

# Programação Paralela e Concorrente

Bolsas de Tarefas e outros modelos

2016

- todos os processos executam o mesmo código
- quantidade de trabalho determinada estaticamente:
  - tarefas pré-definidas
  - quantidade de trabalho em cada tarefa é fixa ou bem conhecida
  - poder computacional homogêneo
  - em aplicações embaraçosamente paralelas: *master-slave*

- ambientes heterogêneos (não em nosso caso)
- ambientes com carga externa à aplicação
- **tarefas em quantidades diferentes de trabalho**
  - caso da quadratura adaptativa
  - ou outra visão: tarefas criadas dinamicamente
  - às vezes chamadas de *aplicações irregulares*
- uma solução é manter número de tarefas muito maior que número de processadores e distribuí-las ao longo da execução
  - **bolsa de tarefas**

- ou **mestre/trabalhador**
- ou ainda distribuição **sob demanda**
- trabalhadores repetidamente consomem tarefas de uma *bolsa de tarefas* e enviam resultados para mestre
- apropriado para programas embarçosamente paralelos
  - dependências não permitem alocação livre
- forma rudimentar mas bastante eficiente de balanceamento de carga
  - atende flutuações em carga interna e externa

- difícil de casar um modelo sob demanda com openMP ou MPI (“feitos” para aplicações SPMD regulares)

- processo mestre lê vetor, distribui para todos, lê matriz e entra em loop:

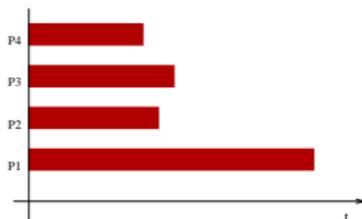
```
do
    recebe pedido
    envia linha
while (tem linha);
```

- demais processos ficam em loop

```
do
    envia pedido
    recebe linha
    if (linha) calcula elemento
while (linha);
```

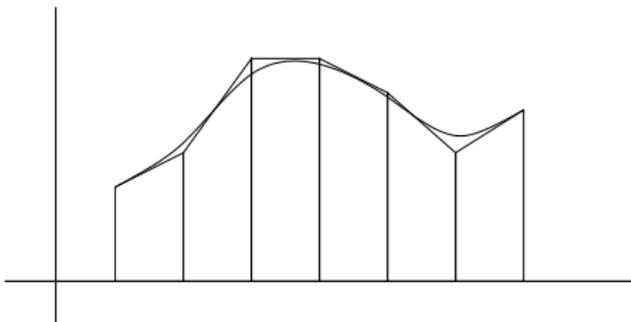
# Utilidade?

- nesse problema a quantidade de trabalho é bem conhecida
  - divisão inicial, estática, evita custos com comunicação
- bolsa de tarefas pode ser útil apenas se:
  - processadores com capacidades diferentes
  - processadores sobrecarregados com outras tarefas



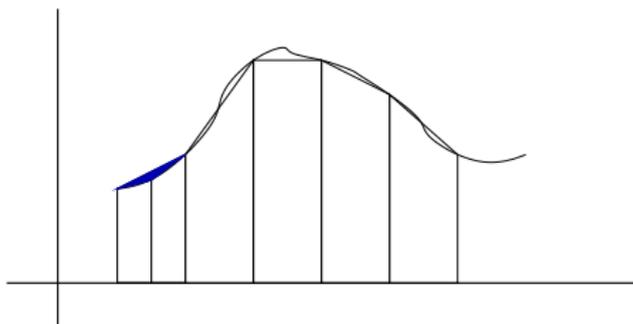
# Exemplo: Quadratura

- cálculo da área abaixo de uma curva: aproximação por trapézios



# Quadratura Adaptativa

- adaptação dinâmica à curva
  - diferença entre área de 1 trapézio e 2 subtrapézios indica se trabalho em certo intervalo deve continuar ou não



- trabalhador pode fazer chamadas recursivas enquanto intervalos não aceitáveis...
  - nesse caso a quantidade de trabalho associada a cada intervalo inicial é diferente
  - pode valer a pena quebrar em muitos intervalos e usar a bolsa de tarefas para distribuí-los dinamicamente
- ou trabalhador pode gerar novas tarefas na bolsa de tarefas

# Implementações possíveis

- mais comum: mestre mantém a bolsa de tarefas
  - mestre pode ou não agir como trabalhador também
    - perda de um processador trabalhador X demoras em respostas
- estruturas de comunicação podem facilitar outras implementações
  - troca de mensagens!
    - canais ou mailboxes
  - espaço de tuplas

# Quadratura – Implementação sob demanda

```
module Manager
  op getTask(result double left, right);
  op putResult(double area);
body Manager
  process manager {
    double a, b;          # interval to integrate
    int numIntervals;    # number of intervals to use
    double width = (b-a)/numIntervals;
    double x = a, totalArea = 0.0;
    int tasksDone = 0;
    while (tasksDone < numIntervals) {
      in getTask(left, right) st x < b ->
        left = x; x += width; right = x;
      [] putResult(area) ->
        totalArea += area;
        tasksDone++;
      ni
    }
    print the result totalArea;
  }
end Manager

double f() { ... }      # function to integrate
double quad(...) { ... } # adaptive quad function

process worker[w = 1 to numWorkers] {
  double left, right, area = 0.0;
  double fleft, fright, lrarea;
  while (true) {
    call getTask(left, right);
    fleft = f(left); fright = f(right);
    lrarea = (fleft + fright) * (right - left) / 2;
    # calculate area recursively as shown in Section 1.5
    area = quad(left, right, fleft, fright, lrarea);
    send putResult(area);
  }
}
```

**Figure 9.2** Adaptive quadrature using manager/workers paradigm.

Copyright © 2000 by Addison Wesley Longman, Inc.

- implementação de quadratura adaptativa:
  - trabalhadores depositam novas tarefas na bolsa
  - mais um tipo de chamada de trabalhadores a mestres