MPI

O que é?

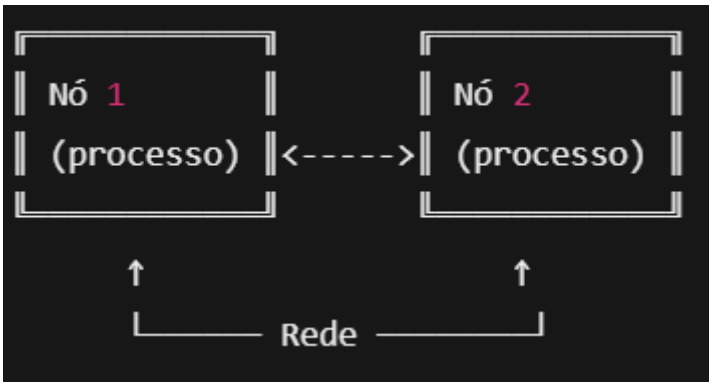
O MPI (Message Passing Interface) é uma especificação de biblioteca de comunicação para programação paralela em sistemas distribuídos. Ele define um conjunto de rotinas e bibliotecas que permitem que os processos em um cluster de computadores se comuniquem entre si de forma eficiente e confiável.

Qual é a Sua Funcionalidade?

O MPI permite que os processos em um cluster de computadores troquem mensagens entre si de forma assíncrona. Cada processo em um cluster é identificado por um número único, chamado de rank. Os processos podem enviar e receber mensagens para outros processos usando rotinas específicas do MPI, como MPI_Send e MPI_Recv. Essas mensagens podem conter dados de qualquer tipo, desde inteiros e floats até estruturas de dados mais complexas.

Forma assíncrona: significa que uma operação **não bloqueia** o restante do programa enquanto ela é executada. Ou seja: o sistema **continua fazendo outras coisas** enquanto essa operação acontece "em segundo plano".

Cluster: Um **cluster** (em computação) é um **conjunto de computadores independentes**, conectados entre si (geralmente por uma rede), que **trabalham juntos como se fossem um único sistema**. A ideia é **somar o poder de processamento** de várias máquinas para resolver problemas maiores ou mais rápido do que uma máquina sozinha conseguiria.



Cada computador no cluster é chamado de **nó** (node). Um dos nós geralmente atua como **mestre** (coordenador) e os outros como **trabalhadores** (workers). Eles se comunicam entre si para dividir tarefas e combinar os resultados. **MPI** (Message Passing Interface) é justamente a **tecnologia** que permite que esses nós **conversem entre si** — passando mensagens de dados.O MPI faz a comunicação entre **processos** que estão rodando em **diferentes máquinas** do cluster ou até no mesmo nó. Sem o MPI (ou tecnologias similares),

seria quase impossível coordenar o trabalho entre múltiplos computadores de forma eficiente

Comunicação Assíncrona em MPI: comunicação assíncrona é quando um processo envia ou recebe uma mensagem sem ficar bloqueado esperando que a operação termine.

Exemplo: Imagine 2 nós, **Nó A** e **Nó B**:

- Nó A **quer mandar** uma mensagem para Nó B.
- Em comunicação **síncrona**, o Nó A **fica parado** esperando o B receber para depois continuar.
- Em comunicação **assíncrona**, o Nó A **manda a mensagem e segue o seu código**, sem esperar confirmação.

Ou seja:

- **Síncrono = "manda e espera"**
- **Assíncrono = "manda e continua"**

Em MPI, isso se traduz em funções específicas:

Comunicação	Função MPI típica	Explicação
Síncrona	`MPI_Send`, `MPI_Recv`	Espera o envio/recebimento acontecer.
Assíncrona	`MPI_Isend`, `MPI_Irecv` + `MPI_Wait`	Inicia o envio/recebimento e continua a execução.

Processo: Processos são partes de um programa que estão executando na CPU, primeiro eles inicializam, vão para o estado de prontos na fila da memória ram, depois são escolhidos, escalonados pelo algoritmo de escalonamento (escalonador) para terem um tempo de execução na CPU, ou

seja, nesse estado eles estão executando. Ciclo de vida do processo (New, Ready, Running, Waiting/Blocked, Terminated)

Rank: Todo o processo tem uma identificação única atribuída pelo sistema quando o processo é inicializado. Essa identificação é contínua representada por um número inteiro, começando de zero até N-1, onde N é o número de processos. Cada processo tem um rank, ele é utilizado para enviar e receber mensagens.

