异构数据中心编程环境
HeterML

崔慧敏

中国科学院计算技术研究所

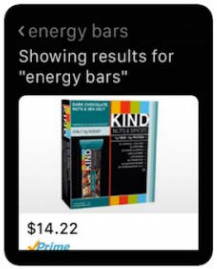
2017.10


AlphaGo

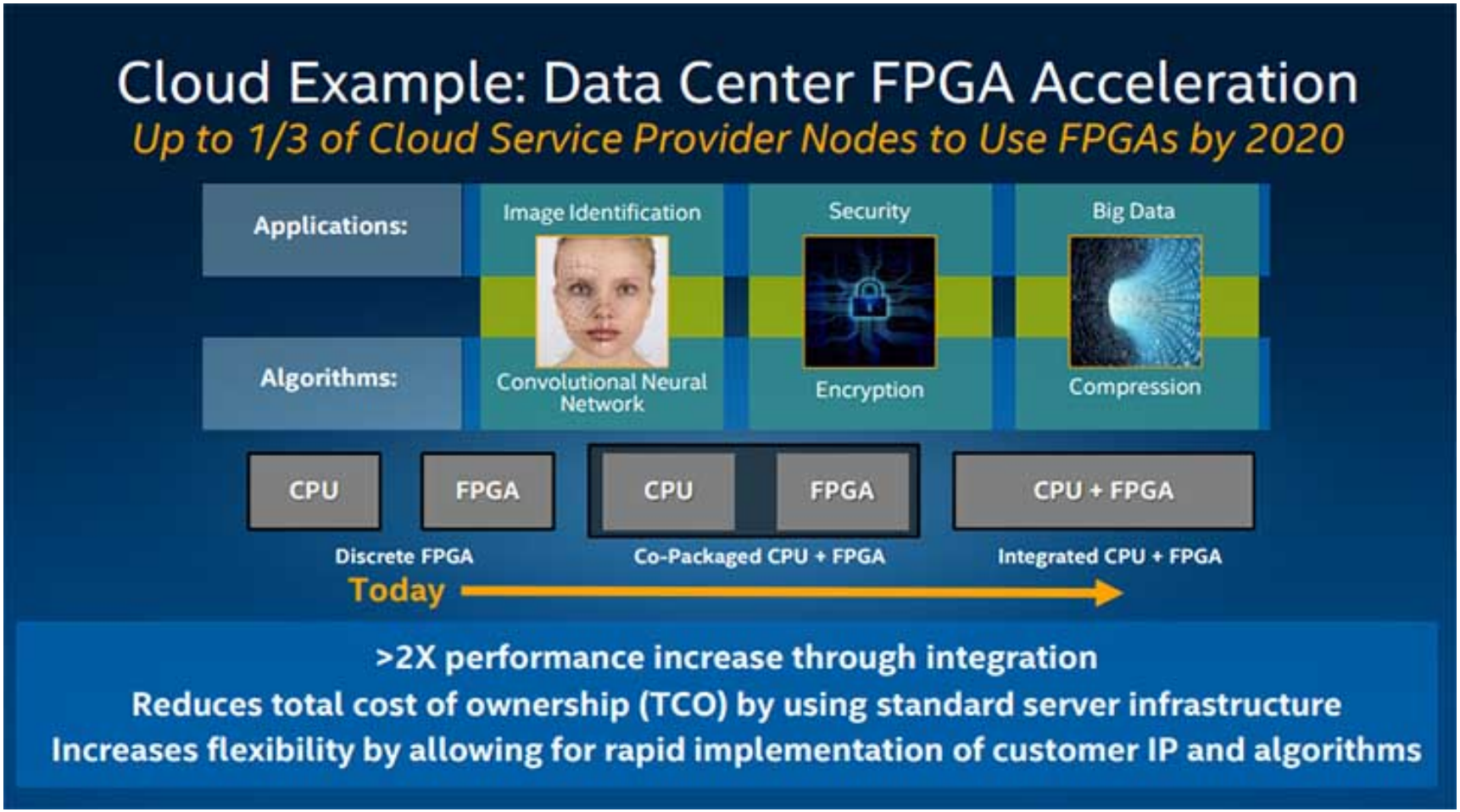

Google
AdSense


amazon
bing
GROUPON
facebook
IMDb
skype
Dropbox
Pinterest
CNN
twitter
Google
yelp
AP
FANDANGO
tripadvisor
amazonkindle
Google+

