

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**IDCA3703 - PROGRAMAÇÃO PARALELA**

**TAREFA 14 - LATÊNCIA DE COMUNICAÇÃO USANDO MPI**  
**RELATÓRIO DE EXECUÇÃO**

ERNANE FERREIRA ROCHA JUNIOR

NATAL/RN, 14 DE MAIO DE 2025

## SUMÁRIO

1. INTRODUÇÃO .....	3
2. METODOLOGIA .....	4
3. RESULTADOS .....	5
4. CONCLUSÃO .....	7
5. ANEXOS .....	8

## 1. INTRODUÇÃO

A medição de desempenho na comunicação entre processos é fundamental para compreender as limitações e capacidades de aplicações paralelas que utilizam o modelo *MPI* (*Message Passing Interface*). Em especial, a latência de comunicação e a largura de banda disponível são fatores críticos que influenciam diretamente na eficiência de algoritmos distribuídos. Nesta atividade, foi implementado um programa simples baseado no padrão *ping-pong*, onde dois processos trocam mensagens consecutivas, com o objetivo de analisar o tempo médio de comunicação em função do tamanho da mensagem. Esse tipo de experimento é frequentemente utilizado para caracterizar o comportamento da rede e do sistema de comunicação subjacente, permitindo identificar os regimes onde a latência fixa se sobressai e onde a largura de banda se torna o fator limitante.

A proposta se insere em um contexto educacional e experimental, utilizando a biblioteca *MPI* e recursos de um ambiente de computação de alto desempenho da *UFRN* (*NPAD*). Por meio da variação do tamanho das mensagens, desde 8 *bytes* até 1 *megabyte*, foi possível medir e registrar o tempo médio de ida e volta entre os dois processos. Os dados obtidos foram, então, utilizados para a construção de um gráfico representativo do comportamento da comunicação, oferecendo uma base para discussão e interpretação dos resultados.

## 2. METODOLOGIA

Para a realização do experimento, foi desenvolvido um programa em *C* utilizando a biblioteca *MPI*, com o objetivo de medir o tempo médio de comunicação entre dois processos. O código segue o padrão clássico de comunicação ponto a ponto conhecido como *ping-pong*, no qual o processo de *rank* 0 envia uma mensagem ao processo de *rank* 1, que por sua vez imediatamente responde com a mesma mensagem. Esse ciclo de envio e resposta é repetido múltiplas vezes para garantir maior precisão nas medições, minimizando o impacto de flutuações ocasionais do sistema.

A medição do tempo foi realizada com a função *MPI\_Wtime*, que fornece uma resolução adequada para capturar pequenas variações de tempo em microssegundos. Antes de iniciar cada experimento, os processos foram sincronizados utilizando *MPI\_Barrier*, garantindo que ambos estivessem prontos para iniciar as trocas de mensagem simultaneamente. O número de iterações definido foi de 10.000, e ao final de cada série de trocas, o tempo total foi dividido pelo número de iterações para calcular o tempo médio de ida e volta de uma mensagem.

O experimento foi repetido para diferentes tamanhos de mensagem: 8 bytes, 64 bytes, 512 bytes, 4096 bytes, 32.768 bytes, 262.144 bytes e 1.048.576 bytes (1 MB), representando um espectro que cobre tanto o regime de latência quanto o regime de largura de banda. O *buffer* de envio foi alocado dinamicamente conforme o tamanho da mensagem, e preenchido com um caractere padrão ('a'), sem impacto no conteúdo da comunicação.

O código-fonte foi compilado com otimização de nível 3 (-O3) utilizando o compilador *mpicc*, e executado em um ambiente de *HPC* gerenciado por *SLURM* no cluster *NPAD* da *UFRN*, com dois processos *MPI* alocados na mesma partição. O *script* de submissão garantiu a execução limpa e controlada, com redirecionamento de saída e uso do transportador *vader* para comunicação intra-nó, evitando interferências de drivers de rede externos.

