





























**使用计算着色器的好处**  
过去，物理模拟主要在 CPU 上运行。GPU 的计算能力仅足以处理图形工作，并且管道中的大多数阶段由只能执行单一任务的专用硬件块实现。随着 GPU 的发展，管道阶段逐渐转移到可以执行不同任务的通用计算块。  
这种灵活性和计算能力的提升使得引擎开发者能够将部分工作负载转移到 GPU 上。除了原始性能外，在 GPU 上运行某些计算还可以避免从 CPU 内存到 GPU 内存的昂贵复制。内存速度的提升速度不及处理器速度，因此尽可能减少设备之间的数据传输是应用程序性能的关键。  
在我们的示例中，布料模拟需要更新所有顶点的位置，并将更新后的数据复制到 GPU。根据网格的大小和要更新的网格数量，这可能会占用相当大比例的帧时间。  
这些工作负载在 GPU 上也可以更好地扩展，因为我们可以并行更新更多的网格。

**计算着色器概述**  
GPU 的执行模型称为 ​**单指令多线程（SIMT）​**。它与现代 CPU 提供的 ​**单指令多数据（SIMD）​** 类似，即通过单条指令操作多个数据项。  
然而，GPU 在单条指令中操作的数据点数量更多。另一个主要区别是，与 SIMD 指令相比，GPU 上的每个线程更加灵活。GPU 架构是一个引人入胜的话题，但其范围超出了本书的范围。我们将在本章末尾提供进一步阅读的参考资料。  
**注意**：  
线程组的名称因 GPU 厂商而异。你可能会在他们的文档中看到术语 ​**warp** 或 ​**wave**。为了避免混淆，我们将使用 ​**线程组**。





