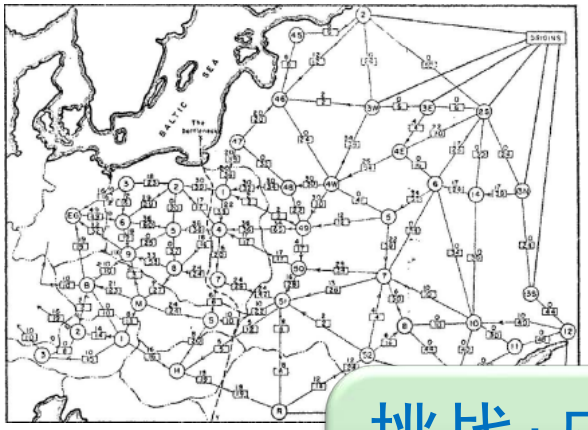

click!
amazon
IMDb
AP
CNN
tripadvisor


energy bars
Showing results for
"energy bars"
KIND
$14.22

异构数据中心

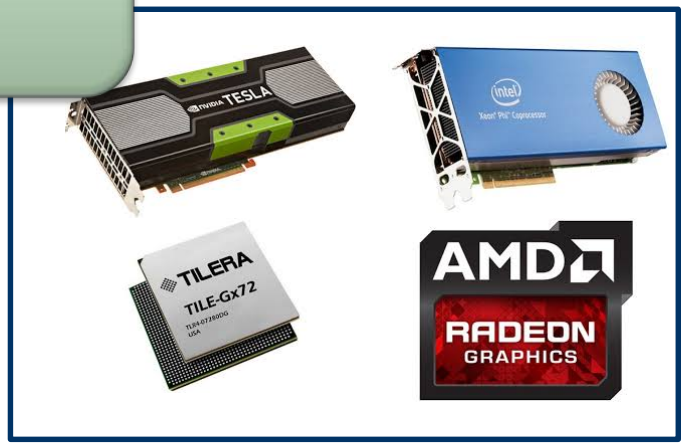

Cloud Example: Data Center FPGA Acceleration
Up to 1/3 of Cloud Service Provider Nodes to Use FPGAs by 2020
Applications:
Image Identification
Security
Big Data
Algorithms:
Convolutional Neural Network
Encryption
Compression
CPU
FPGA
CPU
FPGA
CPU + FPGA
Discrete FPGA
Co-Packaged CPU + FPGA
Integrated CPU + FPGA
Today
>2X performance increase through integration
Reduces total cost of ownership (TCO) by using standard server infrastructure
Increases flexibility by allowing for rapid implementation of customer IP and algorithms

