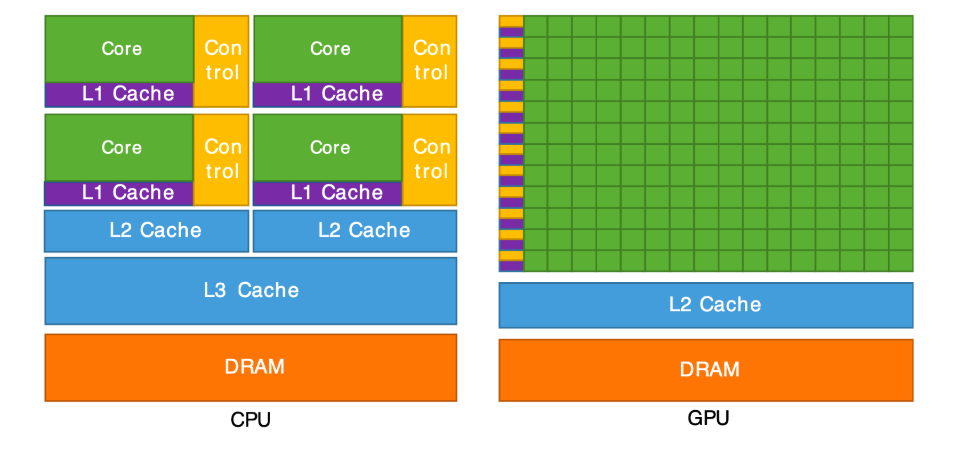
括号内：补充说明，可以供作讲稿参考，不一定要放到PPT里面

1. GPU的体系结构特点

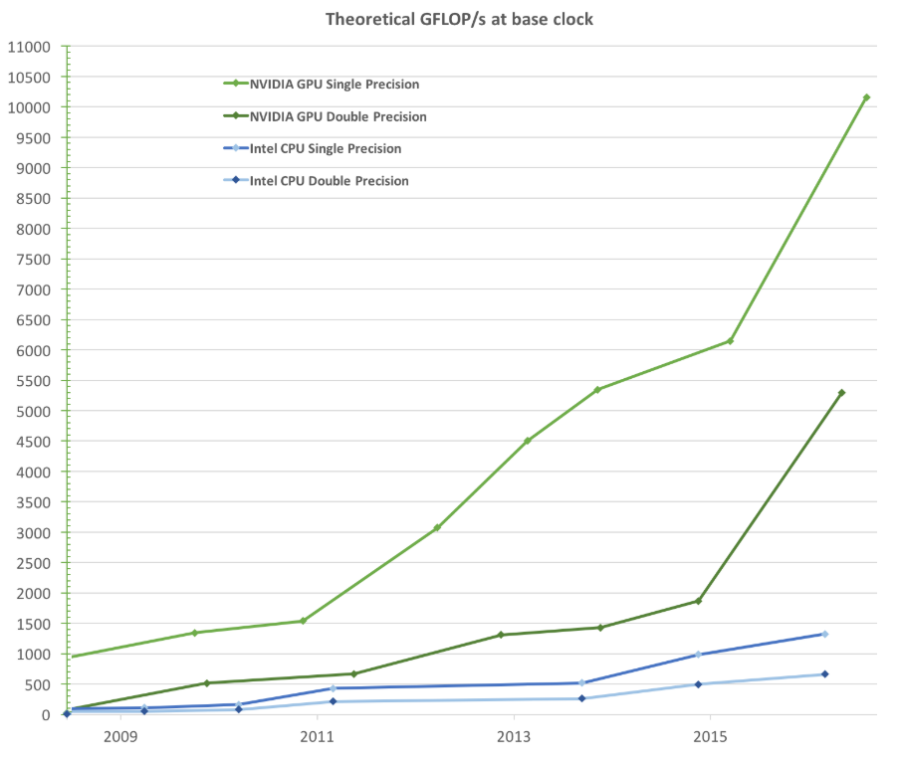
（最初是为了渲染图形而设计的，也叫做图形处理器）



GPU的体系架构中的⼀个最⼤特点就是相比于CPU,增加了⼤量的运算单元ALU，因此能够比CPU提供更高的指令吞吐量和内存带宽。

（具体来看它们硬件结构上的主要差异：绿色代表计算单元ALU，黄色代表控制模块，紫色代表L1缓存，蓝色代表更高级的缓存，橙色代表DRAM。可以看出，GPU的计算单元更多，但是更小更简单，而且一行计算单元只有一个控制模块，也可以看出GPU是专为高度并行化的工作负载而设计的）

