20 Questions - CUDA and GPU Architectures

Question 1

Quel est le rôle principal du Streaming Multiprocessor (SM) dans une architecture GPU ? / What is the primary role of the Streaming Multiprocessor (SM) in a GPU architecture?

- a) Gérer les opérations de transfert de mémoire / Handle memory transfer operations
- b) Exécuter les threads en parallèle / Execute threads in parallel
- c) Gérer les communications CPU-GPU / Manage CPU-GPU communications
- d) Optimiser les instructions SIMD / Optimize SIMD instructions

Réponse / Answer: b) Exécuter les threads en parallèle / Execute threads in parallel

Question 2

Combien de threads composent un warp dans CUDA ? / How many threads are in a warp in CUDA?

- a) 16
- b) 32
- c) 64
- d) 128

Réponse / Answer: b) 32

Question 3

Quelle est la différence clé entre SIMD et SIMT ? / What is the key difference between SIMD and SIMT?

- a) SIMD exécute des instructions différentes pour chaque thread / SIMD executes different instructions for each thread
- b) SIMT permet aux threads d'avoir des chemins d'exécution indépendants / SIMT allows threads to have independent execution paths
- c) SIMD et SIMT sont identiques / SIMD and SIMT are identical
- d) SIMD gère plusieurs threads dans un seul registre / SIMD handles multiple threads in a single register

Réponse / Answer: b) SIMT permet aux threads d'avoir des chemins d'exécution indépendants / SIMT allows threads to have independent execution paths

Question 4

Dans CUDA, quelle commande est utilisée pour synchroniser tous les threads d'un bloc ? / In CUDA, which command is used to synchronize all threads in a block?

- a) __global__
- b) __syncthreads()
- c) cudaMemcpy
- d) cudaDeviceSynchronize()

Réponse / Answer: b) __syncthreads()

Question 5

Que signifie le terme 'warp divergence' ? / What does the term 'warp divergence' mean?

- a) Une divergence dans les données partagées entre threads / A divergence in shared data between threads
- b) Des threads d'un même warp suivant des chemins d'exécution différents / Threads in the same warp following different execution paths
- c) Des threads s'exécutant sur plusieurs SM en parallèle / Threads executing on multiple SMs in parallel
- d) Une interruption dans le flux d'instructions d'un thread / An interruption in the instruction flow of a thread

Réponse / Answer: b) Des threads d'un même warp suivant des chemins d'exécution différents / Threads in the same warp following different execution paths

Question 1

Quel est le rôle principal du Streaming Multiprocessor (SM) dans une architecture GPU ? / What is the primary role of the Streaming Multiprocessor (SM) in a GPU architecture?

- a) Gérer les opérations de transfert de mémoire / Handle memory transfer operations
- b) Exécuter les threads en parallèle / Execute threads in parallel
- c) Gérer les communications CPU-GPU / Manage CPU-GPU communications
- d) Optimiser les instructions SIMD / Optimize SIMD instructions

Réponse / Answer: b) Exécuter les threads en parallèle / Execute threads in parallel

Question 2

Combien de threads composent un warp dans CUDA ? / How many threads are in a warp in CUDA?

- a) 16
- b) 32
- c) 64
- d) 128

Réponse / Answer: b) 32

Question 3

Quelle est la différence clé entre SIMD et SIMT ? / What is the key difference between SIMD and SIMT?

- a) SIMD exécute des instructions différentes pour chaque thread / SIMD executes different instructions for each thread
- b) SIMT permet aux threads d'avoir des chemins d'exécution indépendants / SIMT allows threads to have independent execution paths
- c) SIMD et SIMT sont identiques / SIMD and SIMT are identical
- d) SIMD gère plusieurs threads dans un seul registre / SIMD handles multiple threads in a single register

Réponse / Answer: b) SIMT permet aux threads d'avoir des chemins d'exécution indépendants / SIMT allows threads to have independent execution paths

Question 4

Dans CUDA, quelle commande est utilisée pour synchroniser tous les threads d'un bloc ? / In CUDA, which command is used to synchronize all threads in a block?

- a) __global__
- b) __syncthreads()
- c) cudaMemcpy
- d) cudaDeviceSynchronize()

Réponse / Answer: b) __syncthreads()

Question 5

Que signifie le terme 'warp divergence' ? / What does the term 'warp divergence' mean?

- a) Une divergence dans les données partagées entre threads / A divergence in shared data between threads
- b) Des threads d'un même warp suivant des chemins d'exécution différents / Threads in the same warp following different execution paths

- c) Des threads s'exécutant sur plusieurs SM en parallèle / Threads executing on multiple SMs in parallel
- d) Une interruption dans le flux d'instructions d'un thread / An interruption in the instruction flow of a thread

Réponse / Answer: b) Des threads d'un même warp suivant des chemins d'exécution différents / Threads in the same warp following different execution paths

Question 6

Quelle architecture GPU a introduit les Tensor Cores ? / Which GPU architecture introduced Tensor Cores?

- a) Fermi
- b) Kepler
- c) Volta
- d) Pascal

Réponse / Answer: c) Volta

Question 7

Combien de cycles d'horloge en moyenne faut-il pour accéder à la mémoire globale dans CUDA ? / How many clock cycles on average are needed to access global memory in CUDA?

- a) 10-20
- b) 50-100
- c) 400-800
- d) 1000-2000

Réponse / Answer: c) 400-800

Question 8

Quel outil de profilage est utilisé pour analyser les performances des kernels CUDA ? / Which profiling tool is used to analyze CUDA kernel performance?