异构数据中心编程



挑战：应用retargetability

应用→芯片



两步走

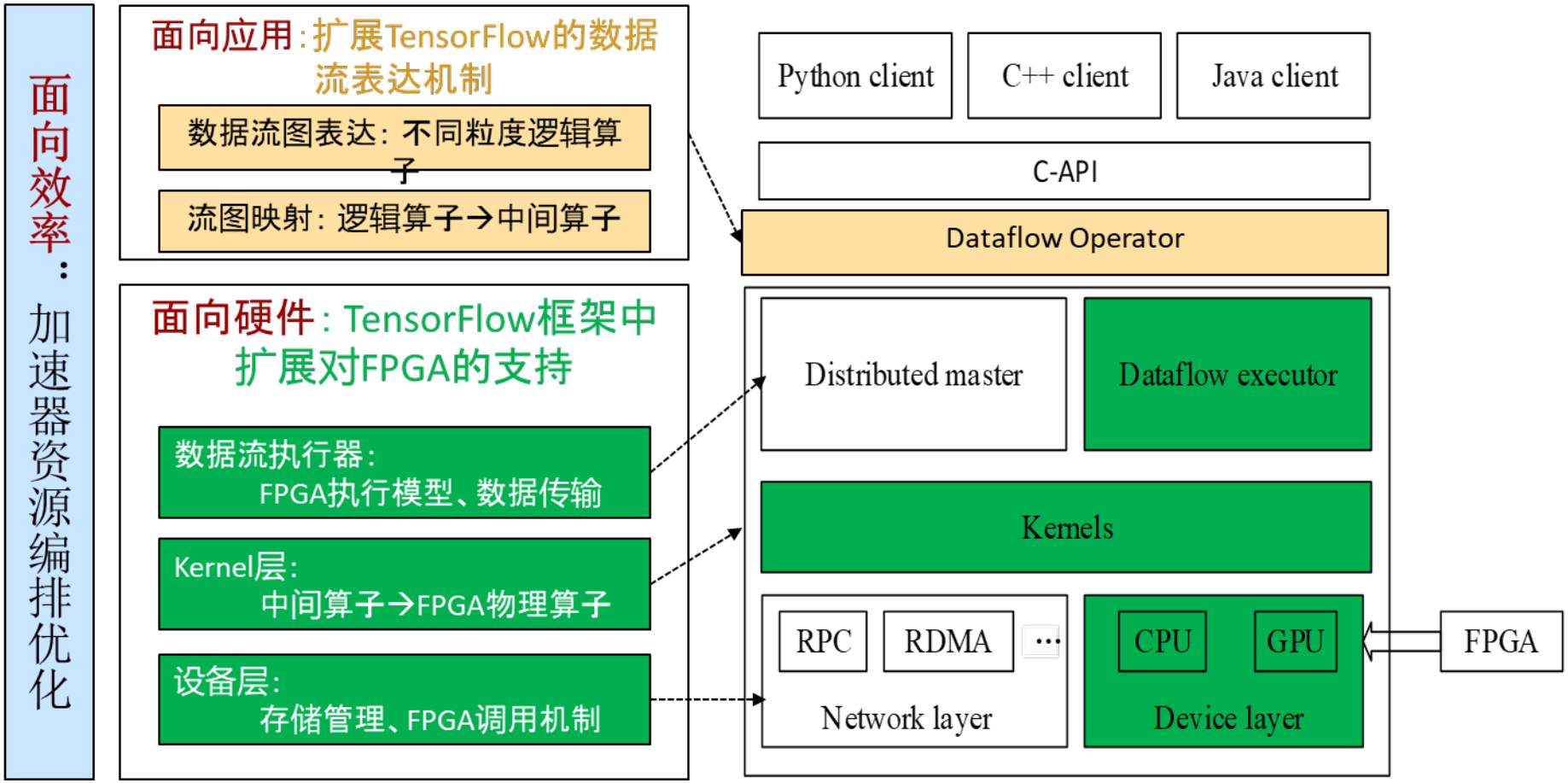
- 第一步：扩展TensorFlow以支持FPGA

  依然是一个针对某个(些)平台特定的编程框架

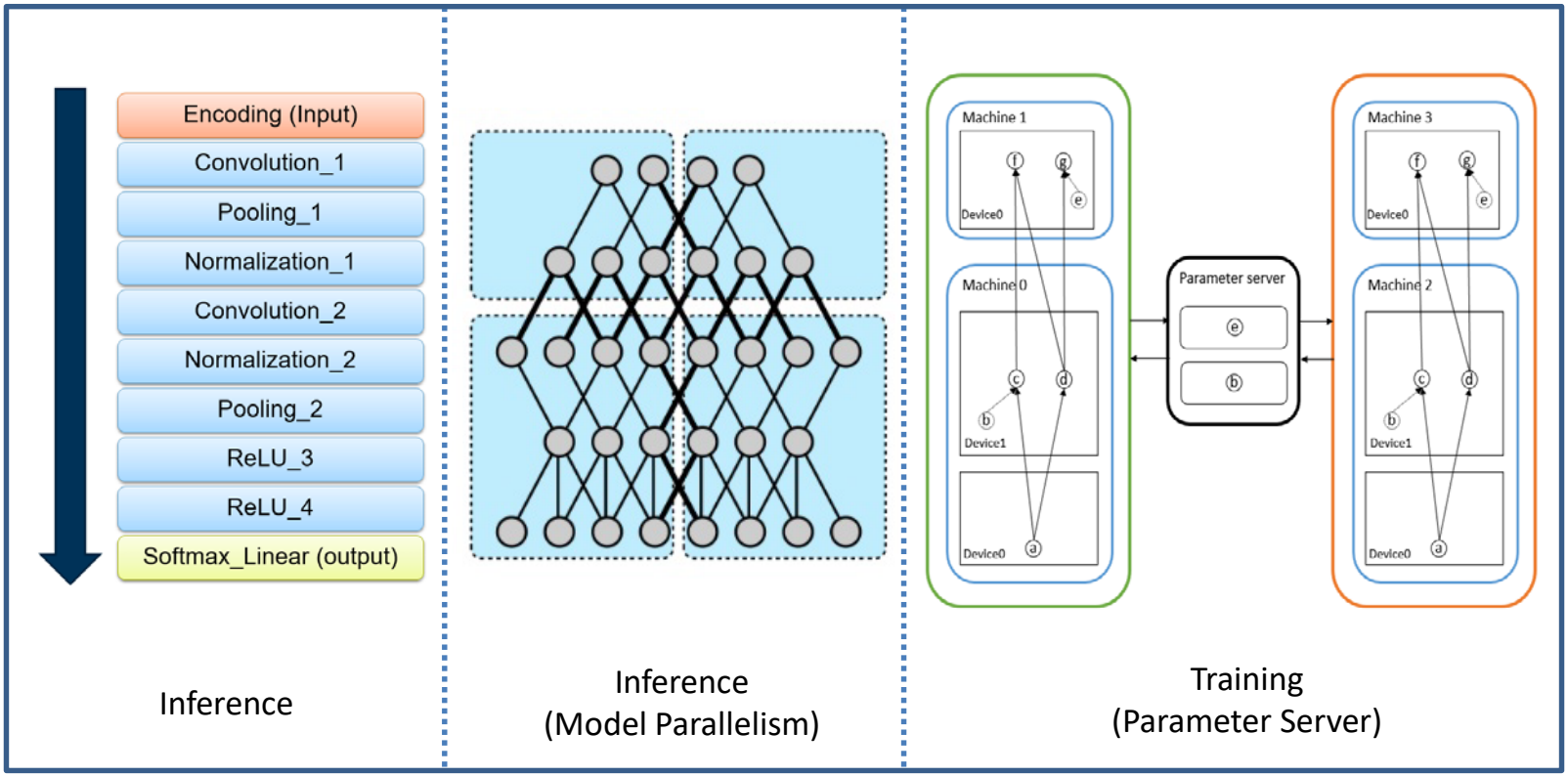
- 第二步：扩展TensorFlow使其具备可移植性

  构建一个可扩展的编程基础设施(infrastructure)

