



**UNIVERSIDADE FEDERAL DE SÃO PAULO
INSTITUTO DE CIÊNCIA E TECNOLOGIA
BACHARELADO EM CIÊNCIA E TECNOLOGIA**

Laboratório 05

Programação em MPI

Nomes: Ana Júlia de Oliveira Bellini
Willian Dihanster Gomes de Oliveira

RA: 111774
RA: 112269

**SÃO JOSÉ DOS CAMPOS
2018**

Especificações da Máquina

Para os experimentos realizados em cada um dos exercícios propostos, as especificações da máquina utilizada são descritas a seguir:

Processador: Intel (R) Core (TM) i5-7200U CPU @ 2.50GHz

Núcleos físicos: 2

Memória Cache: 3Mb L3

Memória RAM: 8 GB

Threads: 4

Hyperthreading: Sim

Sistema Operacional: Ubuntu 18.04.1 64 bits

Compilador: gcc 7.3.0

Exercício 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.

Resultados do Exercício 1

Para o primeiro exercício, foram obtidos os seguintes resultados:

Tempo de Execução, Exercício 1 (ms)

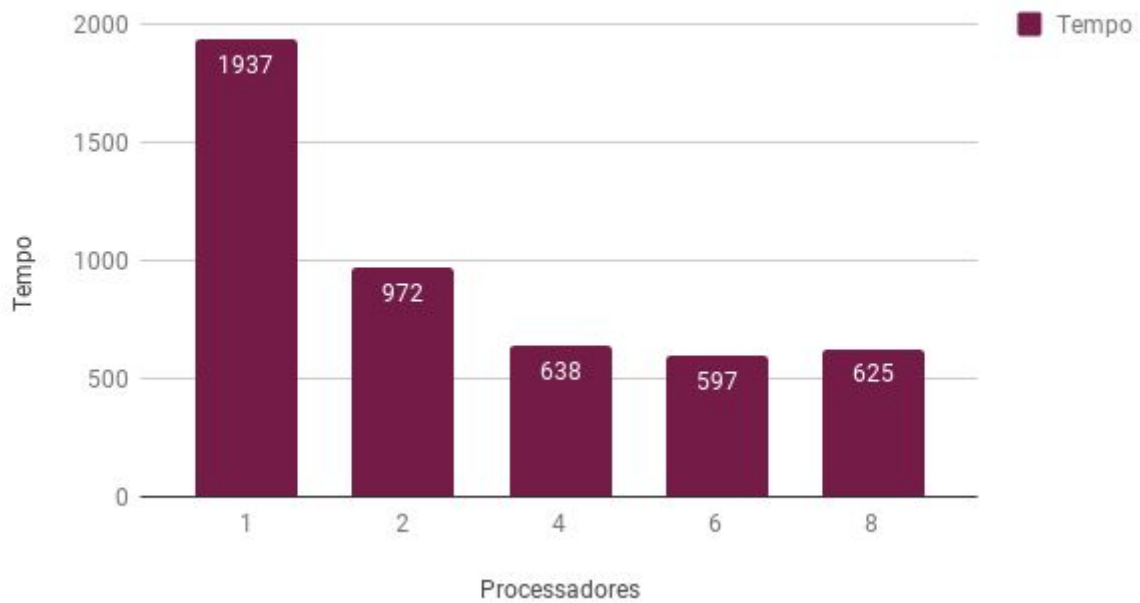


Figura 1: Resultados de tempo de execução obtidos no Exercício 1.