因此，CPU主要是来处理复杂的逻辑控制，而GPU则可以做大规模的并行计算 。GPU和CPU混合设计出的系统可以最大化整体性能。



不同 CPU 和 GPU 的原始计算速度，显示了GPU的计算潜力。

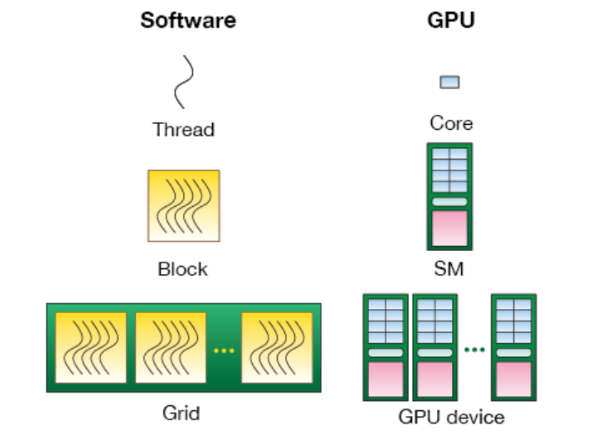
- 顺便一提，CUDA：首批能够进行矩阵操作、快速傅里叶变换和解密算法变成任务的指令集之一，因此可以在GPU上应用，是GPU的关键支持技术之一。

1. GPU内存层级结构介绍（\*）

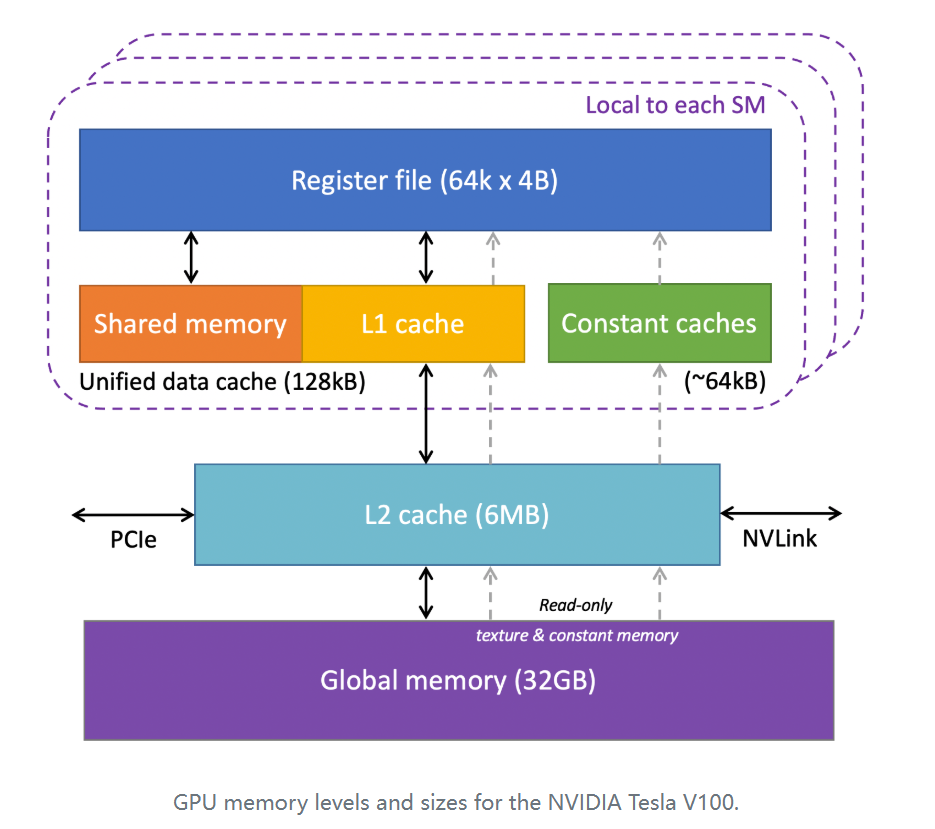
GPU上运行的程序由大量的线程并发执行。这些线程将被划分为线程块，由SM（streaming multiprocessor流式多处理器）处理。

