

SuperComputação

Aula 20 – Mensagens assíncronas

2018 – Engenharia

Igor Montagner, Luciano Soares [<igorsm1@insper.edu.br>](mailto:igorsm1@insper.edu.br)

Objetivos

- Identificar quando o uso de mensagens assíncronas traz ganhos
- Usar a API de mensagens assíncronas do `boost::mpi`

Troca de mensagens

- Programa dividido em processos na mesma ou em outras máquinas;
- Não existe memória compartilhada, dados são trocados enviando (e recebendo) mensagens;
- Mensagens podem ser síncronas ou não;

Mensagens assíncronas

- Programa declara **intenção** de enviar ou receber dados
- Envio/recebimento de fato ocorre em paralelo **enquanto o resto do programa roda**
- Funções para checar se a comunicação terminou e/ou esperar pelo término

Mensagens assíncronas - Vantagens

- Receber mensagens fora de ordem!
 - Mestre divide o trabalho em $N-1$ processos
 - Recebe os resultados na ordem em que são terminados
- Receber e enviar dados grandes
 - Comunicação ocorre em paralelo, é possível realizar outras tarefas enquanto isto.

Mensagens assíncronas- Desvantagens

- Código mais complexo
- Só se transforma em ganhos de desempenho se for bem usado.

Mensagens assíncronas- API

- `isend`, `irecv` são versões assíncronas que recebem os mesmos parâmetros
- `boost::mpi::request` representa uma mensagem enviada/recebida
 - `wait` espera até que a comunicação seja concluída
 - `test` verifica se a comunicação foi concluída, mas não bloqueia

Mensagens assíncronas- API

- Família de funções `wait_*` espera por vários requests ao mesmo tempo:
 - `wait_all`: todos requests terminaram
 - `wait_any`: exatamente um request terminou
 - `wait_some`: um ou mais terminaram

Atividade prática

- Aula 20 no Github
- Atividade 3

Referências

- Livros:
 - Hager, G. ; Wellein, G. **Introduction to High Performance Computing for Scientists and Engineers**. 1ª Ed. CRC Press, 2010.
- Internet:
 - https://www.boost.org/doc/libs/1_67_0/doc/html/mpi/tutorial.html
 -

Insper

www.insper.edu.br