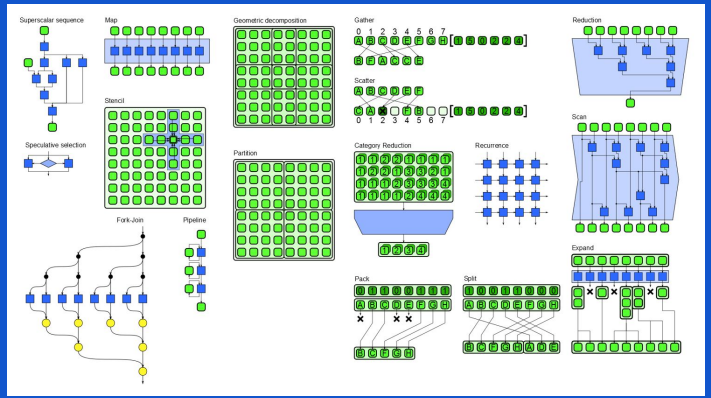
Padrões de Programação Paralela

Luís Fabrício W. Góes (Cabra)

lfwgoes@pucminas.br



Fonte: parallelbook.com

Programação Estruturada

- A “programação estruturada” (sequencial) é composta pelos seguintes padrões
  - Estruturas de Seleção e Repetição
  - Funções, Recursão etc.

- Estas boas práticas de programação eliminam problemas como o GOTO
  - Melhora a manutenabilidade do SW.
  - Aumenta o reuso de SW.
  - Problema: Serialização

Serialização

- Abstração de uma máquina sequencial.
  - Possui ordem temporal.
  - É determinística.
    - Fácil de codificar, depurar e testar.
- Limita o paralelismo sem necessidade.
  - Paralelismo não é determinístico.

Exemplo de Serialização

- Procurar se uma palavra aparece ou não em cada página de um conjunto de páginas Web.

- Como ler este código?

```
for (i = 0; i < numPaginas; i++) {
        encontrada[i] = busca(“palavra”,paginas[i]);
}
```

- Buscar “palavra” na página i, depois em i+1, depois em i+2 etc.

Quebrando a serialização

- A busca na página[i] precisa acontecer antes da busca na página[i+1]?
  - Não. As operações de busca são completamente independentes, ou seja, podem ser realizadas ao mesmo tempo.

- Que tal um for sem serialização?

```
parallel_for (i = 0; i < numPaginas; i++) {
        encontrada[i] = busca(“palavra”,paginas[i]);
 }
```

- **Agora lê-se:** A “palavra” pode ser buscada em todas as páginas “i” ao mesmo tempo, ou seja, em paralelo.