线程块之间是完全独立的了。然后线程块内部连续的32个线程又称作一个warp（线程组），作为数据访问的单位。通过线程组的设置，可以将多个线程的分散访问合并，最大化带宽利用率。

线程的软硬件对应关系：



（不同厂商推出的GPU体系结构不尽相同，但总体框架基本类似。此处以NVIDIA Tesla V100为例。）



（这个图最好每页ppt都放一个感觉比较方便pre）

内存层级大致分为三个部分（从上到下）：local to each SM、L2 Cache、Global memory。

**SM内部**

**Register File**表示直接馈入SM的内存区域。它被组织成32个存储体，正好对应一个warp的32个线程。

**L1 Cache** 指通常的片上存储位置，用于快速访问最近从主存储器 (读取或写入的数据。此外，当活动数据量超过 SM 的寄存器文件可以容纳的数据量时，L1 充当溢出区域，这种情况称为“寄存器溢出”。溢出的数据也是以组的形式。

**Shared Memory** 是物理上与 L1 缓存位于同一内存中的内存区域，但与 L1 的不同之处在于，其所有数据都可以由线程块中的任何线程访问。这允许线程相互通信和共享数据。

* （L1作为cache 读写由硬件单元控制。sharedmemory可以由程序员显式控制，可以采取更灵活的访问策略。）

**Constant Caches**  与全局内存中声明为只读常量的变量有关的特殊缓存。此类变量可由线程块中的任何线程读取。这些缓存的主要和最佳用途是将单个常量值广播给一个线程组的所有线程。

**L2 Cache**

另一个片上缓存，用于保留在 SM 和主内存之间来回传输的数据副本。与 L1 一样，L2 缓存旨在加快后续重新加载速度。但与 L1 缓存不同的是，只有一个 L2 缓存由所有 SM 共享。L2 缓存也位于通过 PCIe 或 NVLink 在设备上或从设备移出的数据的路径中。

**Global Memory**

代表设备主内存的大部分，相当于基于 CPU 的处理器中的 RAM。其中包括本地、纹理和常量内存的区域。

**Texture and Constant Memory** **纹理和常量内存**是设为只读的主内存区域。当提取到 SM 时，带有“纹理”或“常量”声明的变量可由内核中的任何线程读取，用作扩展类型的共享内存。纹理内存缓存在 L1 中，而常量内存缓存在常量缓存中。

**Local Memory** **本地内存** 对分配给每个 SM 的主内存的特殊映射区域。每当“寄存器溢出”溢出特定 SM 上的 L1 缓存时，多余的数据会进一步卸载到 L2，然后卸载到“本地内存”。重新加载溢出寄存器的性能损失对于必须遍历的每个内存级别来说都会变得更大。

L2 缓存作为中间商（？）可能保存两种类型的数据：任何 SM 均可访问的数据（全局、纹理和常量内存）以及仅每个 SM 可单独访问的数据（本地内存）。

1. 对于全局数据，一致性完全通过 L2 来维护。每当 SM 写入全局数据时，包含该数据的 L1 缓存行也要立刻上传到L2（类似于云盘？）。这样所有的SM都能通过L2看到相同的最新值。

2. 相反，当 SM 写入对应的本地内存时，L1 缓存行到 L2 的写回可能会被延迟，可以减少频繁写入，节省带宽和功耗。因为SM对应的本地内存是独有的，所以也不会引发一致性问题。

参考资料：

<https://www.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFD&dbname=CJFDLAST2018&filename=JSJA201712001&uniplatform=OVERSEA&v=zMyM6_bqiTk8c38dW9_eXxCLbVHTC4PVeP-e10Wr-F1U_ZU6vAdGA-rqJIaRQ5FI>

<https://docs.nvidia.com/cuda/cuda-c-programming-guide/>

<https://www.arccompute.io/arc-blog/gpu-101-memory-hierarchy>