Ray Tracing em GPU usando CUDA

Gabriel Moreira - 7º Semestre - Engenharia da Computação

Programando em CUDA

O código em CUDA faz uso tanto da CPU, que é chamada de host, quanto da GPU, chamada de device. Esses disposivos não compartilham suas memórias locais, ou seja, para que o dado da GPU seja lida pela CPU, depois do processamento, ele deve ser copiado para a memória da CPU e vice-versa. Um exemplo da relação entre o device e o host pode ser vista na Figura 1.

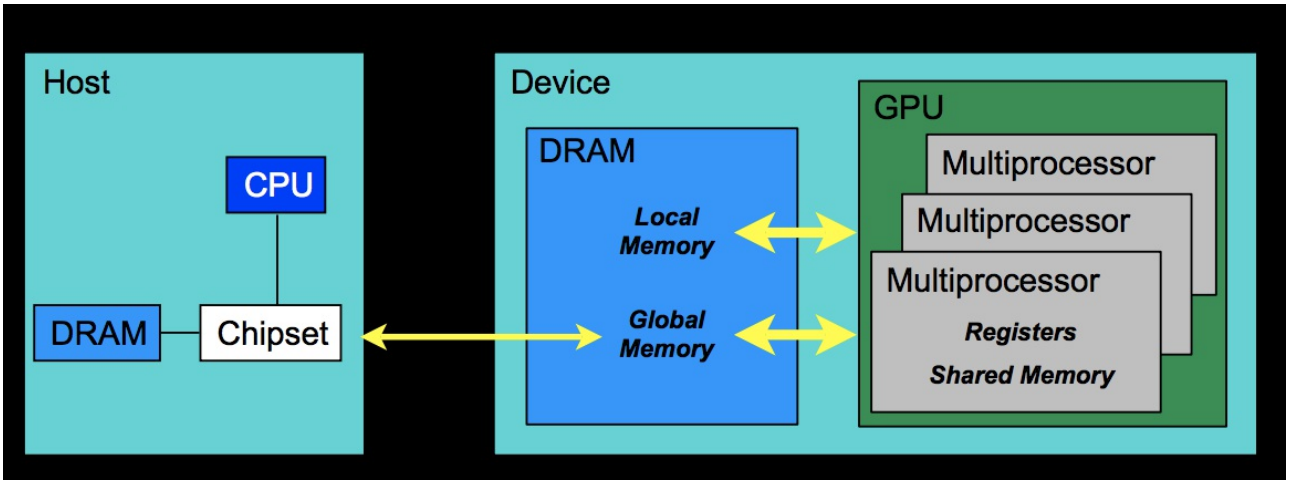

Figura 1: relação device-host

Partes da aplicação na GPU são divididas em porções menores e executadas paralelamente em forma de *kernels*. Esses são executados um de cada vez em diferentes *threads*. As *threads* em GPU possuem pequenos overheads de criação e é fácil de alternar entre elas em relação às de CPU. A placa de vídeos faz uso de milhares de *threads* para aumentar seu desempenho, enquanto uma CPU muti-core possui um número muito menor.

Todas as *threads* na GPU executam o mesmo código, porém elas possuem um *id* que as diferencia e pode ser usado para controlar tais *threads* com maior facilidade.

Kernels executam uma *grid* de blocos de *threads*, sendo que, as *threads* em um mesmo bloco cooperam entre si a partir de uma memória compartilhada. É importante realçar que *threads* em diferentes blocos não cooperam entre si.
Essa relação pode ser vista na Figura 2.

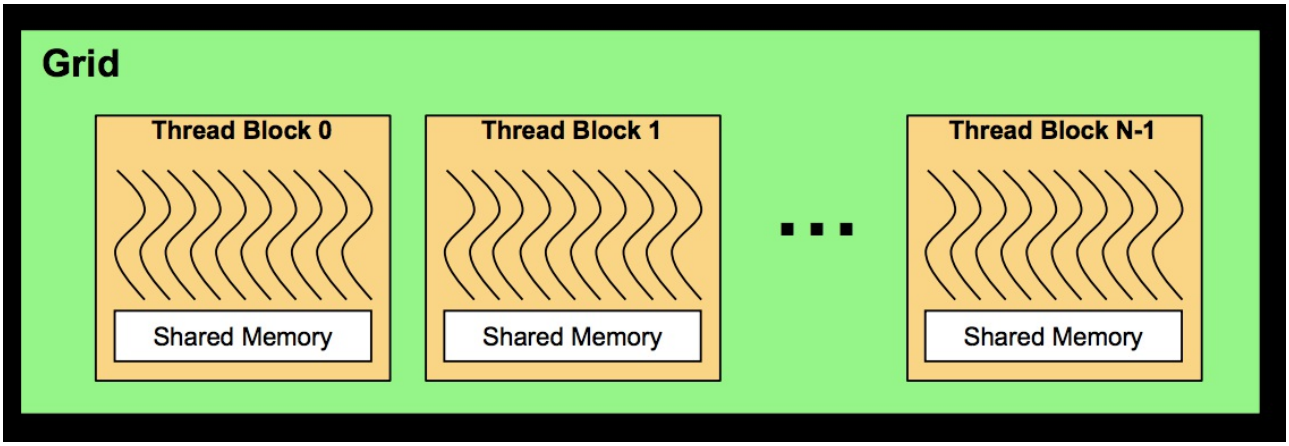

Figura 2: exemplo de kernel grid com suas threads