Programação Paralela Estruturada

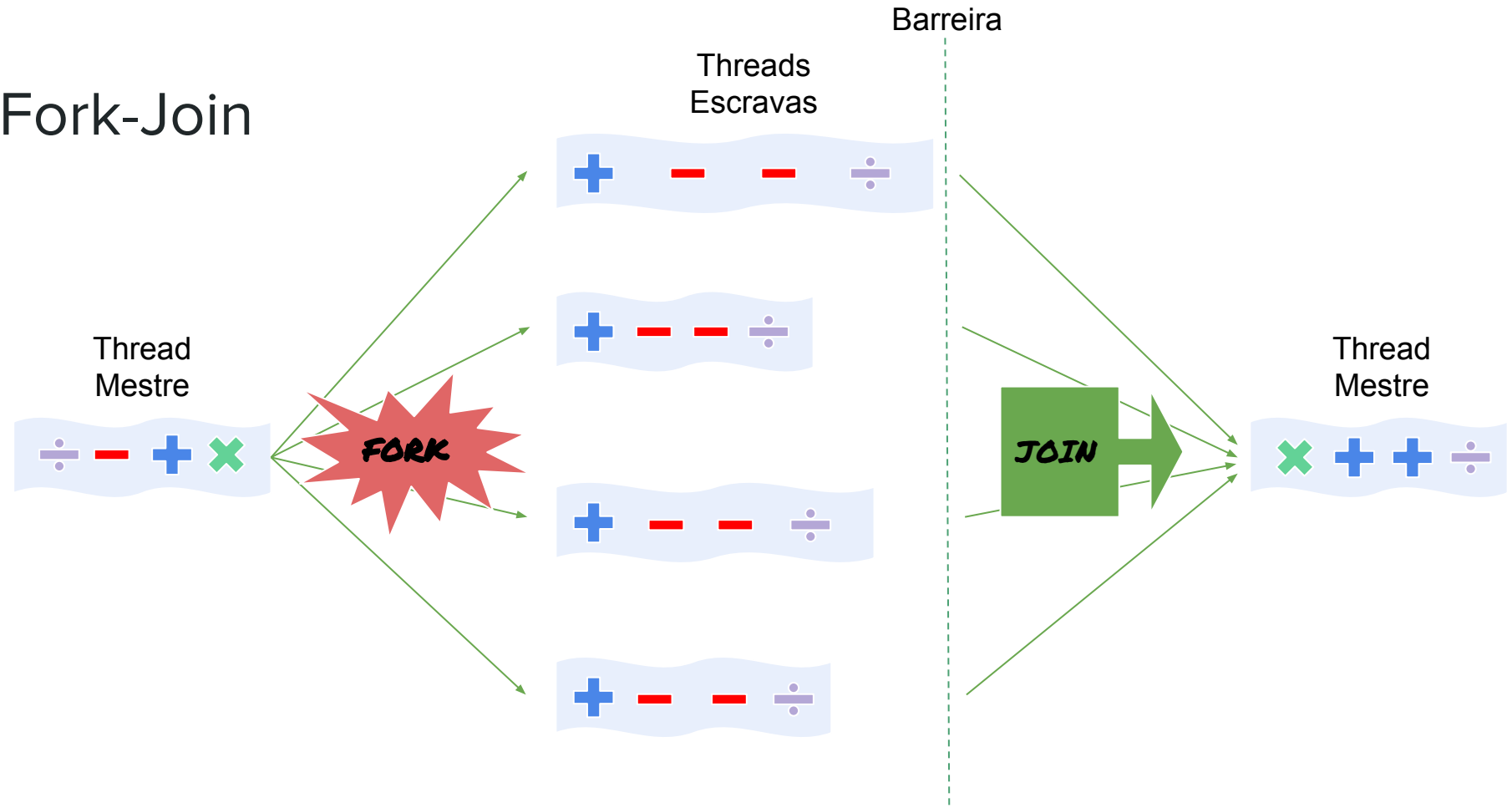
- A “programação paralela estruturada” (paralelo) é composta por padrões paralelos ou esqueletos algoritmicos ou paralelos.
  - Fork-Join, Map, Reduce, Scan, Stencil etc.

- Estes padrões paralelos capturam estruturas comuns a programas paralelos.
  - Elimina o uso explícito de threads.
  - Remove (relaxa) as restrições de serialização dos padrões sequenciais.

Fork-Join

- É o padrão paralelo mais simples.
- Permite que uma mesma seção de código seja executada em paralelo por várias threads trabalhadoras.
- Uma thread mestre dispara (fork) várias threads trabalhadoras e ao final da execução de cada thread, elas sincronizam-se, restando apenas a thread mestre (join).

Fork-Join

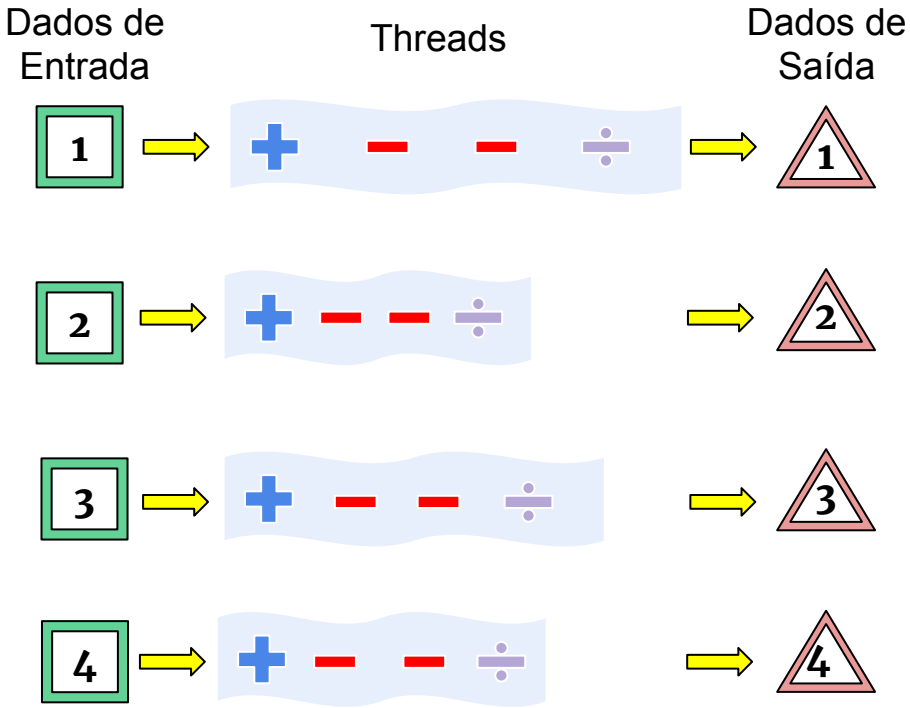

Barreira
Threads
Escravas
Thread
Mestre
FORK
JOIN
Thread
Mestre

Map

- O padrão **Map** replica uma função sobre cada elemento de um conjunto de índices (ou dados diferentes).