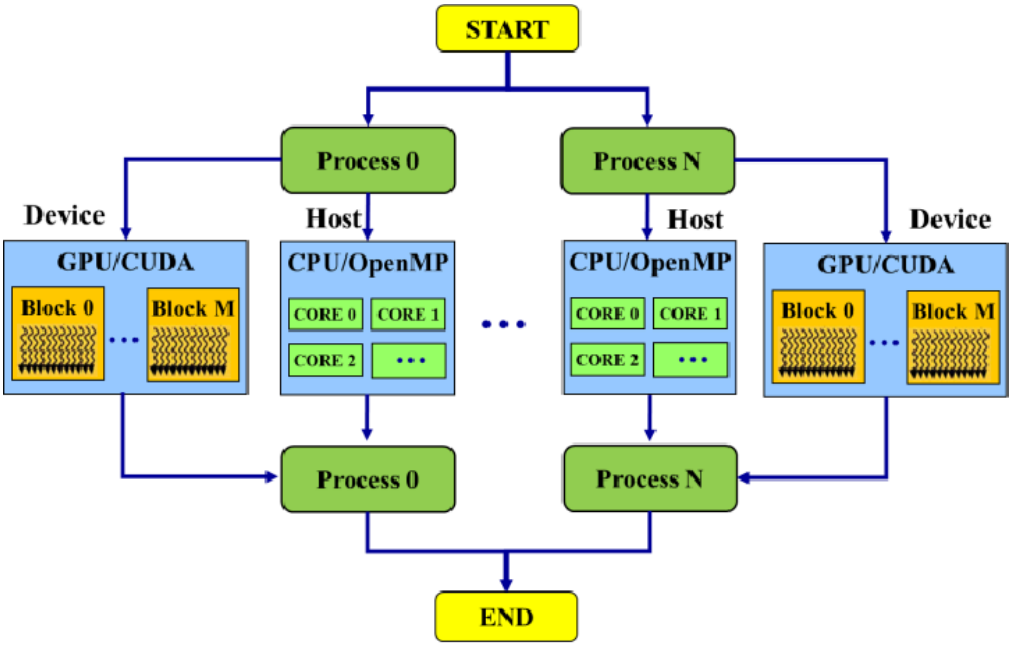
Grupos: Grupo é um conjunto ordenado de N processos. Todo e qualquer grupo é associado a um comunicador muitas vezes já predefinido como "MPI_COMM_WORLD".

Comunicador: O comunicador é um objeto local que representa o domínio (contexto) de uma comunicação (conjunto de processos que podem ser contactados).

