolvimento

O código do trabalho está dividido entre três diferentes arquivos `main.c`, contendo as diferentes versões do programa.

- Para **P1**, o programa faz apenas um `MPI_Reduce` para sincronizar os dados e calcular a área total. Sua implementação é bem simples.
- Para **P2-V1**, nós mantemos no nó mestre uma lista de intervalos a serem calculados, que é esvaziada conforme nós trabalhadores requisitam novos intervalos para calcular. O programa termina quando o contador `splits`, que supervisiona a quantidade de intervalos calculados, chega a zero.
- Para **P2-V2**, a lógica é similar à da versão anterior, com a diferença da existência de um novo tipo de mensagem `NEW_INTERVAL`, que incrementa a variável `splits` e adiciona um novo intervalo na lista do nó mestre. Essa mensagem é enviada por um nó trabalhador quando ele divide um intervalo em dois.

O código se encontra em: <https://github.com/mambrozio/pcp-assignment-02>