

The background features several decorative arcs in white and red, arranged in a circular pattern around the central text. There are four white arcs and four red arcs, each of varying thickness and curvature, creating a dynamic, abstract frame.

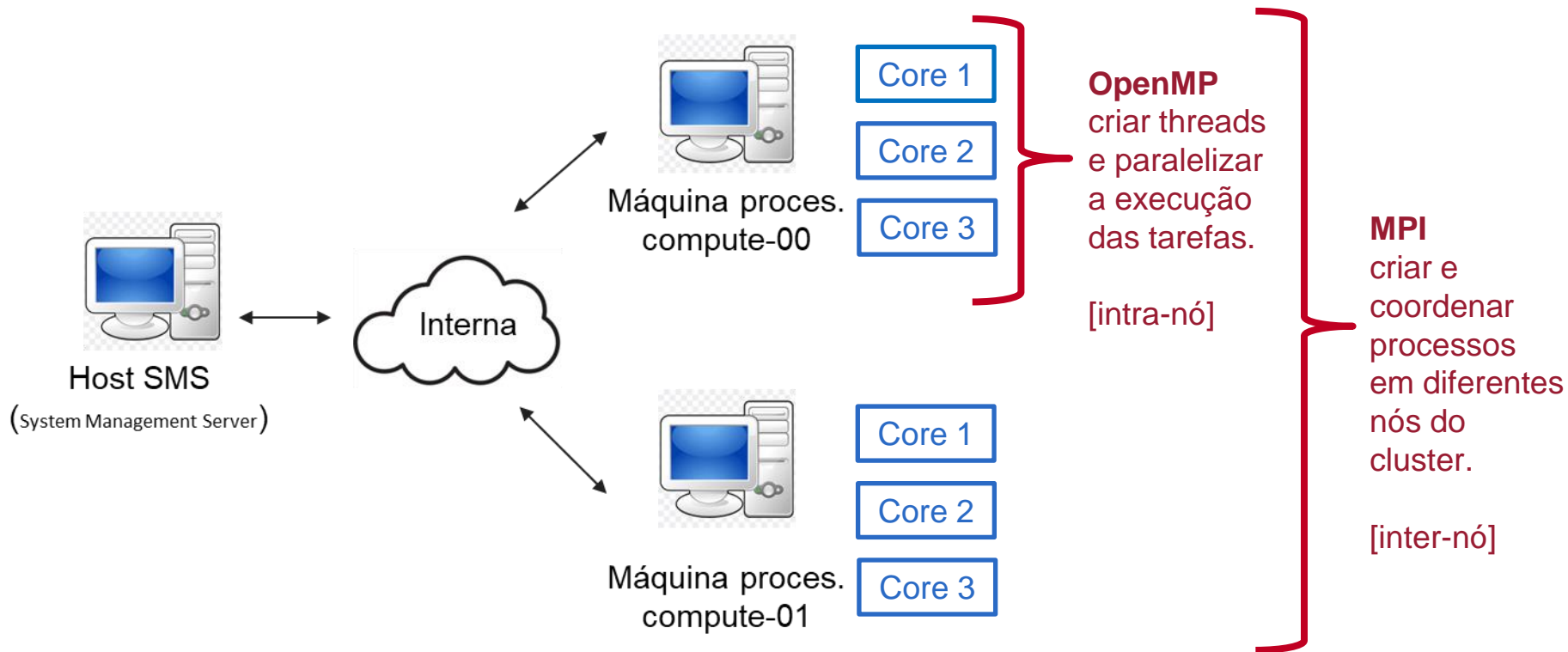
Insper **Supercomputação**



Aula – 19

- OpenMP + MPI

Combinando OpenMP com MPI



Exemplo: multiplicação de elementos

- **Tarefa:** calcular o quadrado de cada elemento em um array bidimensional
- **MPI:** Divide o array entre diferentes processos
- **OpenMP:** Paraleliza o cálculo dentro de cada processo

Exemplo: código

```
#include <iostream>
#include <mpi.h>
#include <omp.h>

int main(int argc, char *argv[]) {
    MPI_Init(&argc, &argv);

    int rank, size;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    const int N = 10; // Dimensões do array bidimensional
    int data[N][N];

    // Inicialização do array pelo processo 0
    if (rank == 0) {
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                data[i][j] = i + j;
            }
        }
    }
}
```

```
// Dividir o array entre os processos
int chunk_size = N / size;
int local_data[chunk_size][N];
MPI_Scatter(data, chunk_size * N, MPI_INT, local_data, chunk_size * N, MPI_INT, 0, MPI_COMM_WORLD);

// Paralelização com OpenMP
#pragma omp parallel for collapse(2)
for (int i = 0; i < chunk_size; i++) {
    for (int j = 0; j < N; j++) {
        local_data[i][j] *= local_data[i][j]; // Calcula o quadrado do elemento
    }
}

// Reunir os resultados no processo 0
MPI_Gather(local_data, chunk_size * N, MPI_INT, data, chunk_size * N, MPI_INT, 0, MPI_COMM_WORLD);

// Processo 0 imprime os resultados
if (rank == 0) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            std::cout << data[i][j] << " ";
        }
        std::cout << std::endl;
    }
}

MPI_Finalize();
return 0;
```



OpenMP + MPI

Paralelização máxima

- Faça os 4 exercícios do roteiro “MPI + OpenMP”
- Implemente-os e execute-os no Cluster!
- Entregue os códigos, “.slurms” e prints das saídas