La GPU es una unidad de procesamiento gráfico que permite ejecutar gráficos de alta definición en el PC, que son la demanda de la informática moderna. La tarea principal de la GPU es calcular funciones 3D. Aunque la GPU nació con fines gráficos, ahora ha evolucionado hacia la computación, la precisión y el rendimiento. La evolución de la GPU a lo largo de los años ha sido hacia un mejor rendimiento en coma flotante.

NVIDIA introdujo su arquitectura paralela masiva llamada "CUDA" en 2006-2007 y cambió toda la perspectiva de la programación de la GPGPU. La arquitectura CUDA cuenta con una serie de núcleos de procesamiento que trabajan juntos para masticar el conjunto de datos dado en la aplicación. El GPU computing o GPGPU es el uso de una GPU para realizar cálculos científicos y de ingeniería de propósito general. Desde el punto de vista del usuario, la aplicación es más rápida porque está utilizando el mejor rendimiento de la GPU para mejorar su propio rendimiento.

CUDA: CUDA es la arquitectura de GPU de NVIDIA que aparece en las tarjetas GPU y que se posiciona como un nuevo medio para la computación de propósito general con las GPU. CUDA C/C++ es una extensión de los lenguajes de programación C/C++ para la computación de propósito general. CUDA ofrece al programador la ventaja de una enorme capacidad de cálculo. Esta enorme capacidad de cálculo paralelo la proporcionan las tarjetas gráficas de Nvidia.

ARQUITECTURA: La GPU es una arquitectura masivamente paralela. Las GPUs tienen una gran cantidad de capacidad aritmética. El gráfico 3 muestra la arquitectura de una GPU típica con capacidad CUDA. CUDA puede verse como un conjunto de procesadores de streaming capaces de realizar un alto grado de roscado.

La rejilla: Cada llamada a CUDA desde la CPU se realiza a través de una rejilla. En los sistemas multi-GPU, las rejillas no pueden compartirse entre las GPUs porque utilizan varias rejillas para obtener la máxima eficiencia.

El bloque: Las rejillas se componen de bloques. Cada bloque es una unidad lógica que contiene un número de hilos de coordinación y una cierta cantidad de memoria compartida. Al igual que las rejillas no se comparten entre GPUs, los bloques no se comparten entre multiprocesadores.

Los hilos: Los hilos se ejecutan en los núcleos individuales de los multiprocesadores, pero a diferencia de las rejillas y los bloques, no están restringidos a un solo núcleo. Los hilos disponen de una cierta cantidad de memoria de registro. El número de hilos que pueden ejecutarse simultáneamente en un bloque se determina la memoria compartida que se especifica y denota la ocupación de ese bloque.

Transcodificación rápida de vídeo: la potencia de cálculo de las GPU puede aprovecharse para transcodificar vídeo mucho más rápido que antes. Otras tareas comunes como el cálculo de valores propios, o las descomposiciones SVD, u otras matemáticas matriciales pueden utilizar CUDA para acelerar los cálculos.

Ventajas: Con CUDA, el lenguaje de alto nivel C puede utilizarse fácilmente para desarrollar aplicaciones y, por tanto, CUDA proporciona flexibilidad. CUDA proporciona un tamaño considerable de memoria compartida.

Limitaciones: CUDA puede provocar cuellos de botella en la GPU y la CPU debido a la latencia entre la GPU y la CPU. Los hilos sólo deben ejecutarse en grupos de 32.