

Introdução à programação paralela

Gabriel Martins de Miranda – 13/0111350

1. Título do capítulo

An Application: Numerical Integration.

2. Objetivo do capítulo

Utilizar a troca de mensagens do MPI com a aplicação de calcular uma integral definida em um intervalo utilizando a regra de Trapézios. São feitos exemplos seriais, em paralelo com variáveis pré-definidas e em paralelo com leitura de entrada do usuário. Nos trabalhos computacionais propostos, exploramos também a técnica de cálculo através da regra de Simpson.

3. Resumo do capítulo

No capítulo não foram mostradas novas funções dos métodos da API, mas foi apresentada uma aplicação com troca de mensagens. Primeiro foi feito um estudo na resolução do problema da integral definida. Aqui foi apresentada a regra do trapézio. Para paralelizar o programa, cada processo se encarregou de resolver um subintervalo do intervalo de integração, sendo este subintervalo dinâmico e igualmente espaçado, significando que poderia escolher-se qualquer número positivo de processos para resolver a questão. Os cálculos locais foram combinados para gerar a soma final, sendo que todos os processos enviam suas somas para o processo 0, que soma tudo e mostra na tela.

Foram discutidas variáveis globais e locais do mundo paralelo, assim como entrada e saída. No primeiro caso, a importância de separar variáveis usadas para todos os processos (globais) de variáveis usadas por processos em específico (locais). Já no segundo, foi visto que para alguns sistemas é permitido à todos os processos ler da entrada padrão e escrever na saída padrão. Porém em outros, tem-se problemas de corridas. Com isto, é preferível que apenas um processo faça estas operações.

4. Solução dos exercícios

Foi usado o MPICH, implementação de alta performance e portabilidade do MPI.

1. Type in the first version of the parallel trapezoidal rule program. Define $f(x)$ to be a function whose integral you can easily calculate by hand (e.g., $f(x) = x^2$). Compile it and run it with different numbers of processes. What happens if you try to run it with just one process?

Para compilar: `$ mpicc ex1.c -o ex1`

Para executar: `$ mpiexec -n 4 ex1`, onde `-n` representa o número de processos criados, sendo cada um executado por um processador (teoricamente). No exemplo são usados 4 processos.

É possível executar o programa com sucesso para qualquer número positivo de processos. No caso de $n=1$ (apenas o processo principal), ele mesmo calcula a integral no intervalo de integração completo, não recebendo mensagem nenhuma de outros processos (inexistentes).

Para $n > 1$, o processo principal sempre calcula o primeiro sub-intervalo de integração, enquanto os outros processos calculam seus intervalos em ordem crescente. Todos os outros processos enviam através de mensagem o valor da integral calculada para o processo principal, que soma todas as partes incluindo a dele mesmo.

Para valores de n divisíveis por 1024, no programa a quantidade total de trapezóides, o valor total da integral saiu em torno de 0.33333. Já no caso de valores não divisíveis, algumas áreas de trapézios não são levadas em consideração por nenhum processo, logo a área total ficou menor.

2. Modify the parallel trapezoidal rule program so that a, b , and n are read in and distributed by process 0 – use the `Get_data` function. Where should the function be called? Do you need to make any modifications other than including the definition of `Get_data` and a call to `Get_data`?

A função é chamada em todos os processos, após ter-se definido o rank de cada um e quantidade de processos existentes. Dentro da função, o processo principal se encarrega de enviar as mensagens aos outros diferenciando os dados pelas tags. Os outros processos devem receber o que vem da main. Não são necessárias maiores modificações.

5. Trabalhos de programação

1. Modify the parallel trapezoidal rule program so that it has several different functions it can integrate, and the chosen function is passed to the `Trap` subroutine. Have the user select which function is to be integrated by giving him a menu of possible functions.

Mais uma variável é enviada para `Get_data`, `func`. O processo principal pergunta ao usuário qual a função será utilizada. Ele envia a mensagem da opção escolhida para os demais processos, que chamam `Trap` com o novo parâmetro `func`. Dentro de `Trap`, o parâmetro é enviado para a função `f` fazer os devidos cálculos do valor no ponto x .

2. a)

Foi feita a regra de Simpson para cálculo da integral de forma serial. Utilizando o mesmo intervalo de integração e subintervalos que a forma paralela usando regra dos trapézios, obteve-se: para $a=0$, $b=1$, $n=1024$ e $f=x^2$, 0.333333 no caso com regra dos trapézios e 0.333252 para a regra de Simpson, ficando bem próximo. Com $a=0$, $b=100$, $n=34$, 278037.875000 na regra dos trapézios e 331026.625000 para Simpson, havendo certa discrepância.

2. b)

Foi feita a regra de Simpson para cálculo da integral de forma paralela. Utilizando o mesmo intervalo de integração e subintervalos que a forma paralela usando regra dos trapézios, obteve-se: para $a=0$, $b=1$, $n=1024$ e $f=x^2$, 0.333333 no caso com regra dos trapézios e 0.333313 para a regra de Simpson, ficando bem próximo. Com $a=0$, $b=100$, $n=34$ e $f=-x$, -4429.065918 na regra dos trapézios e -4429.065918 para Simpson.

6. Conclusão

Por meio do capítulo, foi possível entender bem as possibilidades do uso da programação paralela em um exemplo prático, o de cálculo de uma integral definida num determinado intervalo usando regra de trapézios. O programa foi dividido em processos, onde cada um resolveu sua parte e enviou por mensagens ao processo principal, que reuniu todas as informações em uma só.

Entendeu-se também a diferenciar variáveis de locais e globais e problemas ocorridos com leitura e saída em programas paralelos.

7. Referências consultadas

<http://alunosonline.uol.com.br/matematica/area-trapezio.html>

Pacheco, P. S., (1997) Parallel Programming with MPI. Morgan Kaufmann.

Obs.: As primeiras linhas nos fontes anexados indicam como compilar.