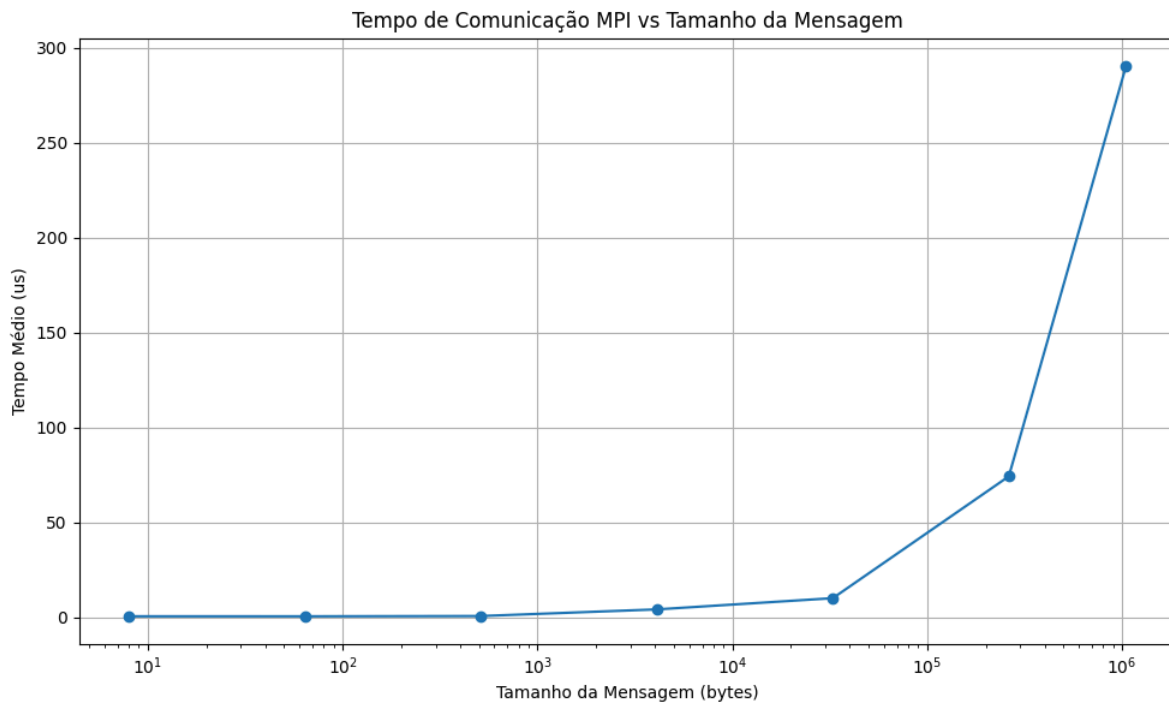
A saída do programa foi salva e posteriormente processada com a biblioteca *pandas* em *Python*, permitindo gerar uma tabela organizada com os tempos médios para cada tamanho de mensagem, além de um gráfico com o comportamento temporal da comunicação, conforme ilustrado na seção de resultados.

### 3. RESULTADOS

A execução do experimento produziu dados consistentes que permitem observar com clareza a influência do tamanho da mensagem no tempo de comunicação entre dois processos *MPI*. A tabela abaixo resume os tempos médios obtidos (em microssegundos) para cada tamanho de mensagem testado:

Tamanho da Mensagem (bytes)	Tempo Médio ( $\mu$ s)
8	0,50
64	0,49
512	0,63
4.096	4,13
32.768	10,04
262.144	74,20
1.048.576	290,47

Os resultados foram visualizados por meio de um gráfico que relaciona o tempo médio de comunicação ao tamanho da mensagem em escala logarítmica no eixo X. Esse tipo de visualização facilita a identificação de diferentes regimes de comportamento:



**Figure 1:** Tempo de Comunicação MPI vs Tamanho da Mensagem

No gráfico, observa-se que, para mensagens pequenas (até algumas centenas de *bytes*), o tempo de comunicação permanece praticamente constante, em torno de 0,5 microssegundos. Esse comportamento é típico de um regime dominado pela latência, em que o tempo de ida e volta da mensagem não depende significativamente de seu tamanho, mas sim do tempo fixo envolvido na inicialização e gerenciamento da comunicação.

À medida que o tamanho da mensagem aumenta, o tempo médio de comunicação cresce de forma mais acentuada. A partir de aproximadamente 4 *KB*, percebe-se uma transição clara para um regime onde o tempo passa a ser proporcional ao volume de dados transferido, indicando que a largura de banda do canal de comunicação se torna o fator predominante. Esse comportamento é consistente com o esperado em sistemas *MPI*, onde a sobrecarga fixa é dominante para mensagens pequenas e o custo de transferência por *byte* se torna mais relevante à medida que o volume cresce.

Portanto, os resultados obtidos não apenas confirmam os comportamentos teóricos esperados em ambientes de comunicação paralela, como também evidenciam a efetividade da metodologia empregada para capturar essas características.

## 4. CONCLUSÃO

Os experimentos realizados permitiram caracterizar de forma clara e objetiva o comportamento da comunicação entre dois processos *MPI* em função do tamanho da mensagem. Através da implementação de um programa do tipo *ping-pong* e da medição do tempo médio de ida e volta de mensagens com diferentes tamanhos, foi possível identificar dois regimes distintos: um inicialmente dominado pela latência, com tempos quase constantes para mensagens pequenas, e outro dominado pela largura de banda, onde o tempo cresce proporcionalmente ao tamanho dos dados transmitidos.

A metodologia adotada, combinando múltiplas iterações para suavizar ruídos estatísticos, sincronização com *MPI\_Barrier* e uso de *MPI\_Wtime* para cronometragem precisa, mostrou-se eficaz para capturar as variações de desempenho. A utilização de um ambiente controlado em *cluster*, com ferramentas adequadas de compilação e execução paralela, garantiu a confiabilidade dos dados obtidos.

Os resultados reforçam conceitos fundamentais da programação paralela com *MPI*, como a importância de minimizar a troca de mensagens pequenas em operações intensivas e o impacto que a largura de banda pode ter em comunicações de grande volume. Além disso, o gráfico produzido serviu como um recurso visual valioso para interpretar o comportamento da comunicação ao longo do espectro de tamanhos testados.

Dessa forma, a atividade cumpriu plenamente seus objetivos pedagógicos e técnicos, oferecendo uma visão prática sobre os fatores que influenciam o desempenho em sistemas distribuídos baseados em troca de mensagens.

## 5. ANEXOS

- Repositório no Github com o programa desenvolvido: <https://github.com/ErnaneJ/parallel-programming-dca3703>



