

Programação Concorrente e Distribuída - Integral

Atividade 9 - Programação em MPI

Rosangela Miyeko Shigenari 92334

1. Considere o código sequencial para calcular numericamente  $\ln 2$ :

```
#include <stdio.h>

#define N 100000000
#define u 2.0

int main(void) {
    double passo, soma,x;
    int i;

    passo = (u-1) / (double) N;
    for (i=0; i<N;i++) {
        x=1+i*passo;
        soma=soma+0.5*(1/x+1/(x+passo));
    }
    printf("ln %f = %20.15f\n", u, passo*soma);
    return 0;
}
```

Faça um programa em linguagem C com MPI para o código acima, dividindo a soma igualmente por todos os processos. As somas parciais devem ser enviadas ao processo 0, que deve calcular o resultado final e indicar o tempo total gasto em milissegundos.

Utilize o número de processos igual ao número de processadores. Teste com 1, 2, 4, 6 e 8 processos (acima de 4 deve-se usar pelo menos duas máquinas interconectadas em rede). Mostre o tempo de execução, speedup e eficiência atingidos na forma de gráficos.

2. O programa anexo no moodle chamado "ftcs.c", faz o cálculo numérico de um problema unidimensional de condução de calor por diferenças finitas através do método FTCS. Deve-se construir uma versão distribuída deste código usando MPI. Teste o programa concorrente com 1, 2, 4, 6 e 8 processos (acima de 4 deve-se usar pelo menos duas máquinas interconectadas em rede). Mostre o tempo de execução, speedup e eficiência atingidos na forma de gráficos.

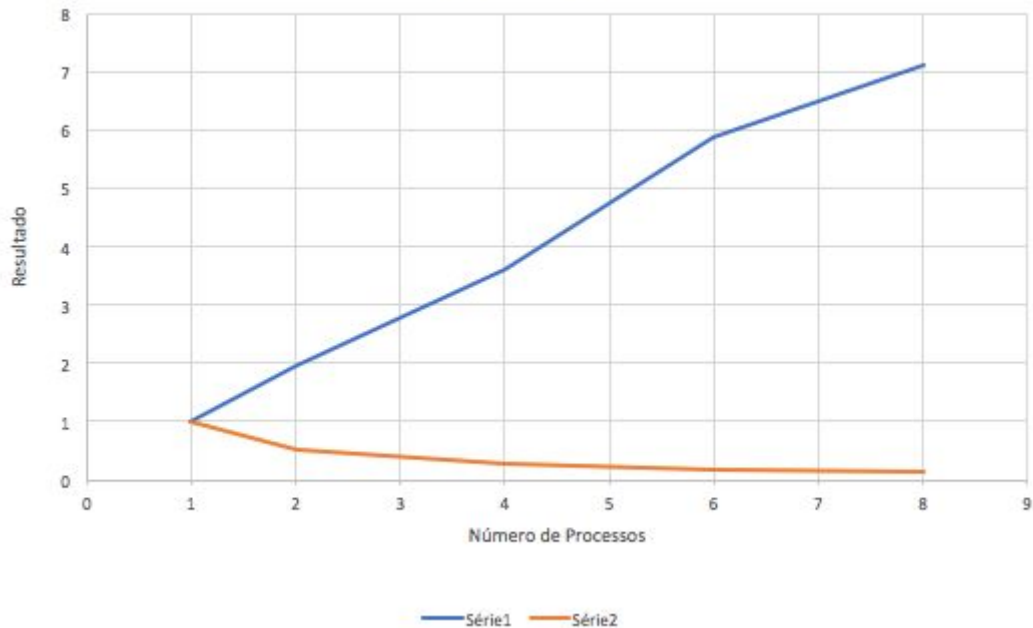
Obs: Cuidado para evitar deadlocks com a troca de dados entre processos adjacentes.

1) Código do programa em anexo como atv9.c

Ao executar o programa foram obtidos os seguintes resultados, mostrados na tabela abaixo.

Processos	Tempo (ms)	SpeedUp	Eficiência
1	1640.54	1	1
2	838.73	1.96	0.51
4	453.22	3.59	0.28
6	441.68	5.88	0.17
8	436.51	7.1	0.14

Após a construção da tabela foi possível plotar o gráfico que relaciona número de processo com o *SpeedUp* e eficiência.



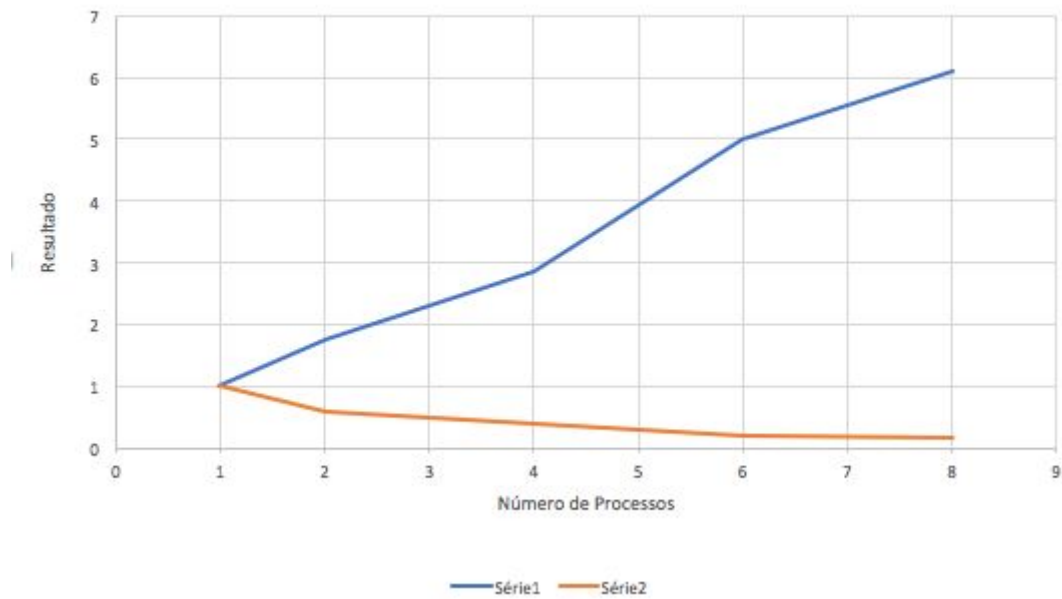
A série 1 corresponde ao *SpeedUp* e a série 2 a eficiência.

2) Código do programa em anexo como `atv92.c`

Ao executar o programa foram obtidos os seguintes resultados, mostrados na tabela abaixo.

Processos	Tempo (s)	SpeedUp	Eficiência
1	6.52	1	1
2	3.98	1.76	0.57
4	3.66	2.85	0.39
6	3.43	4.99	0.2
8	3.28	6.09	0.16

Após a construção da tabela foi possível plotar o gráfico que relaciona número de processo com o *SpeedUp* e eficiência.



A série 1 corresponde ao *SpeedUp* e a série 2 a eficiência.

### Considerações Finais

A partir da análise dos resultados foi possível concluir que o aumento no número de processos aumentava o *speedUp*, devido a paralelização da tarefa pelos processos correntes. Durante a execução, necessário rodar com o Slurm de 2 4 4, para que a eficiência não fosse perdida, a tentativa de rodar com um valor acima de 4, diminuiu a eficiência.