

1. O que é SIMD ? Dê um exemplo diferente do existente nos slides de aula.

(Single Instruction Multiple Data). Paralelismo de dados

Significa que todas as unidades paralelas compartilham a mesma instrução, mas a realizam em diferentes elementos de dados.

EXEMPLO: processamento de imagens ou algoritmos matemáticos pois nestes casos as aplicações no processador podem ser associadas a uma sub-matriz independente de dados.

2. O que é MIMD ? Dê um exemplo diferente do existente nos slides de aula.

(Multiple Instruction Multiple Data). paralelismo de tarefas

Significa que as unidades paralelas têm instruções distintas, então cada uma delas pode fazer algo diferente em um dado momento.

EXEMPLO: redes de interconexão utilizadas em sistemas MIMD com memória compartilhada.

3. Qual destes modelos de paralelismo é mais adequado para as GPUs ?

SIMD. A GPU tem arquitetura paralela contendo centenas de núcleos, a torna adequada para processar dados em paralelo.

4. O que caracteriza o modelo de programação do OpenMP ?

Especificação para programação paralela em memória compartilhada.

5. (Verdade/Falso) Pode-se executar paralelismo de dados e de tarefas ao mesmo tempo numa aplicação rodando em GPU ? Comente a eficiência da GPU, neste caso.

Falso

As GPUs funcionam muito bem para paralelismo de dados (códigos iguais) e são incapazes de executar tarefas de códigos diferentes em paralelo, ao mesmo tempo.

6. O que é a criação de um contexto na programação em OpenCL ?

OpenCL oferece aos desenvolvedores um ambiente de programação paralela para escrever códigos portáteis para estes sistemas heterogêneos.

7. De que forma um programa-host em OpenCL, se comunica com uma GPU ?

Um programa OpenCL consiste em duas partes: kernels que executam em um ou mais aparelhos OpenCL e um programa host que gerencia a execução dos kernels

8. Um kernel OpenCL corresponde a parte paralelizada executada numa GPU. Um kernel é descrito por um código-fonte parametrizado, representando a implementação de uma função.

(a) O que são instâncias de um kernel ?

São instâncias utilizadas para o processamento de todo o conjunto de dados no momento da execução.

(b) No modelo de execução de OpenCL como são denominadas essas instâncias ? E como essas podem ser agrupadas ?

O OpenCL runtime permite ao desenvolvedor enfileirar comandos para execução nos dispositivos (GPUs), sendo também é responsável por gerenciar os recursos de memória e computação disponíveis.

(c) Como são materializadas essas instâncias paralelizadas nos Elementos de Processamento (EP) do modelo de plataforma quanto a uma GPU ?

Cada dispositivo possui uma ou mais unidades de computação (compute units), sendo estas compostas de um ou mais elementos de processamento (processing elements).

9. Visualize um quadro comparativo dos conceitos de OpenCL e CUDA (Compute Unified Device Architecture).

O modelo de programação em OpenCL é semelhante ao utilizado em CUDA.

10. De que forma OpenCL pode ser utilizado para se implementar jogos (gaming). Pesquise.

O OpenCL™ permite aos programadores preservar o investimento em código-fonte e visar com facilidade CPUs, GPUs e APUs com vários núcleos.

A proposta da OpenCL é prover uma especificação que, quando implementada, permita acessar os diversos dispositivos da mesma forma, podendo-se trocar o dispositivo e executar os mesmos kernels (programas) em outros dispositivos. Na prática, isto significa que o mesmo kernel pode ser compilado e executado em uma GPU conectada a um PC

Fonte: <http://www.amd.com/pt-br/solutions/professional/hpc/opencv> acessado em 09.10.2017