Speedup, Exercício 1

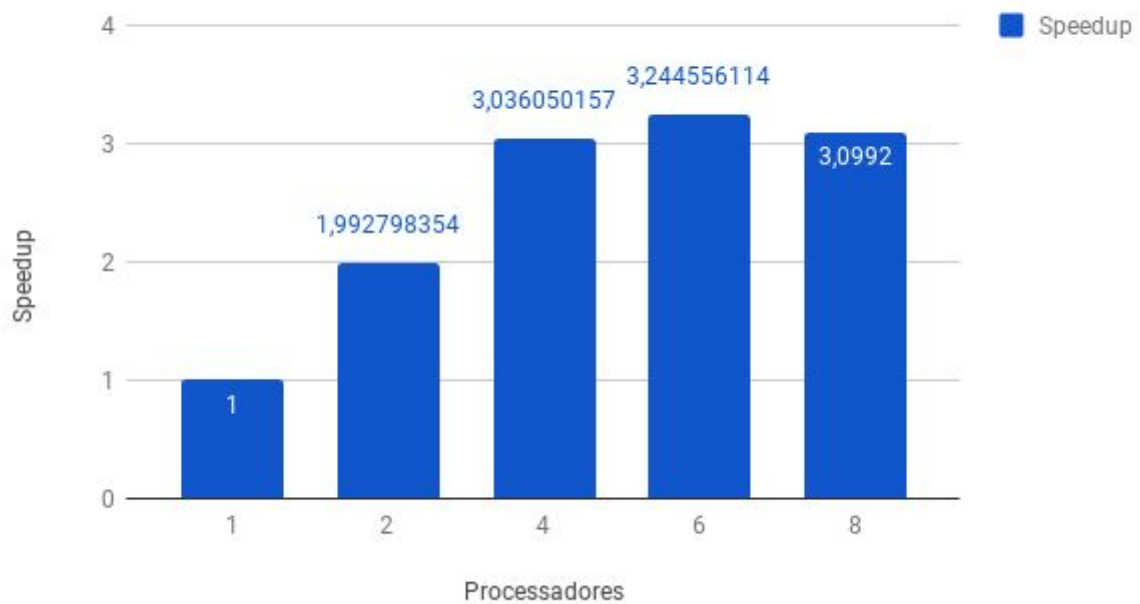


Figura 2: Resultados de Speedup de execução obtidos no Exercício 1.

