

# Scheduling II

Linguagem de Programação Específica para IA - Lux.AI

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



# Anteriormente

- Exploramos principalmente a estrutura dos loops de um forma mais “macro”
- Iremos então trabalhar com diretivas mais localizadas e explorar outras características do trilema de performance

# reorder

- Serve para reordenar as Vars
- “reorder(var1, var2, ..., varn)” faz com que “var1” seja o loop mais interno e “varn”, o mais externo
- Pode ser útil para melhor se ajustar ao layout de memória dos Buffers ou como passo preliminar ao uso de outras diretivas

# split

- Possibilita separar uma Var em dois loops intermediários
- Não altera ordem da computação
- Normalmente tem mais utilidade se combinado com outras diretivas
  - ◆ split abre possibilidades de experimentação

# **fuse**

- Ao contrário do “split”, funde o loop de duas Vars (sem alterar ordem da computação)
- Pode ter usos mas é mais incomum que “split”

# tile

- Sintaxe simplificada para “split” em duas dimensões + “reorder”
- Computação em regiões retangulares
- Abre possibilidades (como o “split”) e pode ser benéfico para a performance

# vectorize

- Habilita computação com instruções vetoriais (SIMD)
- Utilizado junto com “split”
- Vetorizar a variável mais interna

# parallel

- Marca um loop (representado por uma Var) para execução em paralelo
- Halide irá sinalizar algumas combinações problemáticas de “parallel” com outras diretivas



## Leitura recomendada

- [https://halide-lang.org/docs/tutorial\\_2lesson\\_05\\_scheduling\\_1\\_8cpp-example.html](https://halide-lang.org/docs/tutorial_2lesson_05_scheduling_1_8cpp-example.html)
- [https://halide-lang.org/docs/class\\_halide\\_1\\_1\\_func.html#details](https://halide-lang.org/docs/class_halide_1_1_func.html#details)

# Obrigado pela atenção!

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO

