

Paradigmas de programação concorrente

one. 3/3

L.1

- 1) Paralelismo iterativo
- 2) Paralelismo recursivo
- 3) Sistemas produtor/consumidor
- 4) Clientes e servidores
- 5) Iteração entre pares

Paralelismo iterativo

Quando um programa contém diversos processos, cada um com um ou mais laços

Os processos cooperam para resolver um único problema.

cooperação → comunicação → sincronização

Ex.: multiplicação de matrizes

-2

```
double a[n,n], b[n,n], c[n,n];
for [i=0 to n-1] {
  for [j=0 to n-1] {
    c[i,j] = 0.0;
    for [k=0 to n-1] {
      c[i,j] = c[i,j] + a[i,k] *
      b[k,j];
    }
  }
}
```

trivialmente paralelizado

Independência

Definição: duas operações são independentes se o conjunto de variáveis alteradas (e lidas) são disjuntos.

- trocar os 2 primeiros FORs por

CO. = sincronização implícita. Começa a execução em paralelo e aguarda o seu término. E se n é muito grande?

'custo de criação de thread maior que ~~o~~

- de multiplicações. Logo não vale a pena fazer uma multiplicação em cada thread $O(n \cdot n^2)$ "

" problema de o cache carregar bloco que será invalidado por outro " Solução ;

trocar os COs de i com j ^(linhas) e colocar FOR no i.

~~em múltiplos chamados~~

L-4

primitiva `procarr` \cong co executado em background
pode ser interessante criar apenas P `processor`
`procarr worker` [$w = 1$ to P]

```
1 int f = (w - 1) * n / P;
```

```
1 int l = f + n / P - 1;
```

```
for [i = f to l] {
```

```
1 for [j = 0 to n - 1] {
```

```
    c[i, j] = 0 0;
```

```
1     }  $\square$  > último for do anterior
```

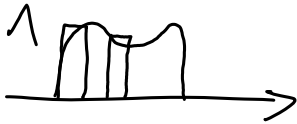
```
1 }
```

-) Paralelismo recursivo

Um programa recursivo pode ser implementado usando concorrência quando

tem múltiplas chamadas recursivas 25
independentes.

Ex: Cálculo de integral



versão iterativa

```
double fleft = f(a), fright, area = 0.0;  
double largura = (b-a) / MAX;  
for [ x = (a + largura) to b by largura ] {  
    fright = f(x);  
    area = area + (fleft + fright) * largura / 2;  
    fleft = fright;  
}
```

método divisão e conquista

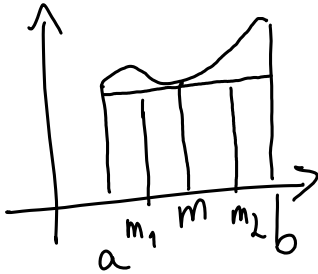
ache o ponto médio m

- ache as áreas $a-m$, $m-b$ e $a-b$

u estimer proximo OK

26

c.c. continue



double quad (double left, right, fleft,
right, harea)

$$\text{double mid} = (\text{left} + \text{right}) / 2.0;$$

$$\text{double fmid} = f(\text{mid})$$

$$\text{double larea} = (\text{fleft} + \text{fmid}) * (\text{mid} - \text{left}) / 2.0;$$

$$\text{double rarea} = (\text{fmid} + \text{fright}) * (\text{right} - \text{mid}) / 2.0;$$

if (|larea + rarea - harea| > ϵ) {
 larea

$$\text{larea} = \text{quad}(\text{left}, \text{mid}, \text{fleft}, \text{fmid},$$

larea);

127

rarea = quad(mid, right, fmid, frigt,
rarea);

}

return (larea + rarea),

}

CO no bloco de cálculo de quadr.

problema: possível excesso de concorrência

redução: parâmetro a main = profundidade.

problema: quando um lado se aprofunda muito mais que outra

outros exemplos: ordenação, n-rambas