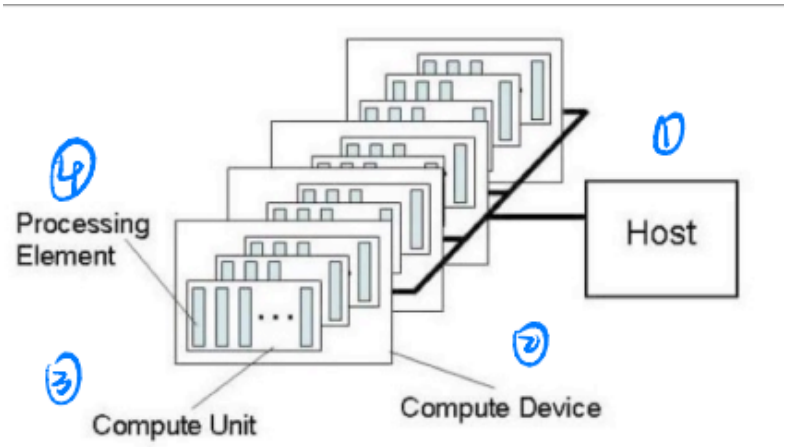
10.2 OpenCL

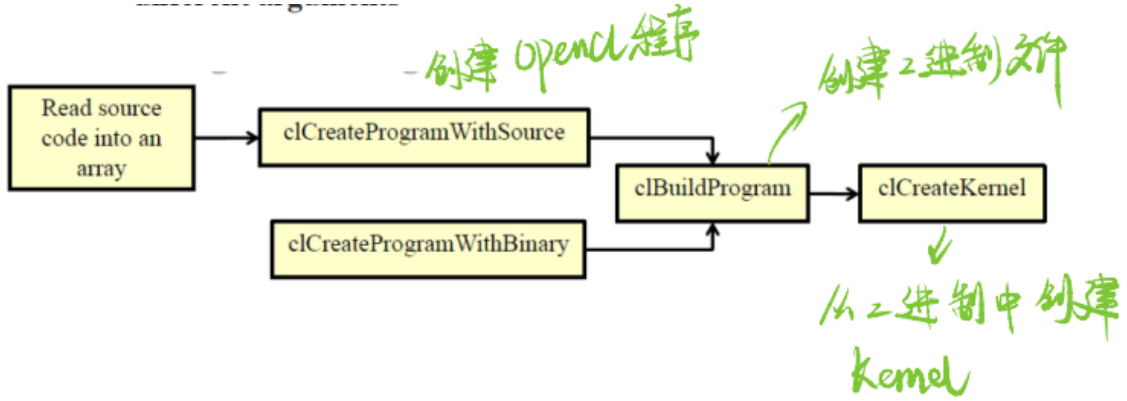
- 概览
  - OpenCL 允许在异构设备上进行并行计算
    - CPU、GPU、其他处理器（Cell、DSP、FPGA 等）
    - 提供可移植的加速代码
  - OpenCL 中的基本概念
    - 平台型号
    - 执行模型
    - 内存模型
    - 编程模型
- 并行软件——SPMD
  - 使用单程序多数据 (SPMD) 编程模型编写的 GPU 程序（内核）
    - SPMD 独立执行同一程序的多个实例，其中每个程序处理数据的不同部分
  - 对于数据并行的科学和工程应用，将 SPMD 与循环带状挖掘相结合是一种非常常见的并行编程技术
    - 消息传递接口 (MPI) 用于在分布式集群上运行 SPMD
    - POSIX 线程 (pthreads) 用于在共享内存系统上运行 SPMD
    - 内核在 GPU 中运行 SPMD
  - 在向量加法示例中，每个数据块都可以作为独立线程执行
  - 在现代 CPU 上，创建线程的开销如此之高，以至于块需要很大
  - 在实践中，通常是几个线程（大约与 CPU 内核的数量一样多），每个线程都有大量工作要做
  - 对于 GPU 编程，线程创建的开销很低，因此我们可以在每次循环迭代中创建一个线程
- 平台模型
  - 每个 OpenCL 实现（即来自 AMD、NVIDIA 等的 OpenCL 库）定义了使主机系统能够与支持 OpenCL 的设备交互的平台
    - 目前，每个供应商每个实施只提供一个平台
  - OpenCL 使用“可安装的客户端驱动程序”（Installable Client Driver）模型
    - 目标是允许来自不同供应商的平台共存
    - 当前系统的设备驱动模型不允许不同供应商的 GPU 同时运行
  - 该模型由连接到一个或多个 OpenCL 设备的主机组成
  - 一个设备被划分为一个或多个计算单元
  - 计算单元被划分为一个或多个处理元素
    - 每个处理元件都有自己的程序计数器



