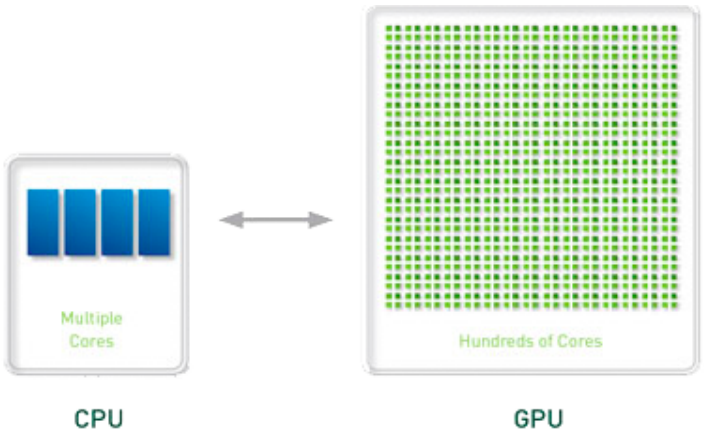
Introduction summary :

What is GPU?

A Graphics Processing Unit (GPU) is a single-chip processor primarily used to manage and boost the performance of video and graphics. GPU features include.To have the idea about GPU let me consider few features.

- •2-D or 3-D graphics
- •Digital output to flat panel display monitors
- •Texture mapping
- •Application support for high-intensity graphics software

This term is also called as virtual processing unit(VPU).A GPU is not only used in a PC on a video card or motherboard;it is also used in mobile phones, display adapters, workstations and game.It just lessens the work of CPU



A CPU consists of four to eight CPU cores, while the GPU consists of hundreds of smaller cores. Together, they operate to crunch through the data in the application. This massively parallel architecture is what gives the GPU its high compute performance.

So, What is GPU computing model?

GPU computing is the use of a GPU (graphics processing unit) as a co-processor to accelerate CPUs for general-purpose scientific and engineering computing.

Previous works have considered more complex,warp-speccialized kernels based on producer-consumer named barriers(that is available on current hardware)

what is warp-specialized kernels?

What is GPU warp?A “warp” (or “wavefront”) is the most basic unit of scheduling of GPU. Other equivalent definitions include: “is the smallest executable unit of code” OR “processes a single instruction over all of the threads in it at the same time” OR “is the minimum size of the data processed .”

Optimizing the performance of a GPU kernel is primarily about managing constrained resources such as registers, shared memory, instruction issue slots, and memory bandwidth. Warp specialization allows subsets of threads within a CTA to have their behaviour tuned for a particular purpose which enables more efficient consumption of constrained resources.(akanksha’s defination)

Warp specialization improves performance by allowing the compiler to do a better job of instruction scheduling and resource allocation. Warp specialization separates memory and compute operations into two different instruction streams.

While there are several APIs for programming GPUs, they all implement variations of the same programming model. We use CUDA as a proxy for the standard GPU programming model as it is the only interface that currently supports the synchronization primitives necessary for warp specialization.

Compute Unified Device Architecture(CUDA)

In CUDA(CUDA is a development toolchain for creating programs that can run on nVidia GPUs, as well as an API for controlling such programs from the CPU.It gives good performance), the host refers to the CPU and its memory, while the device refers to the GPU and its memory. Code run on the host can manage memory on both the host and device, and also launches kernels which are functions executed on the device. These kernels are executed by many GPU threads in parallel.

We have a doubt now,is CUDA C or C++?

Not realized by many,**CUDA** is actually two new programming languages, both derived from **C++**. One is for writing code that runs on GPUs and is a subset of **C++**. Its function is similar to HLSL (DirectX) or Cg (OpenGL) but with more features and compatibility with **C++**.

How many threads is a warp?

On the hardware side, a **thread** block is composed of '**warps**'. A **warp** is a set of 32 **threads** within a **thread** block such that all the **threads in a warp** execute the same instruction. These **threads** are selected serially by the SM. Consider a **warp** of 32 **threads** executing an instruction.

Named barriers?

These are limited physical resources also a thread synchronization construct .

So 3 Important properties to check for warp-specialized kernels

Deadlock Freedom:m aking sure that the use of named barriers does'nt cause deadlocks.

Safe Barrier Recycling :making sure whether or not Ids of named barriers are properly re-used.

Race Freedom: Making sure whether or not named barriers are free of racing.

What is WEFT?

As we all know it is a verifier that reserves all the above three properties also complete and efficient which runs on warp-specialized kernels.

>It has solved persisted bugs of many years.

>No False – positives,reported actual bugs

>practically efficient for verifying real codes.

Just for knowledge-->(**warp scheduler** distributes **warps** of 32 threads to its execution units. Threads are scheduled in groups of 32 threads called **warps**.)