Programação Paralela em GPU

Redução e Scan com CUDA

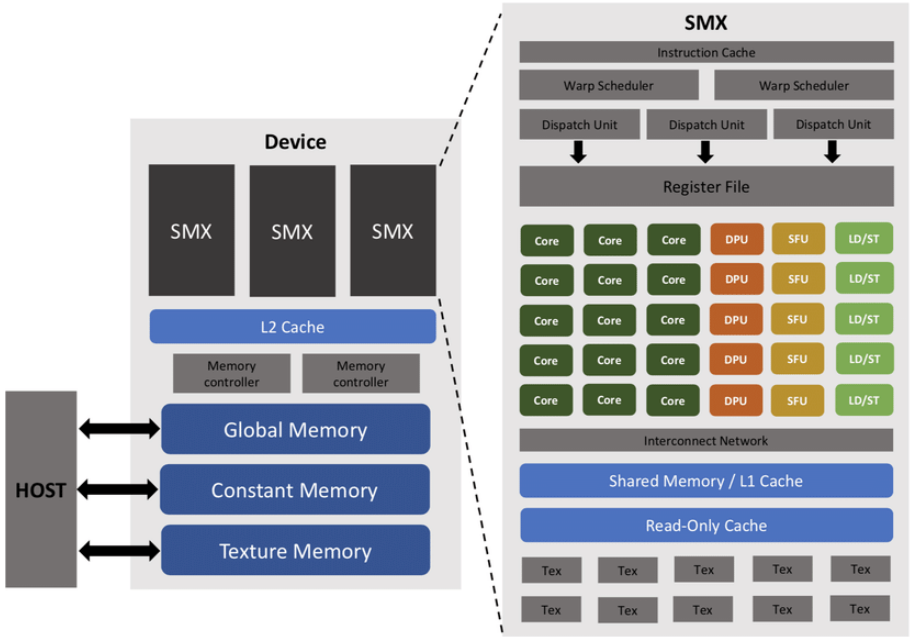
Engenharia de Computação
Lícia Sales Costa Lima
2026-1

Técnica de Otimização: Tiling

O Conceito

Dividir o problema global em blocos menores (tiles) que cabem em memórias rápidas.

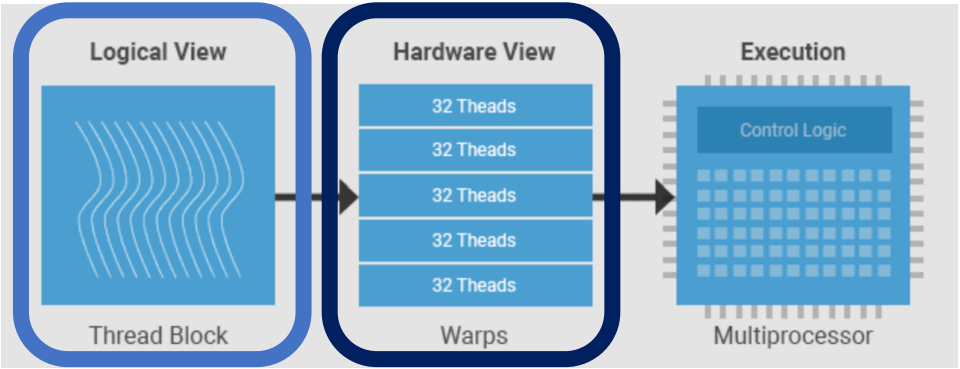
> Memória Global: **LENTA**
>
> Shared Memory: **RÁPIDA**
>
> A Shared Memory é acessível por todas as threads de um mesmo bloco.



Insper

WARPs

- **Warp:** Unidade primária de execução composta por 32 threads.
- **SIMT:** *Single Instruction, Multiple Thread* — todas as threads executam a mesma instrução simultaneamente.



Regra de Ouro

Para maximizar a ocupação e evitar desperdício de recursos, sempre utilize **blocos com múltiplos de 32 threads**.
Isso garante que nenhum warp seja lançado com threads inativas desnecessariamente.

