Revisão Alto desempenho

1. Qual das alternativas descreve corretamente a função principal da GPU em sistemas híbridos?
    - A) Gerenciar o sistema operacional
    - B) Executar tarefas sequenciais complexas
    - **C) Executar paralelamente grandes volumes de dados**
    - D) Coordenar os dispositivos periféricos

2. Em arquiteturas integradas CPU/GPU, o principal benefício é:
    - A) Maior latência de acesso à memória
    - B) Necessidade de cópia explícita de dados
    - **C) Comunicação mais rápida e baixo consumo**
    - D) Separação rígida entre CPU e GPU

3. O que caracteriza a memória **global** em CUDA?
    - A) Acesso extremamente rápido
    - B) Compartilhada entre threads de um bloco
    - **C) Alta capacidade e maior latência**
    - D) Exclusiva para cada thread

4. Qual ferramenta **NVIDIA** permite depuração de código CUDA?
    - A) CodeXL
    - **B) CUDA-GDB**

- C) ROCm Profiler
- D) VTune

5. Em CUDA, uma função kernel é definida com o qual qualificador?
    - A) device
    - B) shared
    - **C) global**
    - D) host

6. Qual vantagem do OpenCL em relação ao CUDA?
    - A) Melhor desempenho em GPUs NVIDIA
    - B) Integração mais profunda com hardware proprietário
    - **C) Suporte multiplataforma**
    - D) Sintaxe mais simples que C++

7. O modelo de execução **NDRange** é característico de qual tecnologia?
    - A) CUDA
    - B) OpenCL
    - **C) OpenMP**
    - D) SYCL

8. Quais tecnologias **facilitam a computação paralela em Python**?
    - A) NVCC e NSight
    - **B) joblib e Dask**
    - C) OpenACC e ROCm

- D) SYCL e Kokkos

9. Qual tipo de arquitetura é mais comum em laptops com Apple Silicon?
    - A) GPU discreta
    - B) CPU multicore apenas
    - **C) APU integrada com memória unificada**
    - D) Multi-GPU com barramento PCIe

10. Qual API permite o uso de **aceleradores AMD** e é uma alternativa ao CUDA?

- A) NSight
- B) oneAPI
- **C) ROCm**
- D) cuDNN

Dissertativas (11 a 20)

11. Explique a principal diferença entre memória compartilhada e memória local em CUDA. **Memória compartilhada é acessada por todas as threads dentro de um bloco, ideal para colaboração e comunicação. Já a memória local é privada para cada thread e usada para armazenar variáveis locais temporárias, com latência maior.**

12. Descreva as etapas básicas de execução de um kernel em OpenCL.
    a. As etapas básicas em OpenCL são:
    b. (1) Obter plataforma e dispositivo,
    c. (2) Criar contexto e fila de comandos,
    d. (3) Criar e compilar o programa,
    e. (4) Criar kernel,
    f. (5) Definir argumentos,
    g. (6) Escrever dados nos buffers,
    h. (7) Executar kernel com clEnqueueNDRangeKernel,
    i. (8) Ler resultados com clEnqueueReadBuffer,
    j. (9) Liberar recursos.

13. Compare o uso de `__global__`, `__device__` e `__host__` em CUDA com exemplos de uso.

__global__ define uma função kernel que roda na GPU e é chamada pela CPU.

__device__ é uma função que roda na GPU e é chamada apenas por outras funções da GPU.

__host__ é executada na CPU, podendo ser usada junto a __device__ para funções compartilhadas.

14. Quais são os principais desafios enfrentados por arquiteturas híbridas CPU/GPU atualmente?

    Os principais desafios são: complexidade de programação, diferenças entre APIs e plataformas, latência de transferência entre CPU/GPU, limitações de largura de banda e consumo energético elevado.

15. Como o balanceamento de carga é feito entre CPU e GPU em sistemas híbridos?

    O balanceamento de carga é feito por meio da análise de desempenho e técnicas como offload automático, onde tarefas paralelizáveis são enviadas para a GPU e as demais permanecem na CPU, maximizando o uso dos recursos.

16. Explique a importância da hierarquia de threads em CUDA e como ela impacta o desempenho.

    A hierarquia de threads (threads, blocos, grids) permite organizar o paralelismo, otimizando o uso de memória compartilhada e o desempenho geral. Escolhas erradas podem gerar baixa eficiência e conflitos de acesso.

17. Descreva os principais componentes da API OpenCL utilizados na execução de kernels.

    Principais componentes: cl_platform_id (plataforma), cl_device_id (dispositivo), cl_context (ambiente), cl_program (código-fonte), cl_kernel (função compilada), cl_command_queue (fila de execução), cl_mem (buffers).

18. Diferencie o uso de CUDA e OpenCL do ponto de vista da portabilidade e da curva de aprendizado.

    CUDA é mais simples e integrado ao C++, ideal para NVIDIA. OpenCL é multiplataforma, mas mais complexo e verboso. CUDA oferece maior desempenho em hardware NVIDIA, enquanto OpenCL foca em portabilidade.

19. Quais são os benefícios do uso de GPUs no treinamento de redes neurais em IA?

GPUs oferecem paralelismo massivo, essencial para operações matriciais em redes neurais. Permitem treinar modelos de IA mais rapidamente, e com bibliotecas otimizadas (cuDNN), o desempenho é ainda maior.

20. Comente sobre as tendências futuras das arquiteturas híbridas com foco em computação de alto desempenho.

    Tendências incluem: uso de HBM, maior coerência entre CPU/GPU, integração com ASICs e TPUs, suporte à computação quântica, uso de APIs unificadas como SYCL e oneAPI para facilitar o desenvolvimento heterogêneo.