- Ele é mais comumente encontraddo na forma de uma estrutura de repetição sem a restrição de ordem de execução das iterações.
  - É necessário que as funções (iterações) sejam independentes.
  - Ex: parallel_for

Map

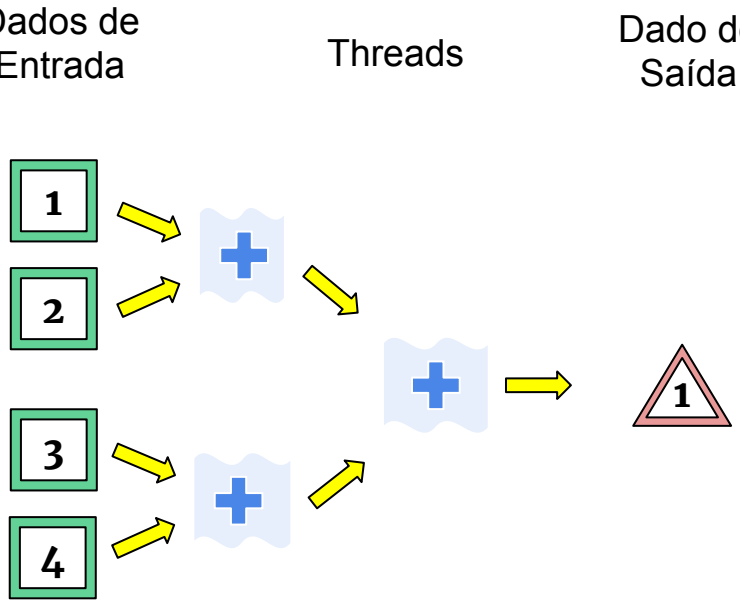


Reduce

- O padrão **Reduce** combina cada elemento de uma coleção, geralmente em pares, utilizando um operador, até reduzi-los a um único elemento.

- Exemplos
  - Realizar um somatório de inteiros.
  - Encontrar o maior valor de um vetor de inteiros.