Warps e Paralelismo

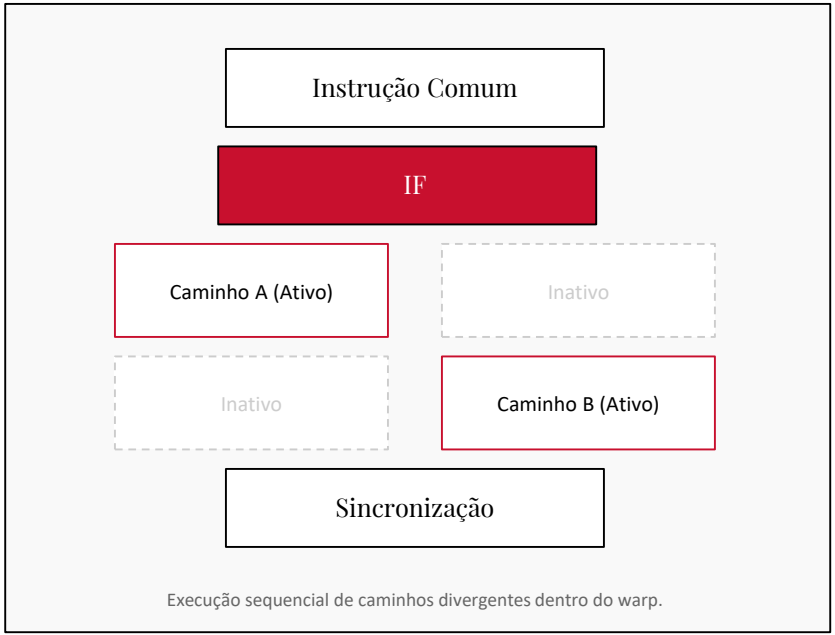
Ocultação de Latência

O hardware alterna instantaneamente entre warps prontos prontos para execução quando um warp atual é bloqueado bloqueado por acessos à memória global.

Divergência de Controle

Ocorre quando threads de um mesmo warp seguem caminhos diferentes em estruturas condicionais (if-else).

Impacto: O hardware serializa os caminhos divergentes, executando um executando um de cada vez, o que reduz a eficiência do paralelismo paralelismo efetivo.



Execução sequencial de caminhos divergentes dentro do warp.

Scan: Conceito e Aplicações

Operação	Resultado	Uso Principal
Redução	Valor Único	Soma total, Min/Max
Scan	Vetor de Prefixos	Somas parciais acumuladas

Enquanto a redução descarta somas intermediárias, intermediárias, o **Scan** preserva a informação de de acúmulo até cada ponto do vetor.

Aplicações Práticas

- Compactação de vetores
- Geração de índices de escrita
- Alocação dinâmica paralela
- Processamento de streams

Scan Otimizado: Duas Fases

Fase 1: Up-Sweep

Funciona como uma redução paralela, combinando pares de elementos até o elementos até o topo da árvore para calcular a soma total do bloco.

- Passo 1: [3,1,7,0] → [4, 7]
- Passo 2: [4, 7] → [11]

Fase 2: Down-Sweep

Distribui os prefixos acumulados de volta para a base da árvore,gerando o scan árvore,gerando o scan exclusivo (exclui o próprio elemento).

- Zera o último elemento
- Propaga somas parciais
- Gera vetor de prefixos

Conclusão

GRANULARIDADE

Ajustar o trabalho por thread é muito importante. Aumentar a granularidade granularidade reduz redundâncias e o o overhead de gerenciamento.

OCUPAÇÃO

A ocupação da GPU depende do uso uso inteligente dos warps e da memória memória compartilhada. Uma ocupação ocupação maior ajuda a esconder latências de memória global.

ALINHAMENTO

O desempenho máximo só é atingido atingido quando o algoritmo respeita a respeita a arquitetura física do hardware hardware