

# 计算机科学进展

---

## 智能编程语言

### 智能应用编程接口

传统计算机系统是计算机的控制和计算部分都由计算机的CPU控制，编程接口会相对比较简单。而因在智能计算系统中很多都是异构的系统，主控和计算是由两个不同的部分完成的。比如主控是由CPU完成，而计算是由另一个深度智能计算DLP完成，是有两个分离的部分完成。因此，需要接口和运行池等部分对于深度学习处理器进行管控，对它的任务进行分配管理 在DLP上执行的任务叫Kernel

- Kernel函数接口
  - Kernel指的是在DLP上执行的任务。
  - 为了充分利用并行资源，需要在Kernel内部对任务进行有效切分，同时在主机端配置和调用相应的Kernel函数接口
  - 主机端Kernel函数接口
    - 用于将智能编程语言编写的程序加载到深度学习处理器上执行；与Kernel函数相关的接口主要关注Kernel参数设置和Kernel调用
    - GetKernerlParamBuffer(KernelParamsBuffer\_t\*params)
      - 返回一个KernelParamsBuffer参数块，用于存放传递给设备端Kernel函数的各种参数。
    - CopyKernerlParamBuffer(KernelParamsBuffer\_t dstbuf,KernelParamsBuffer\_t srcbuf)
      - 用于完成两个Kernel参数块中的数据拷贝，也是为KernelParamsBuffer初始化的一种方式
    - KernelParamsVufferAddParam(KernelParamsBuffer\_t params, void\* data, size\_t bytes)
      - 用于对于Kernel的参数块进行设置的一个主要函数，相当于往其中增加一些常量的参数，运用在传输运行给设备中的Kernel函数增加参数。
    - Destroy
      - 销毁KernelParamsBuffer
    - InvokeKernel()
      - 启动Kernel执行的核心函数
      - function 表示要启动的Kernel函数的参数名
      - dim 表示的是任务的规模 有x,y,z三个维度
      - Params 传递的参数
      - functionTYpe 包含的是类型，表示单核还是多和并行任务
      - queue 是任务的执行队列。

为了提高速度，要尽可能减少InvokeKernel次数。因为InvokeKernel中包含了很多调用的过程，执行操作较为复杂。由于智能计算系统所追求的是速度上的提升，因此，可以在优化提升性能的过程中，一部分的重点也是在于减少InvokeKernel的次数。举例来说可以吧几个简单的Kernel函数合称为一个整体的Kernel，通过一次调用，代替原来的调用多次，实现性能的提升。

- 运行时接口
  - 设备管理：主要涉及初始化，设备设置，设备销毁等操作
    - init
    - count

- handle
- devic
- Destroy():所销毁的不是芯片本身，而是销毁芯片中所存储的资源。释放相关的数据结构的操作
- 队列管理
  - 在智能计算系统中，队列是一个很重要的概念。在运行时，是先把Kernel丢入到队列中，再在运行的过程中从队列取出Kernel进行运行。队列适用于执行任务的环境。计算任务可以下发到队列中执行。同一个队列可以容纳多个任务。也可以有多个朋友队列。具体来说，队列具有以下属性：
    - 串行行：下发到同一个队列中的任务，按下发顺序串行执行
    - 异步性：任务下发到队列是异步过程，即下发完成后程序控制流回到主机，主机程序继续往下执行。运行时环境提供队列的同步接口SyncQueue用于等待队列中所有任务完成。
    - 并行性：不同队列中的任务并行执行。如果希望任务键并行执行，用户可以创建多个队列并将任务分配到不同的队列中。
    - 举个例子：如果不加同步有3个Kernel被放入了队列中去，在队列中Kernel的执行速率是位置的。在Kernel2执行完了之后，你不知道Kernel1有没有执行完毕；在Kernel3执行完了之后，你也不知道Kernel1有没有执行完毕。如果希望确保队列中的Kernel是顺序执行的，你只要增加一个syncQueue就可以实现了一个顺序功能。在SyncQueue之前的没有执行完就不会执行下一个Kernel函数。这样保证了Kernel函数在队列中的执行顺序。
- 内存管理
  - 主机端内存管理、设备端内存管理和主机与设备端内存拷贝三类：
    - hostMalloc主机端
    - hostFree主机端
    - dev..设备端
    - Memcpy设备端内存拷贝

## 智能应用功能调试

### 配套的系统实现功能调试

### 智能应用性能调优