

Exercício de Programação Paralela: Token Ring de Processos MPI/OMP

Complexidade: Baixa
Categoria: Pas. de mens.

Linguagem: C
Mod.de Prog.: MPI com OpenMP

Autor: Paulo Sérgio Lopes de Souza

Este material pode ser utilizado e modificado desde que os direitos autorais sejam explicitamente mencionados e referenciados. Utilizar considerando a licença GPLv2 (GNU General Public License version 2) ou posterior.

Última revisão em Junho de 2020 por Paulo Sergio Lopes de Souza.

Faça um programa paralelo em C com MPI e OpenMP que implemente um Token Ring de processos MPI, estes dispostos logicamente na forma de uma fila circular. O primeiro processo (de rank zero), deverá gerar um token = 0 (zero, um valor inteiro), repassar este valor para o próximo processo (rank um) e aguarda o retorno do token vindo do último processo (rank n-1). Após receber a sua mensagem do processo que o precede na comunicação, cada processo MPI deve gerar NUMTHREADS threads para incrementarem o token recebido. Detalhando com outras palavras, após um processo (incluindo o zero) receber o token, com um determinado valor, as suas threads irão incrementá-lo em NUMTHREADS unidades (cada thread executa um incremento sobre a variável token que é global a elas). Todos os processos MPI diferentes de zero enviarão o novo token para o próximo processo na fila circular. O processo MPI zero imprimirá este valor do novo token e então a aplicação finaliza. Com quatro processos MPI ($-np\ 4$) e NUMTHREADS = 4, a saída da aplicação tokenring deve ser 16 (token inicia em zero).

A macro NUMTHREADS deve ser definida no código (`#define`).

O número de processos MPI deve ser determinado pelo argumento `-np` do `mpirun`.