

PCP - Trabalho 3

Matheus Ambrozio & Renan Almeida

October 23, 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`.

As funções usadas para teste estão definidas em `/src/weird.c`. A função `weird1` é bastante lenta, portanto, usamos a função `weird2` para as medições. Para ambas as funções, calculamos a área usando o mesmo intervalo $[-5, 5]$. As funções matemáticas correspondentes são:

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

$$f2(x) = e^{\sin(x)} \quad (2)$$

Medições

Realizamos os testes e medições somente no nó mestre do cluster (devido aos problemas de execução já discutidos). Como esse nó só possui 4 núcleos, os testes com `-np` maior que 4 não puderam ser propriamente avaliados. 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 `-np`. 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.

	P1	P2-V1 (32)	P2-V1 (128)	P2-V1 (512)	P2-V1 (1024)	P2-V2
1	0.231	-	-	-	-	-
2	0.130	0.232	0.246	0.247	0.194	4.673
4	0.069	0.075	0.070	0.066	0.070	2.155
8	0.097	0.106	0.163	0.181	0.211	4.983
16	-	0.114	0.153	0.248	0.282	11.496

A execução mais rápida foi o **P2-V1 (512)**, terminando em 0.066 segundos. Percebemos que todas as versões executam com tempo semelhante para `-np 4`, a exceção da **P2-V2**.

Acreditamos que a quantidade elevada de troca de mensagens, aliada ao baixo número de **workers**, sejam causa parcial para essas medições mais lentas. Para investigar melhor, gostaríamos de testar os mesmo problema usando vários nós com vários núcleos.

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.