- 主机/设备
  - 主机是 OpenCL 库在其上运行任何
    - 适用于 NVIDIA 和 AMD 的 x86 CPU
  - 设备是库可以与之通信的处理器
    - CPU、GPU 和通用加速器
  - 对于 AMD
    - 所有 CPU 都组合成一个设备（每个内核都是一个计算单元和处理元件）
    - 每个 GPU 都是一个单独的设备
- 程序
  - 程序对象基本上是 OpenCL 内核的集合
    - 可以是源代码（文本）或预编译的二进制文件——CUDA只能运行预编译好的
    - 还可以包含常量数据和辅助功能
  - 创建程序对象需要读取字符串（源代码）或预编译的二进制文件
  - 编译程序
    - 指定目标设备
      - 为每个设备编译程序
    - 传入编译器标志（可选）
    - 检查编译错误（可选，输出到屏幕）
  - 创建程序——函数
    - 此函数从源代码字符串创建程序对象
      - count 指定字符串的数量
      - 用户必须创建一个函数以将源代码读入字符串
    - 如果字符串不是以 NULL 结尾的，则长度字段用于指定字符串长度
  - 编译程序——函数
    - 此函数为上下文中的每个设备编译并链接来自程序对象的可执行文件
    - 如果提供了 device_list，则仅针对那些设备
    - options 参数可以提供可选的预处理器、优化和其他选项
  - kernel

- 内核是在 OpenCL 设备上执行的程序中声明的函数
  - 内核对象是一个内核函数及其相关参数
- 内核对象是从编译的程序创建的
- 必须将参数（内存对象、原语等）与内核对象显式关联起来

- **运行时编译过程**



- 编译程序和创建内核的开销很大
  - 每个操作只需执行一次（在程序开始时）
- 通过设置不同的参数，内核对象可以被多次重用

- 线程结构
  - 通常编写大量并行程序，以便每个线程计算问题的一部分
    - 对于向量加法，我们将从两个数组中添加相应的元素，因此每个线程将执行一次加法
  - 如果我们从视觉上考虑线程结构，线程通常会以与数据相同的形状排列

- 内存模型
  - OpenCL 内存模型定义了各种类型的内存（与 GPU 内存层次结构密切相关）

Memory	Description
Global	Accessible by all work-items
Constant	Read-only, global
Local	Local to a work-group
Private	Private to a work-item

  - 内存管理是显式的
    - 必须将数据从主机内存移动到设备全局内存，从全局内存到本地内存，然后返回
  - 工作组被分配在计算单元上执行
    - 不同工作组之间没有保证的通信/一致性（OpenCL 规范中没有软件机制）

- 编程模型
  - 数据并行
    - 工作项和内存对象中的元素之间的一对一映射
    - 工作组可以显式定义（如 CUDA）或隐式定义（指定工作项的数量，OpenCL 创建工作组）
  - 任务并行
    - 内核独立于索引空间执行
    - 表示并行性的其他方式：将多个任务排入队列，使用特定于设备的向量类型等。
  - 同步
    - 可能在工作组中的项目之间
    - 可能在上下文命令队列中的命令之间
- 总结
  - OpenCL 提供了主机与加速器设备交互的接口
  - 创建一个包含执行 OpenCL 程序所需的所有信息和数据的上下文
    - 创建的内存对象可以在设备上和设备上移动
    - 命令队列允许主机请求设备执行的操作
    - 程序和内核包含设备需要执行的代码

以上内容整理于 幕布文档