Desempenho de Programas com MPI

Tópicos:
- Exemplo: Integração Numérica
- Medição de Tempo com MPI
- Modelos de Desempenho
- Interface para Medidas em MPI

Referência: Pacheco,P.S. *Parallel Programming with MPI* Morgan Kaufmann, San Francisco, 1997.

Exemplo: Integração Numérica

- Dada: função não-negativa f(x)
- Objetivo: Calcular a integral de f(x) em [a,b]
- Método: Regra do Trapézio
  - Dividir intervalo [a,b] em n partes, largura h = (b-a)/n
  - Aproximar área de cada intervalo com um trapázio
  - i-ésimo intervalo: [x_{i-1} , x_{i-1} + h] = [x_{i-1} , x_i]
  - Área do i-ésimo intervalo ≈ [$f(x_{i-1}) + f(x_i)$] . h / 2
  - Área total: [$f(x_0)/2 + f(x_n)/2 + f(x_1)+f(x_2)+ ... +f(_{xn-1})$] . h

Ex.: Integração Numérica (cont.)

Implementação Serial:

```
float f(float x);                    → função externa
printf("Enter a, b, and n\n");
scanf("%f %f %d", &a, &b, &n);

h = (b-a)/n;
integral = (f(a) + f(b))/2.0;
x = a;
for (i = 1; i <= n-1; i++) {
   x = x + h;
   integral = integral + f(x);
}
integral = integral*h;
```


Ex.: Integração Numérica (cont.)

Estratégia de Paralelização:
- Dividir os *n* intervalos pelos *P* processadores
- Cada processador calcula a integral em sua parte
- Cada processador envia seu resultado ao proc.0
- Processador 0 calcula área total

Questões a serem resolvidas:
- Fazer entrada de dados (a,b,n) para os processadores
  → Solução trivial: Processador 0 lê dados
- Determinar intervalos de trabalho de cada processador
  → Possível solução:Cada proc.pega intervalos vizinhos

Ex.: Integração Numérica (cont.)

Implementação Paralela:

```
h = (b-a)/n;    local_n = n/p;
local_a = a + my_rank*local_n*h;  local_b = local_a + local_n*h;
integral = Trap(local_a, local_b, local_n, h);
if (my_rank == 0) {
      total = integral;
      for (source = 1; source < p; source++) {
         MPI_Recv(&integral, 1, MPI_FLOAT, source, tag,
            MPI_COMM_WORLD, &status);
         total = total + integral;
      }
} else {
      MPI_Send(&integral, 1, MPI_FLOAT, dest,
         tag, MPI_COMM_WORLD); }
```


Medição de Tempo com MPI

Função para leitura de tempo:

double MPI_Wtime( ) : Retorna tempo "atual", em seg.

Uso Típico:

```
double inicio, final, delta;
...
MPI_Barrier( );
inicio = MPI_Wtime( );
. . .   (código a ser medido)
MPI_Barrier( );
final = MPI_Wtime( );
delta = final - inicio ;
```

Modelos de Desempenho

Definição:
- Modelos matemáticos que representam o tempo de execução de um programa

Utilidade:
- Representação do tempo de execução como função de *n* (tamanho do problema) ou *p* (número de processadores)
- Capacidade de predição, para diferentes valores de *n*, *p*

Em Geral:
- Tempo de um programa serial: $T = T_\sigma(n)$
- Tempo de um programa paralelo: $T = T_\pi(n , p)$

Modelos de Desempenho (cont.)

Speedup de uma execução paralela:

$S(n , p) = T_\sigma(n) / T_\pi(n , p)$

Eficiência de uma execução paralela:

$E(n , p) = S(n , p) / p = T_\sigma(n) / p \, . \, T_\pi(n , p)$

Componentes do Custo de Execução:
- Tempo de Computação: proporc. ao núm.comandos
- Tempo de Comunicação:
  - Tempo de uma mensagem: $\alpha + \beta x$, onde x = tam.da msg
  - Tempo de várias mensagens: depende do hardware do sistema
- Tempo de E/S : iremos ignorar

Modelos de Desempenho (cont.)

Parâmetros do tempo de uma mensagem:

α : latência

β : 1/larg.faixa

Valores Típicos em Sistemas Reais (em μs):

	α	β
Intel Paragon	29	0.052
Intel iPSC/860	65	2.7
IBM SP-2	35	0.23
nCUBE2	170	4.7
TMC-CM5	95	0.89

Modelos de Desempenho (cont.)

Caso de várias mensagens (Ex: broadcast)
- Num sistema do tipo hipercubo: $T = k \, . \, \log_2 p$
- Num sistema com barramento: $T = k \, . \, (p\text{-}1)$

Tempo Total do Ex. de Integração Numérica:
- Num hipercubo: $T_\pi = k_1 \, n / p + k_2 \, . \, \log_2 p + k_3$
- Com barramento: $T_\pi = k_1 \, n / p + k_2 \, (p\text{-}1) + k_3$

Speedup (supondo barramento):

$S(n , p) = (k_4 \, n + k_5) / [k_1 \, n / p + k_2 \, (p\text{-}1) + k_3]$

Interface para Medidas em MPI

Objetivo:
- Obter estatísticas sobre o uso de funções de MPI
- Ex: Número de chamadas a MPI_Send( ) ; Tempo, etc.

Problema:
- Chamadas a MPI_Send podem estar dentro de outras funções de MPI (p.ex. MPI_Bcast)

Solução:
- No padrão MPI: MPI_Send( ) ≡ PMPI_Send( )
- MPI_Send( ): pode ser reescrita pelo usuário, incluindo cálculo de estatísticas e chamada a PMPI_Send( )

Interface para Medidas em MPI (cont.)

Exemplo: acumular o tempo gasto em MPI_Send( )

```
static double send_time = 0.0;
int MPI_Send(void* buffer, int count,
      MPI_Datatype datatype, int dest, int tag, MPI_Comm comm) {
   double start_time, finish_time;
   int return_val;

   start_time = MPI_Wtime();
   return_val = PMPI_Send(buffer, count, datatype, dest, tag, comm);
   finish_time = MPI_Wtime();
   send_time = send_time + finish_time - start_time;
   return return_val;
}
```

Obs: Esta rotina deve ser "linkada" antes da bibliot. de MPI