Arquiteturas de Computadores

Algoritmos paralelos/padrões de paralelismo

João Luís Ferreira Sobral
jls@...

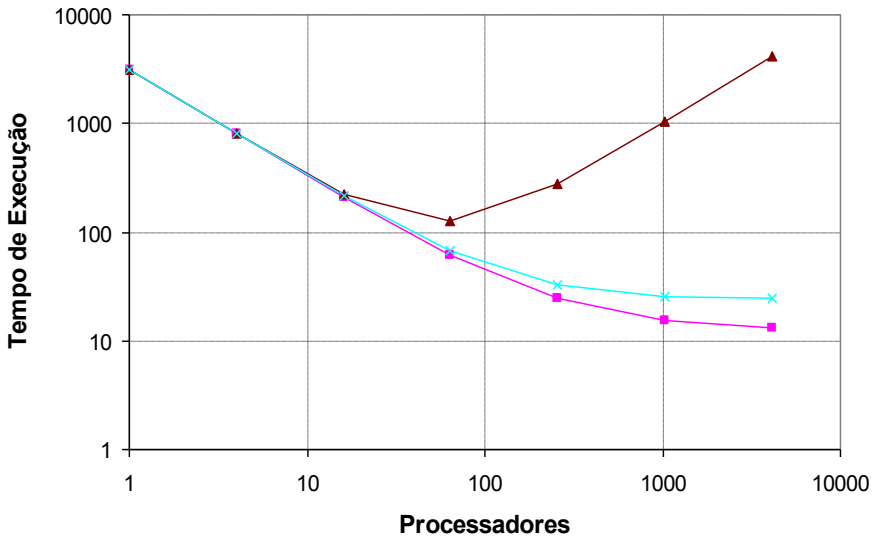
Algoritmos paralelos

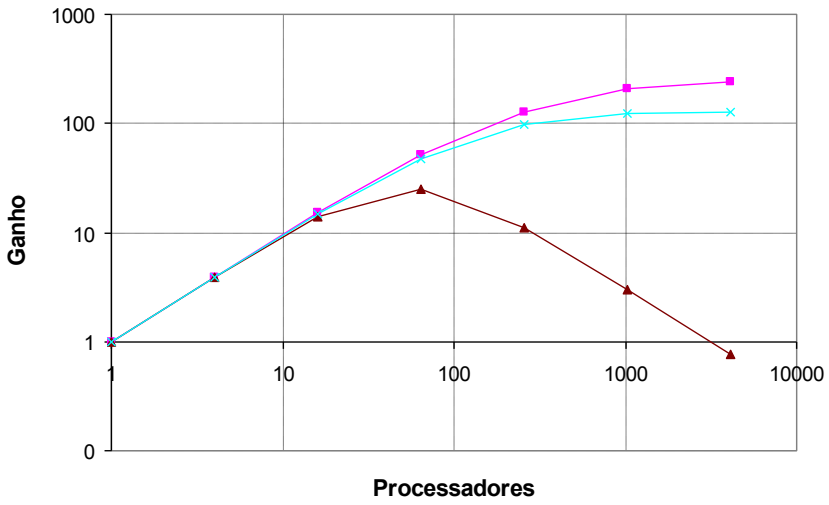
Conteúdos	**5 – Processamento paralelo**
	5.1 - Processadores Multi-Núcleo
Resultados de Aprendizagem	R5.2 – Identificar oportunidades de processamento paralelo R5.3 – Caracterizar as limitações inerentes ao processamento paralelo

Algoritmos paralelos

Os melhores algoritmos para as arquiteturas paralelas (e.g., multi-núcleo) podem ser significativamente diferentes das variantes sequenciais

- **Como comparar dois algoritmos diferentes?**
  - **Complexidade (O(???))**
    - **Usada para comparar algoritmos sequenciais**
  - **Escalabilidade**
    - **Analisar como evolui o tempo de execução do algoritmo quando se aumenta o número de núcleos**
    - **Em alternativa pode ser usado o *ganho* relativamente à *melhor* versão sequencial:**
      - **ganho = Texe versão paralela / Texe versão sequencial**





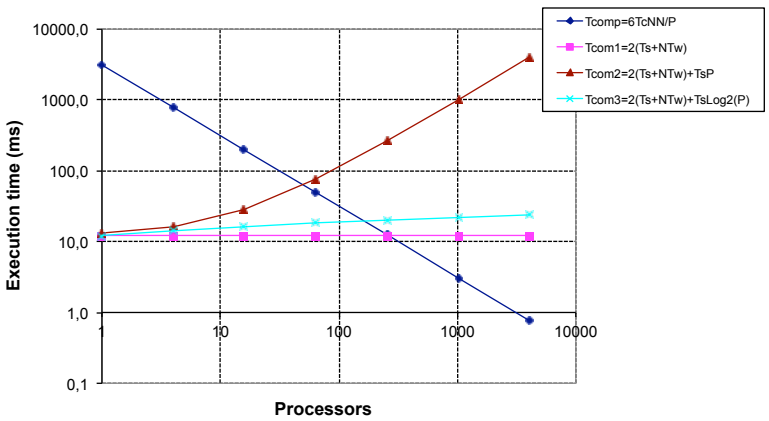
Algoritmos paralelos

Como comparar algoritmos paralelos?