整体方案

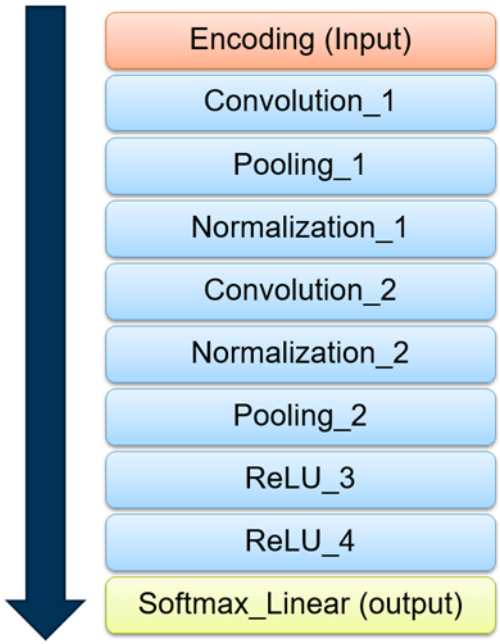

面向效率：加速器资源编排优化
面向应用：扩展TensorFlow的数据流表达机制
数据流图表达：不同粒度逻辑算子
流图映射：逻辑算子→中间算子
面向硬件：TensorFlow框架中扩展对FPGA的支持
数据流执行器：
FPGA执行模型、数据传输
Kernel层：
中间算子→FPGA物理算子
设备层：
存储管理、FPGA调用机制
Python client
C++ client
Java client
C-API
Dataflow Operator
Distributed master
Dataflow executor
Kernels
RPC
RDMA
...
Network layer
CPU
GPU
Device layer
FPGA