Reduce

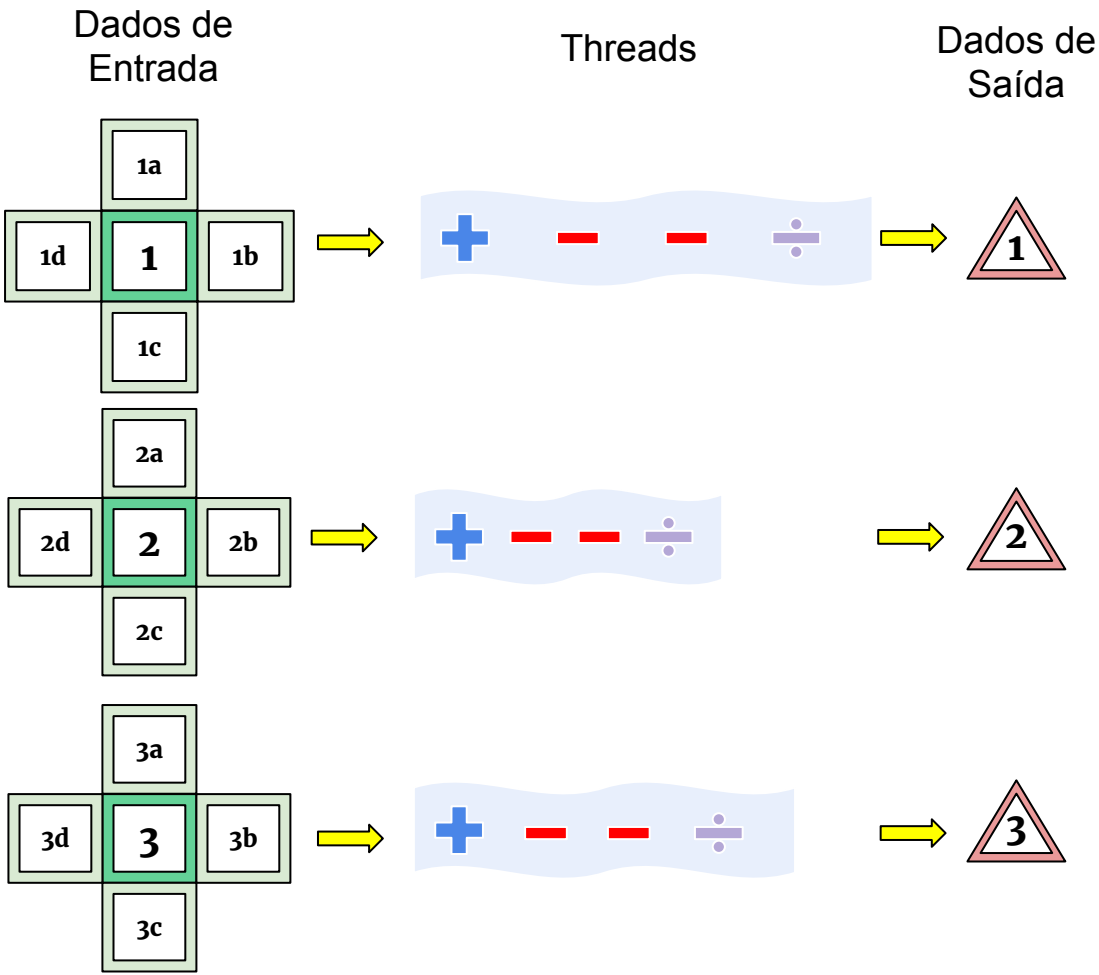


Stencil

- O padrão **Stencil** é uma generalização do MAP, onde cada função acessa não somente um único elemento, mas também seus vizinhos.
  - Condições de borda precisam ser checadas.

- Exemplos
  - Convolução de imagens
  - Detecção de quinas
  - Simulação de nuvens

Stencil


Dados de Entrada
Threads
Dados de Saída
1a
1d
1
1b
1c
2a
2d
2
2b
2c
3a
3d
3
3b
3c
1
2
3

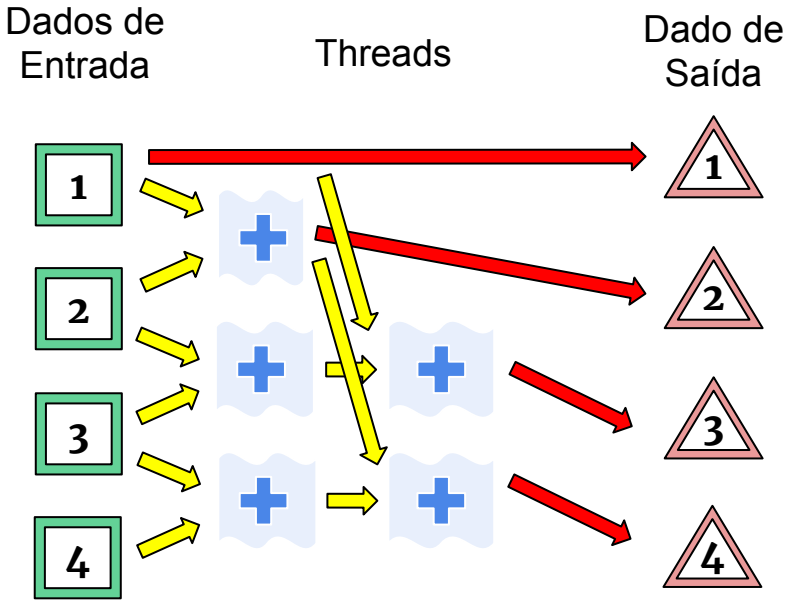
Scan

- O padrão **Scan** computa todas as reduções parciais de uma coleção de elementos, ou seja, para cada saída, uma redução parcial é computada até o elemento atual.

- Exemplo
  - Somatórios parciais

```
for (i = 1; i < n; i++) {
    a[i] += a[i-1];
}
```

Scan



Semântica e Implementação

- Semântica
  - O que o padrão faz?
  - É apenas a funcionalidade do padrão (visão de fora).

- Implementação
  - Como o padrão executa na prática?
  - Várias implementações são possíveis.
  - Visão de dentro do padrão.

Suporte aos Padrões pelos Modelos de Programação

- Open Multi-processing (OpenMP)
  - Fork-Join, Map e Reduce
- Compute Unified Device Architecture (CUDA)
  Open Computing Language (OpenCL)
  - Map
- Intel Thread Building Blocks (TBB)
  - Fork-Join, Map, Reduction, Pipeline, Scan e outros
- Message Passing Interface (MPI)
  - Map, Gather, Scatter, Reduce e outros