- **Exemplo: Somar N elementos de um vector em *p* processadores**
  - **Versão 1:**
    1. **Cada processador soma N/p elementos: N/p passos**
    2. **Um dos processadores efetua a soma final: p passos**

       Texe = N/p + p
  - **Versão 2:**
    - **Igual à versão 1: N/p passos**
    - **Em log(p) passos:**
      - **Cada processador soma duas soma parciais**

        Texe = N/p + log(p)



Versão 1 para 8 processadores

```
double soma=0;
 double s[8];
 for(int w=0;w<8;w++) s[w]=0;

#pragma omp parallel for shared(s)
for(int i=0; i<1000000;i++) {
     s[omp_get_thread_num()] += sin(i);
}
 for(int w=0;w<8;w++) soma+=s[w];
```

Versão 2 (redução em paralelo)

```
...  // parte inicial igual à versão 1
// for(int w=0;w<8;w++) soma+=s[w];
// log(8) passos
int myID = omp_get_thread_num();

if (myID<4)
   s[myID] += s[myID+4]

if (myID<2)
   s[myID] += s[myID+2]

if (myID<1)
   s[myID] += s[myID+1]
```

Algoritmos paralelos

Alguns factores que limitam a escalabilidade dos algoritmos

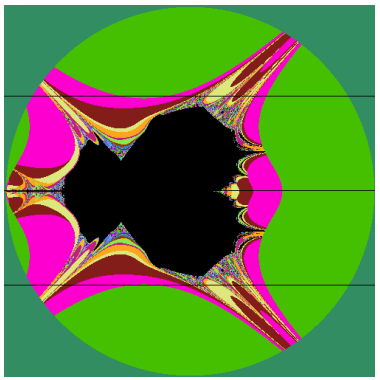
- **Fração sequencial dos algoritmos**
  - **Lei de Amdahl**
- **Largura de banda de acesso à memória insuficiente**
  - **Largura de banda disponível é dividida pelos núcleos em execução**
  - **Problema é aliviado se o algoritmo/codificação for amigável da *cache***
- **Sincronização entre os vários núcleos**
  - **Serializam a execução e introduzem #I adicionais**
- **Sobrecargas do paralelismo**
  - **Em proporção, aumentam quando o número de núcleos aumenta**
- **Trabalho redundante**
  - **Operações realizadas simultaneamente pelos vários processadores que não existem na versão sequencial**

Padrões comuns de paralelismo

Algoritmos embaraçosamente paralelos

- **Tipicamente são constituídos por ciclos com iterações independentes**
- **Podem existir problemas de balanceamento de carga**
- **Exemplo em OpenMP:**

```
#pragma omp parallel for
for(int i=0; i<ImgSize; i++)
   for(int j=0; j<ImgSize; j++)
     Image[i][j] = Mandelbrot(i,j);
```



Padrões comuns de paralelismo

Funções recursivas com grau superior a 1 (split & merge / divide and conquer)

- **As chamadas às subfunções podem ser realizadas em paralelo**
  - **Exemplo:**
    - **Fibonacci; quicksort**

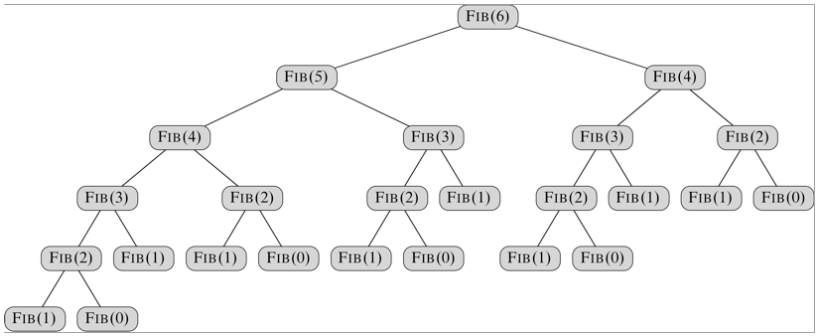
- **Exemplo em OpenMP:**
  - **OpenMP introduziu o construtor *task* para suportar este tipo de paralelismo:**

```
double fib(double n) {
       double i, j;
       if (n<2) return(n);

#pragma omp task shared(i), firstprivate(n)
       i = fib(n-1);
       j = fib(n-2);

#pragma omp taskwait
       return(i+j);
}
int main() {

#pragma omp parallel
#pragma omp single
    double r = fib(47);
 }
```



Padrões comuns de paralelismo

Algoritmos iterativos (heartbeat)

- **Executam, repetidamente, uma operação sobre uma estrutura de dados**
  - **Os valores de uma iteração podem ser executados em paralelo, mas não pode ser iniciada a iteração seguinte sem terminar a anterior**
  - **Exemplo:**
    - **Método de Jacobi**

$$X_{i,j}^{(t+1)} = X_{i,j}^{(t)} + X_{i-1,j}^{(t)} + X_{i,j-1}^{(t)} + X_{i+1,j}^{(t)} + X_{i,j+1}^{(t)}$$

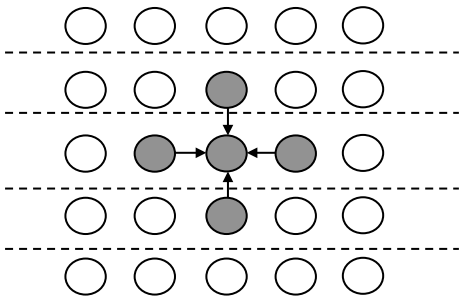
- **Exemplo em OpenMP:**

```
#pragma omp parallel
for(int t=0; t<Niterações; t++) {

    #pragma omp for
    for(int i=1; i<N-1; i++)
        for(int j=1; j<N-1; j++)
            r[i][j]=(x[i-1][j] + x[i+1][j] +
                    x[i][j-1] + x[i][j+1] + x[i][j])/5;

    // #pragma omp barrier  // implícito no « omp for »
    // x = r na próxima iteração

}
```



Programação de arquiteturas multi-núcleo