关键技术1--数据流图的表达



从应用中提取的几种代表性的数据流图

关键技术1--数据流图的表达

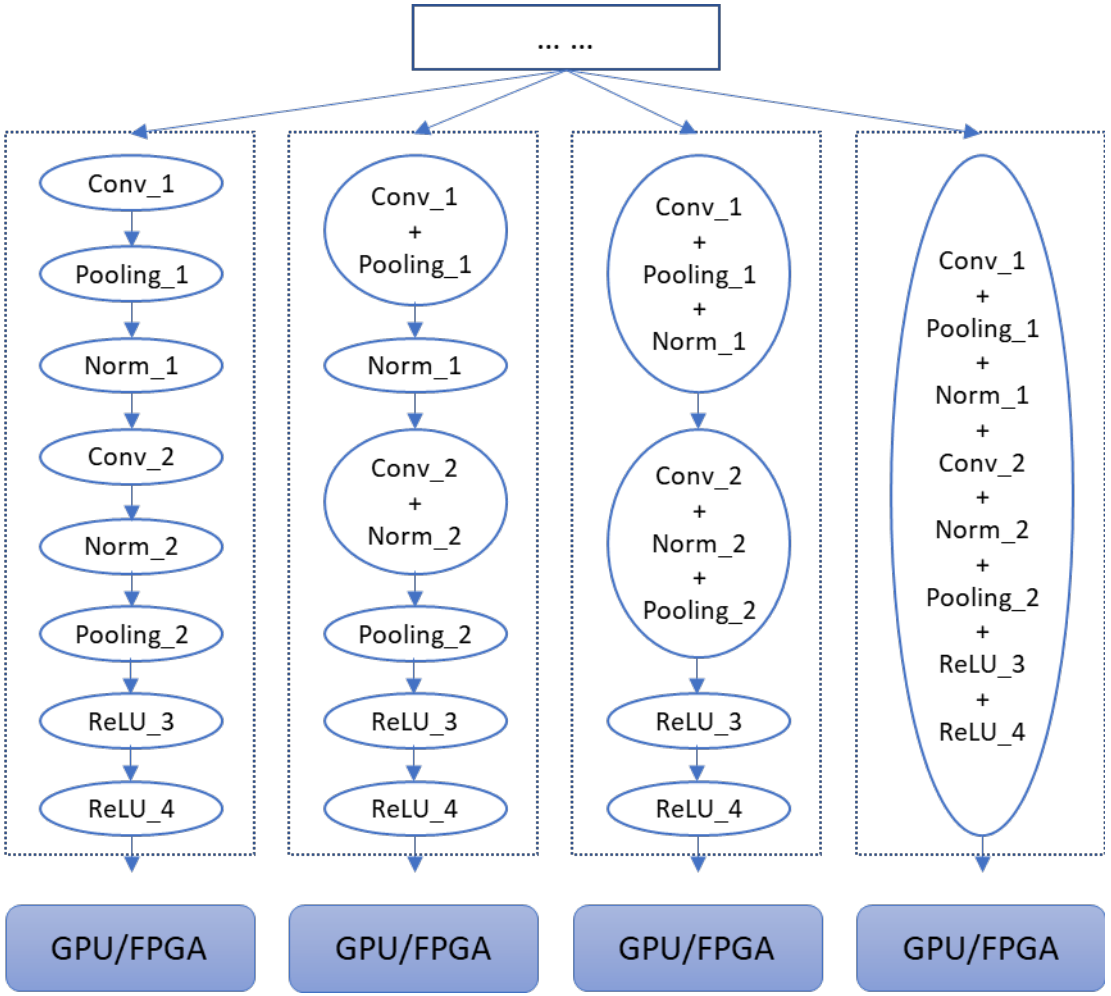


逻辑算子的粒度：

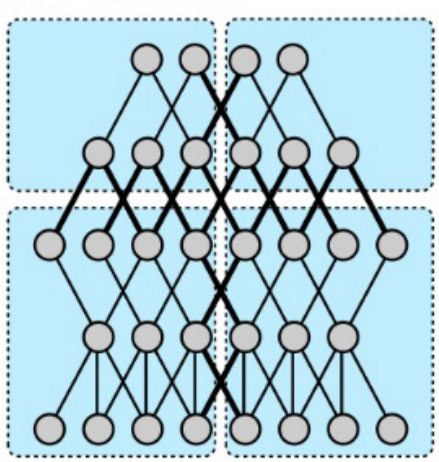
单层 – 网络中的一层

多层 – 网络中的子图

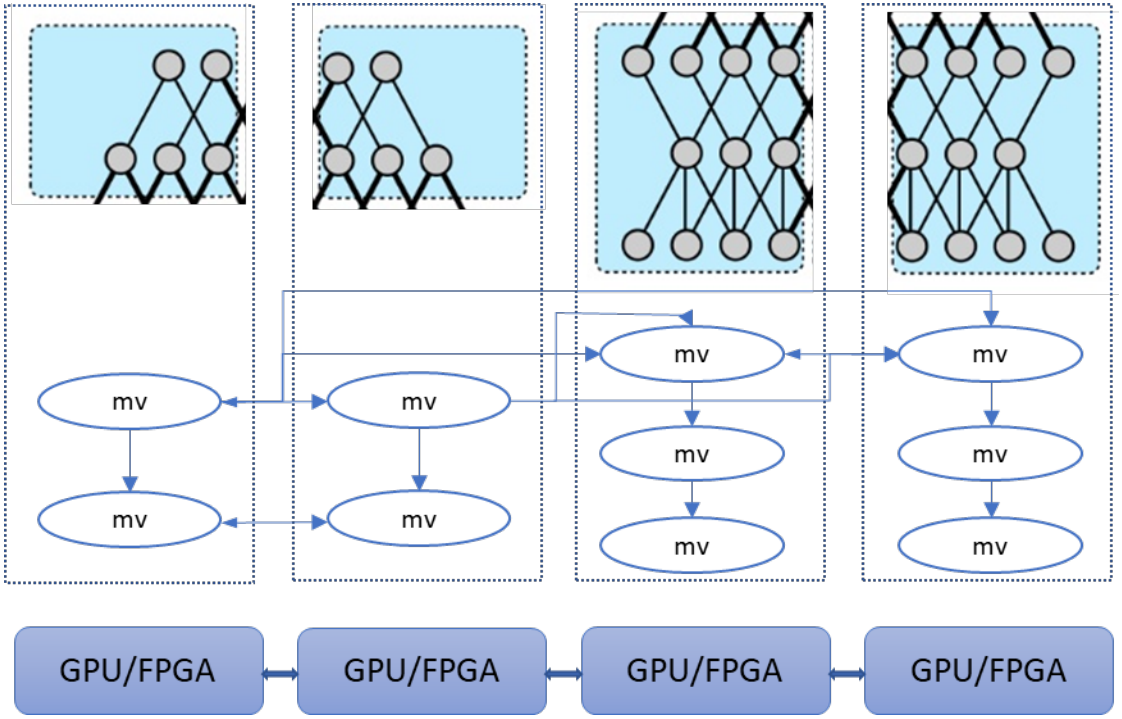
整个网络 – 最粗粒度算子



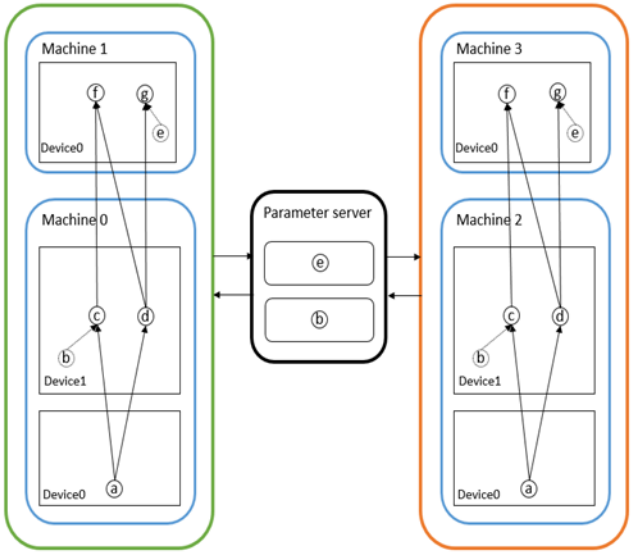
关键技术1--数据流图的表达

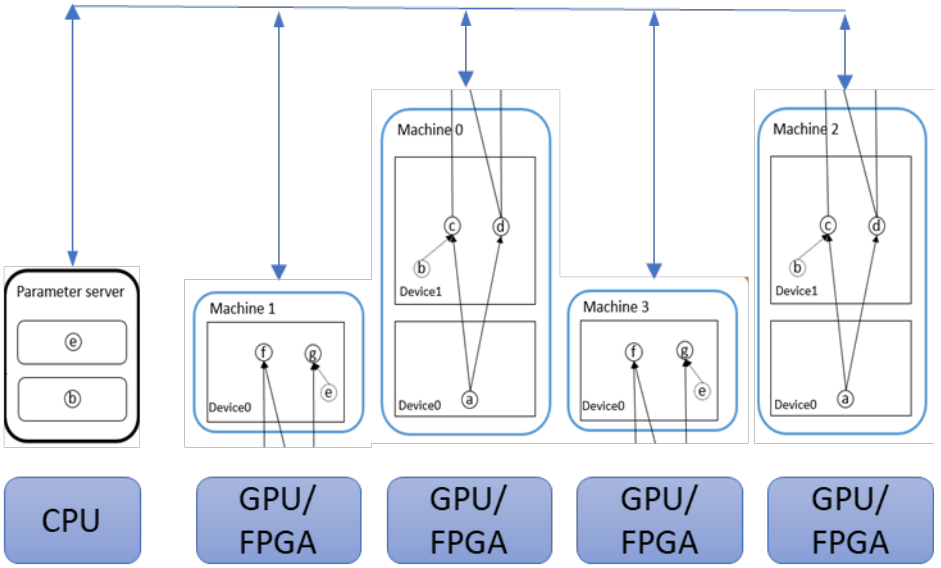


逻辑算子的粒度：
神经元 – 最细的粒度
多个神经元 – 向量操作



关键技术1--数据流图的表达