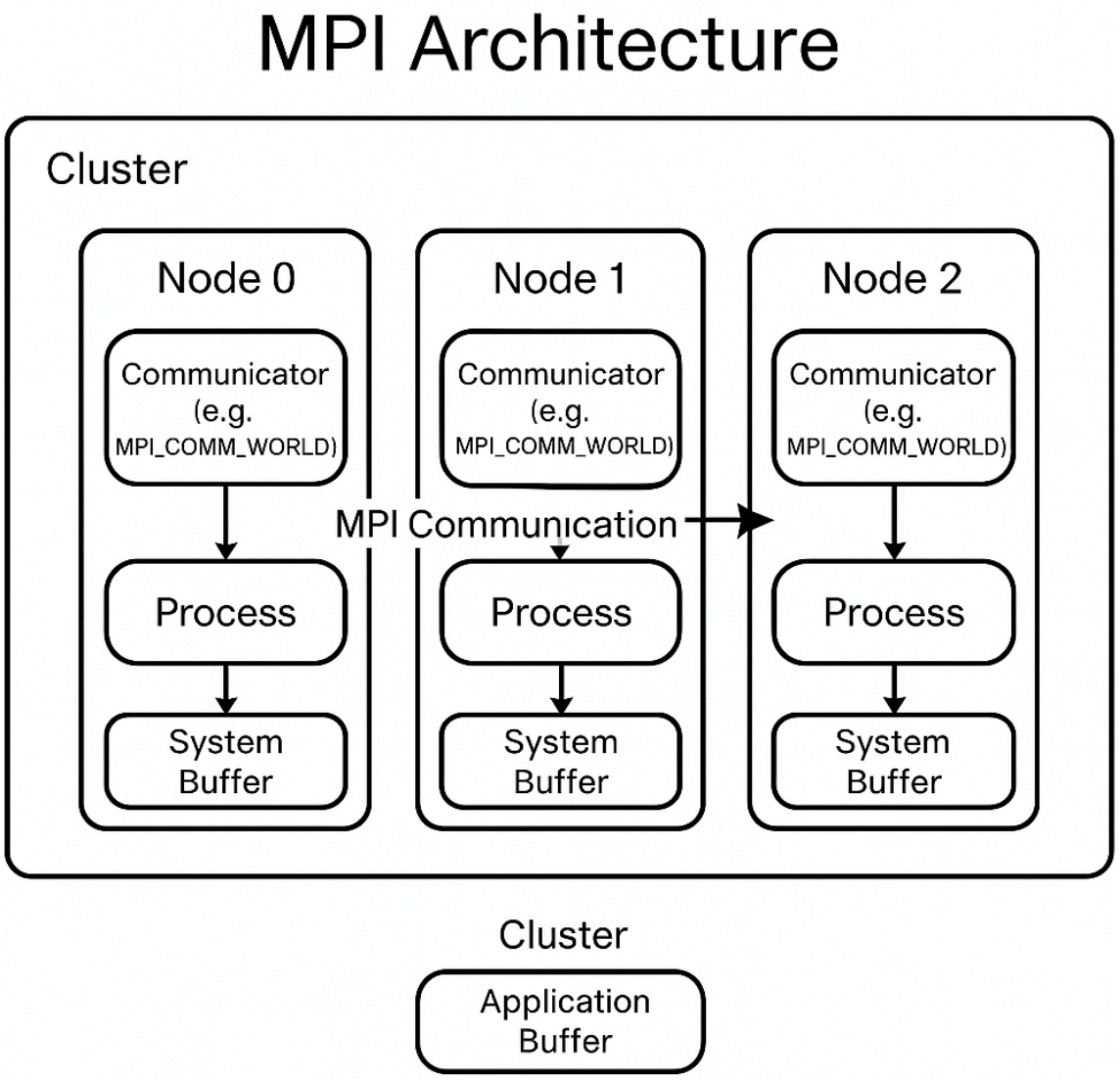
O MPI_COMM_WORLD é o comunicador predefinido que inclui todos os processos definidos pelo usuário numa aplicação MPI

Application Buffer: É o endereço de memória, gerenciado pela aplicação, que armazena um dado que o processo necessita enviar ou receber.

System Buffer: É um endereço de memória reservado pelo sistema para armazenar mensagens.



Exemplo de imagem 1



Exemplo de imagem 2

Tipos de Cluster

Tipo	Descrição rápida
High Performance (HPC)	**Usado para cálculos científicos, simulações pesadas**
Alta Disponibilidade (HA)	**Para serviços que não podem parar (tipo bancos)**
Load Balancing	**Para distribuir cargas (tipo servidores web gigantes)**

Principais vantagens e desvantagens?

Aspecto	Vantagens	Desvantagens
Desempenho	Altíssimo desempenho em clusters — comunicação eficiente entre nós.	Comunicação ainda tem overhead (custos) e pode gerar latência em redes lentas.
Portabilidade	Programas MPI podem ser executados em diversos tipos de hardware (supercomputadores, clusters).	Exige que todos os nós tenham o ambiente MPI instalado e configurado corretamente.
Escalabilidade	MPI escala muito bem para milhares de processos. Ideal para problemas de grande escala.	Programar para alta escalabilidade pode aumentar bastante a complexidade do código.
Controle	Dá ao programador controle fino sobre comunicação, sincronização e dados.	Mais controle = Mais responsabilidade. Erros como deadlocks, perda de mensagens são mais comuns.
Padrão amplamente aceito	MPI é o padrão dominante para computação paralela distribuída.	Pode ser considerado **baixo nível** para aplicações simples: exige mais código para tarefas básicas.

Comunicação eficiente	Comunicação ponto-a-ponto e coletiva bem otimizadas (broadcast, scatter, gather, etc.).	Comunicação explícita pode ser mais difícil de gerenciar do que modelos de memória compartilhada.
Suporte a comunicação assíncrona	Permite sobreposição de comunicação e computação (manda dados e já processa).	Programar comunicação assíncrona é mais **difícil** e exige controle preciso da lógica
Ambiente heterogêneo	Pode rodar em clusters com máquinas diferentes.	Se as máquinas forem muito diferentes, podem ocorrer problemas de desempenho ou compatibilidade.

Fontes:

- O que é: MPI (Message Passing Interface) - SÓ ESCOLA
- https://www.lncc.br/~borges/ist/SO2/trabalhos/Introducao%20ao%20MPI.pdf