- a) nvcc
- b) cudaMemcpy
- c) nvprof
- d) gprof

Réponse / Answer: c) nvprof

Question 9

Quelle unité exécute les instructions pour les threads dans un warp ? / Which unit executes instructions for threads in a warp?

- a) CUDA Cores
- b) Tensor Cores
- c) Warp Scheduler
- d) Shared Memory

Réponse / Answer: a) CUDA Cores

Question 10

Quel est le rôle principal de la mémoire partagée (shared memory) dans CUDA ? / What is the primary role of shared memory in CUDA?

- a) Stocker les données pour un accès rapide entre threads d'un même bloc / Store data for fast access among threads in the same block
- b) Synchroniser les threads entre différents blocs / Synchronize threads across different blocks
- c) Charger les données de la mémoire globale pour l'utiliser plus tard / Load data from global memory for later use
- d) Remplacer les registres en cas de dépassement / Replace registers in case of overflow

Réponse / Answer: a) Stocker les données pour un accès rapide entre threads d'un même bloc / Store data for fast access among threads in the same block

Question 11

Quelle architecture CUDA a introduit le parallélisme dynamique ? / Which CUDA architecture introduced dynamic parallelism?

- a) Fermi
- b) Kepler
- c) Volta
- d) Ampere

Réponse / Answer: b) Kepler

Question 12

Quelle commande CUDA est utilisée pour allouer de la mémoire sur le GPU ? / Which CUDA command is used to allocate memory on the GPU?

- a) cudaMalloc
- b) cudaMemcpy
- c) cudaFree
- d) cudaSync

Réponse / Answer: a) cudaMalloc

Question 13

Que signifie SIMT dans le contexte de CUDA ? / What does SIMT mean in the context of CUDA?

- a) Single Instruction Multiple Threads
- b) Single Instruction Multiple Tasks
- c) Scalable Independent Multiple Threads
- d) Synchronized Independent Multiple Tasks

Réponse / Answer: a) Single Instruction Multiple Threads

Question 14

Quelle est la latence typique de la mémoire partagée par rapport à la mémoire globale ? / What is the typical latency of shared memory compared to global memory?

- a) Plus rapide de 10 à 20x / 10 to 20x faster
- b) Plus rapide de 100 à 150x / 100 to 150x faster
- c) Plus lente de 2x / 2x slower
- d) Identique / Identical

Réponse / Answer: a) Plus rapide de 10 à 20x / 10 to 20x faster

Question 15

Dans CUDA, que fait la fonction cudaMemcpy ? / What does the cudaMemcpy function do in CUDA?

- a) Alloue de la mémoire sur le GPU / Allocates memory on the GPU
- b) Libère la mémoire sur le GPU / Frees memory on the GPU

- c) Copie les données entre l'hôte et le périphérique / Copies data between host and device
- d) Synchronise les threads sur le GPU / Synchronizes threads on the GPU

Réponse / Answer: c) Copie les données entre l'hôte et le périphérique / Copies data between host and device

Question 16

Comment les threads sont-ils regroupés dans CUDA ? / How are threads grouped in CUDA?

- a) En blocs et grilles / In blocks and grids
- b) En matrices et vecteurs / In matrices and vectors
- c) En pages et clusters / In pages and clusters
- d) En bancs et warps / In banks and warps

Réponse / Answer: a) En blocs et grilles / In blocks and grids

Question 17

Quel est le rôle principal de la mémoire constante dans CUDA ? / What is the primary role of constant memory in CUDA?

- a) Stocker les instructions du programme / Store program instructions
- b) Fournir un accès rapide à des données immuables pour tous les threads / Provide fast access to immutable data for all threads
- c) Synchroniser les threads / Synchronize threads
- d) Remplacer la mémoire partagée / Replace shared memory

Réponse / Answer: b) Fournir un accès rapide à des données immuables pour tous les threads / Provide fast access to immutable data for all threads

Question 18

Quelle est la taille maximale d'un warp dans CUDA ? / What is the maximum size of a warp in CUDA?

- a) 16
- b) 32
- c) 64
- d) 128

Réponse / Answer: b) 32

Question 19

Comment un kernel CUDA est-il défini ? / How is a CUDA kernel defined?

- a) Avec __device__
- b) Avec __global__
- c) Avec __shared__
- d) Avec __host__

Réponse / Answer: b) Avec __global__

Question 20

Qu'est-ce qu'un warp scheduler ? / What is a warp scheduler?

- a) Une unité qui exécute plusieurs threads en parallèle / A unit that executes multiple threads in parallel
- b) Une unité qui ordonne l'exécution des warps / A unit that schedules the execution of warps
- c) Une unité qui gère la mémoire partagée / A unit that manages shared memory
- d) Une unité qui synchronise les threads / A unit that synchronizes threads

Réponse / Answer: b) Une unité qui ordonne l'exécution des warps / A unit that schedules the execution of warps

Question 21

Dans CUDA, que signifie le terme 'global memory coalescing' ? / What does 'global memory coalescing' mean in CUDA?

- a) Une réduction de la latence des threads / A reduction in thread latency
- b) L'accès simultané efficace à la mémoire globale / Efficient simultaneous access to global memory
- c) La synchronisation des threads dans un warp / Thread synchronization within a warp
- d) L'optimisation des instructions dans un warp / Optimization of instructions in a warp

Réponse / Answer: b) L'accès simultané efficace à la mémoire globale / Efficient simultaneous access to global memory