为了支持分布式训练过程，提供了参数服务器相关的逻辑算子，包括参数的上传、下发、更新。

关键技术2--TensorFlow对FPGA扩展

软件系统

神经网络模型
描述DSL

TensorFlow™

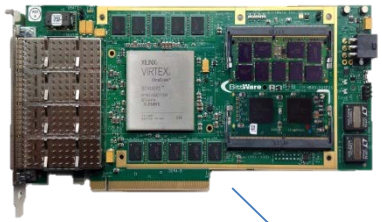
问题1：如何在高层软件中对FPGA功能单元进行抽象？

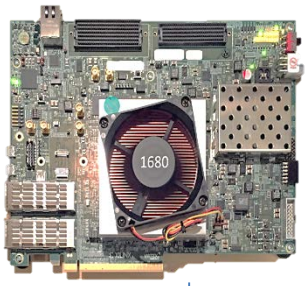
Operating System

Linux

硬件系统

PCIe-attached FPGA Card





…

RTL或HLS对硬件功能进行描述

问题2：如何解决硬件算子及平台多样性的影响？

FaaS框架及标准建议

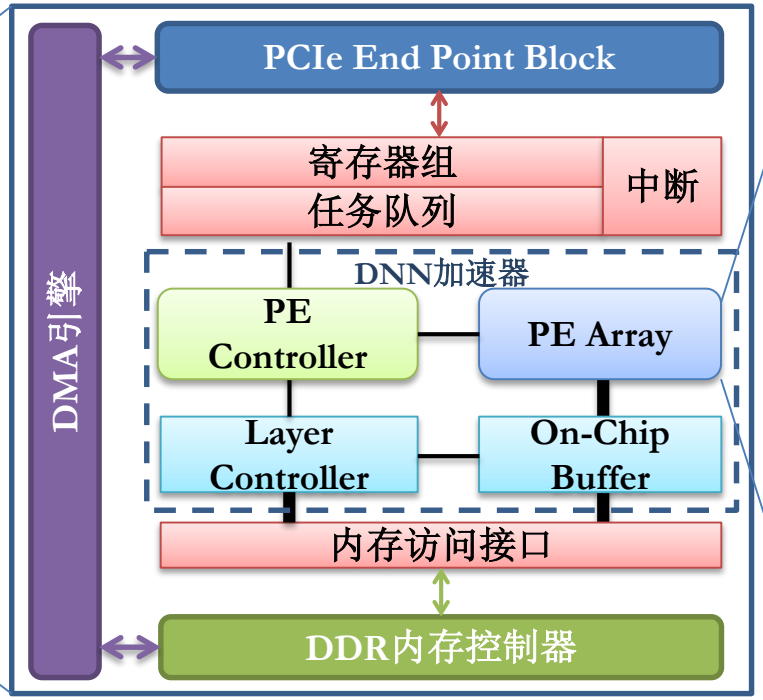
- 抽象不同硬件算子的通用功能
- 隐藏不同厂商芯片及板卡的差异
- 为应用/框架提供统一的通用API接口