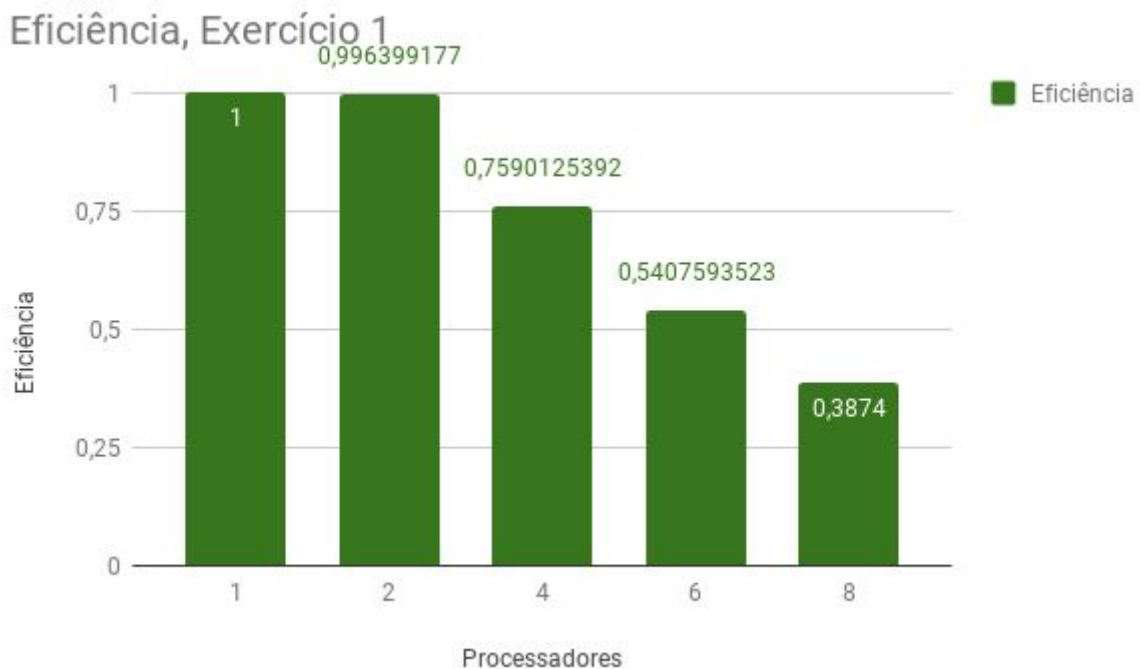


Figura 3: Resultados de eficiência de execução obtidos no Exercício 1.

```

Processo 2 tem bloco tam = 25000000
Processo 3 tem bloco tam = 25000000
Processo 1 tem bloco tam = 25000000
Processo 1 tem soma_local = 0.182322
Processo 0 tem soma_local = 0.223144
Processo 2 tem soma_local = 0.154151
Processo 3 tem soma_local = 0.133531
Ln(2.0) = 0.693147180559914
Com 4 processos tempo gasto = 638 milis
willian@willian-dihanster:~/Downloads/PCD/Lab 05$ mpirun -np 2 lab
Processo 1 tem bloco tam = 50000000
Processo 0 tem bloco tam = 50000000
Processo 1 tem soma_local = 0.287682
Processo 0 tem soma_local = 0.405465
Ln(2.0) = 0.693147180559895
Com 2 processos tempo gasto = 972 milis
willian@willian-dihanster:~/Downloads/PCD/Lab 05$ mpirun -np 1 lab
Processo 0 tem bloco tam = 100000000
Processo 0 tem soma_local = 0.693147
Ln(2.0) = 0.693147180559876
Com 1 processos tempo gasto = 1937 milis
willian@willian-dihanster:~/Downloads/PCD/Lab 05$ ^C
willian@willian-dihanster:~/Downloads/PCD/Lab 05$ mpirun -np 6 lab
Processo 1 tem bloco tam = 16666666
Processo 2 tem bloco tam = 16666666
Processo 3 tem bloco tam = 16666666
Processo 0 tem bloco tam = 16666666
Processo 5 tem bloco tam = 16666670
Processo 4 tem bloco tam = 16666666
Processo 1 tem soma_local = 0.133531
Processo 0 tem soma_local = 0.154151
Processo 3 tem soma_local = 0.105361
Processo 2 tem soma_local = 0.117783
Processo 5 tem soma_local = 0.087011
Processo 4 tem soma_local = 0.095310
Ln(2.0) = 0.693147171469000
Com 6 processos tempo gasto = 597 milis
willian@willian-dihanster:~/Downloads/PCD/Lab 05$

```

Figura 4: Demonstração do funcionamento do código implementado para o Exercício 1.

Exercício 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.

Resultados do Exercício 2

Neste segundo exercício, por sua vez, foram obtidos os seguintes resultados:

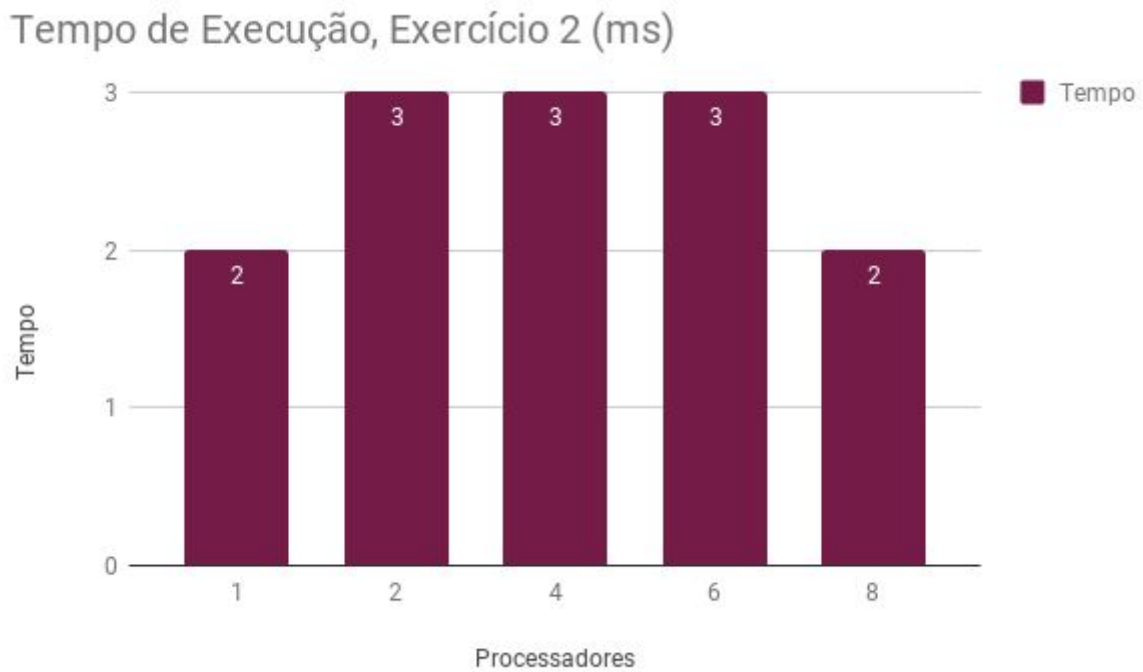


Figura 5: Resultados de tempo de execução obtidos no Exercício 2.

