

## Aula 05 MPI com OpenMP e exercícios de fixação

Para que o OpenMP funcione corretamente no contexto do MPI, este último precisa ser thread safe, isto é, as rotinas MPI devem funcionar corretamente durante execução simultânea de várias threads. Isto implica em dar respostas corretas mesmo com o compartilhamento de dados e de outros recursos durante as execuções das threads.

Para que isso seja possível, o MPI utiliza um conjunto de funções, sendo que a principal é a `MPI_Init_thread()`.

```
int MPI_Init_thread(int *argc, char ***argv, int required, int *provided)
```

A função `MPI_Init_thread` é análoga à `MPI_Init`, porém, também requisita um nível de suporte a execução de múltiplas threads no argumento *required* que pode ser:

`MPI_THREAD_SINGLE`: somente uma thread executará, o que é equivalente a `MPI_Init()`;

`MPI_THREAD_FUNNELED`: se o processo é multithreaded, então a thread que executou `MPI_Init_thread` poderá fazer chamadas MPI.

`MPI_THREAD_SERIALIZED`: se o processo é multithreaded, então somente uma thread fará chamadas à biblioteca MPI por vez.

`MPI_THREAD_MULTIPLE`: se o processo é multithreaded, então múltiplas threads poderão fazer chamadas ao MPI sem restrições.

O suporte dado às múltiplas threads é determinado no argumento de retorno *provided* (C++ é diferente, verificar). Para o Open MPI este valor de retorno depende de como a biblioteca do MPI foi configurada na instalação. Não há garantia *provided* será maior ou igual ao solicitado em *required*.

Todos os programas MPI devem executar `MPI_Init` ou `MPI_Init_thread` antes de executar funções MPI, com poucas exceções.

A passagem de mensagens dentro de regiões paralelas do OMP estão exemplificadas nos códigos usados como exemplo. Os códigos também exemplificam a compilação e a execução por meio do `mpirun`.

Um aspecto imperativo neste modelo de programação híbrido de passagem de mensagem e memória compartilhada é o projeto da aplicação a ser desenvolvida com o PCAM. Sem ele feito, a complexidade do desenvolvimento cresce muito, muitas vezes impossibilitando o desenvolvimento de soluções corretas.

Verificar códigos exemplo disponibilizados no edisciplinas.

Desenvolver os códigos solicitados como exercícios de MPI e OpenMP: TokenRing de processos MPI com threads OMP e Multiplicação de Matrizes com MPI e OpenMP.

Bibliografia:

Barlas, G. (2014). *Multicore and GPU Programming: An integrated approach*. Elsevier. Capítulo 5, Seções 5.16 e 5.21.2.

[https://www.open-mpi.org/doc/v3.0/man3/MPI\\_Init\\_thread.3.php](https://www.open-mpi.org/doc/v3.0/man3/MPI_Init_thread.3.php)