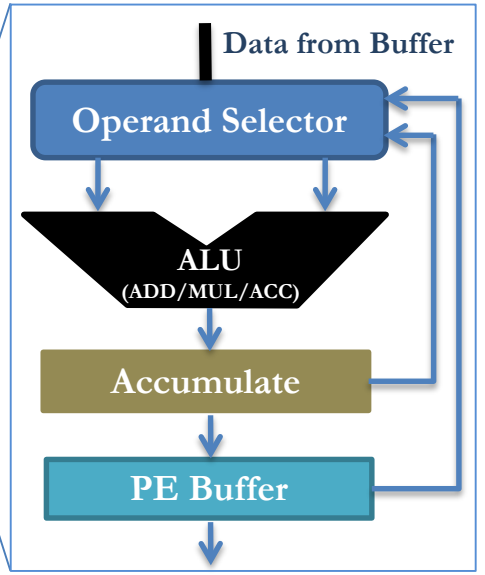
中科院计算所联合国内外多家企业共同讨论并起草

FaaS标准建议草案

FaaS框架实现实例

基于x86商用服务器及
自研FPGA板卡实现

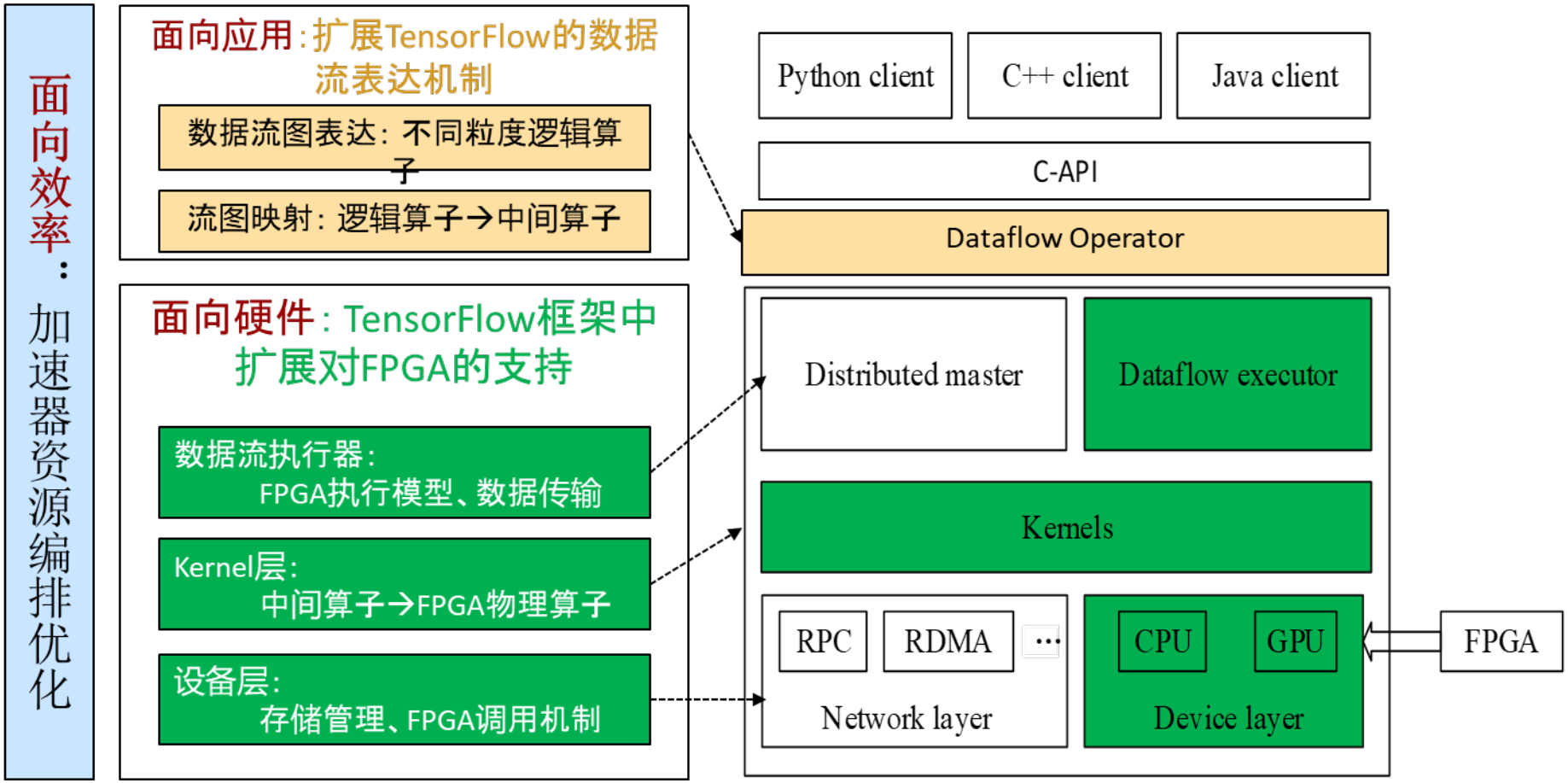




PE单元结构

- ✓ 符合FaaS标准(草案版本)的完整软硬件框架
- ✓ 实现支持多种模型的深度神经网络加速器
- ✓ 与TensoFlow、Caffe运行时框架集成

整体方案

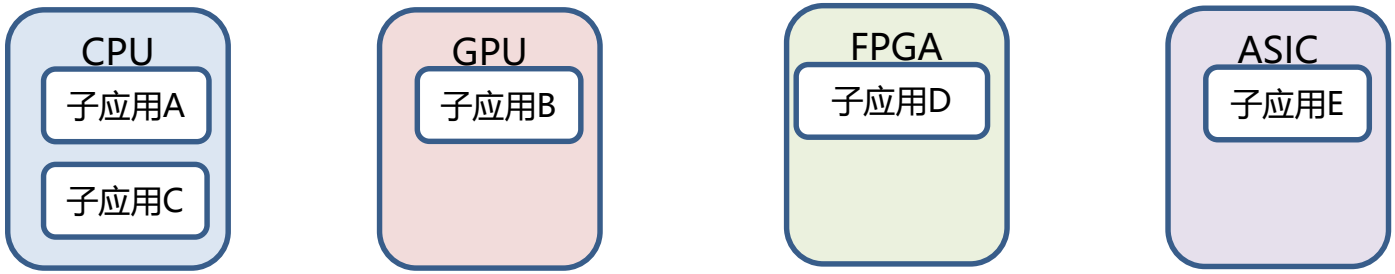

面向效率：加速器资源编排优化
面向应用：扩展TensorFlow的数据流表达机制
数据流图表达：不同粒度逻辑算子
流图映射：逻辑算子→中间算子
面向硬件：TensorFlow框架中扩展对FPGA的支持
数据流执行器：
FPGA执行模型、数据传输
Kernel层：
中间算子→FPGA物理算子
设备层：
存储管理、FPGA调用机制