Speedup, Exercício 2

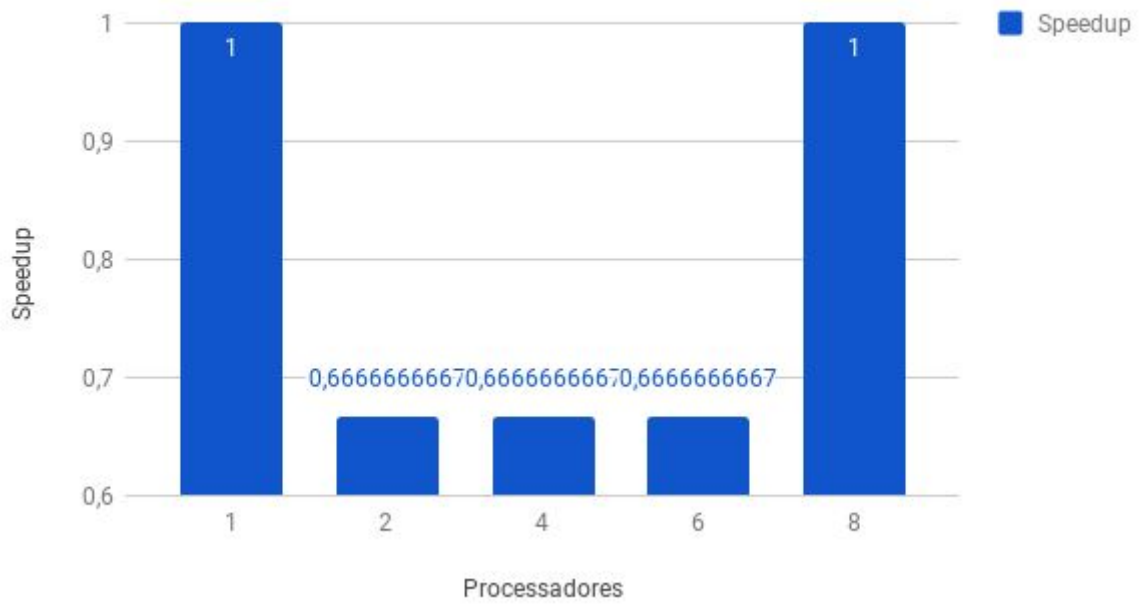


Figura 6: Resultados de Speedup de execução obtidos no Exercício 2.

Eficiência, Exercício 2

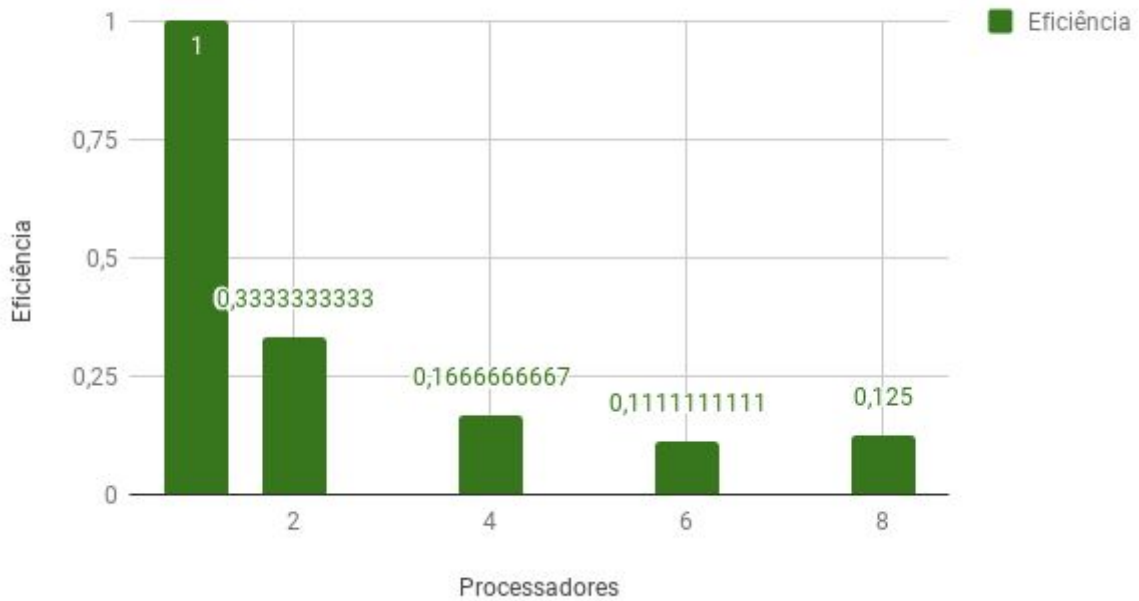


Figura 7: Resultados de eficiência de execução obtidos no Exercício 2.