Python client
C++ client
Java client
C-API
Dataflow Operator
Distributed master
Dataflow executor
Kernels
RPC
RDMA
...
Network layer
CPU
GPU
Device layer
FPGA

关键技术3 – 异构资源利用

- 四种常用芯片 – CPU、GPU、FPGA、ASIC
  - CPU：通用性强，性能一般
  - GPU：通用性<CPU，性能高，功耗高
  - FPGA：通用+定制，性能高，功耗低
  - ASIC：算法定制，通用性最差，功耗低/性能高

关键技术3 – 异构资源利用

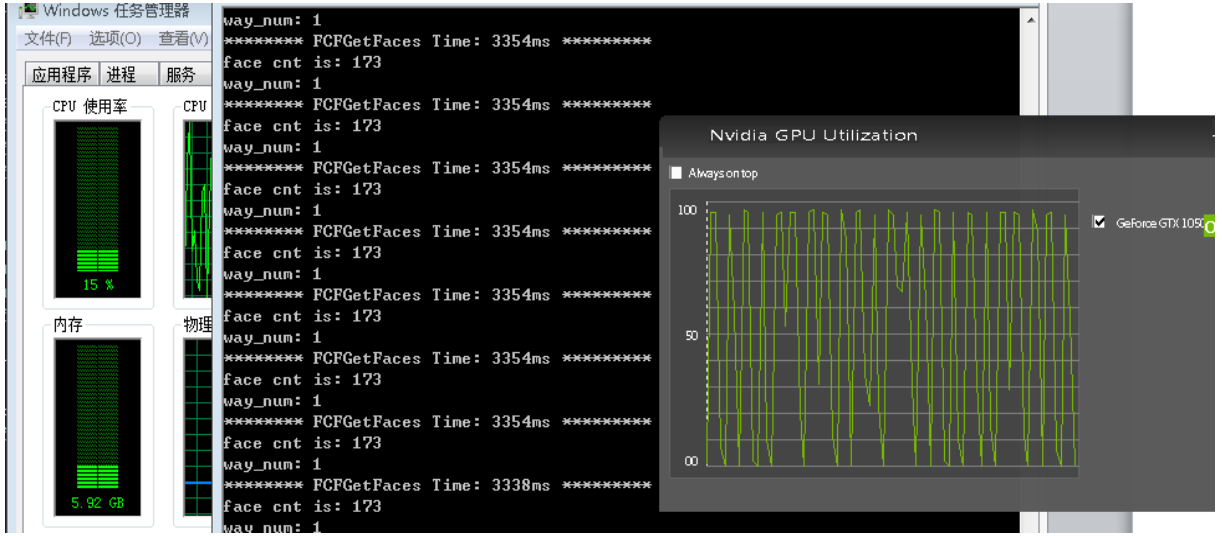
- 将应用根据计算特点进行划分为若干个子应用，各子应用进入最合适的工作单元进行处理。



- 根据应用的最优目标，充分利用各芯片进行协作。当A利用率较高而B利用率较低时用B来对A进行分担。

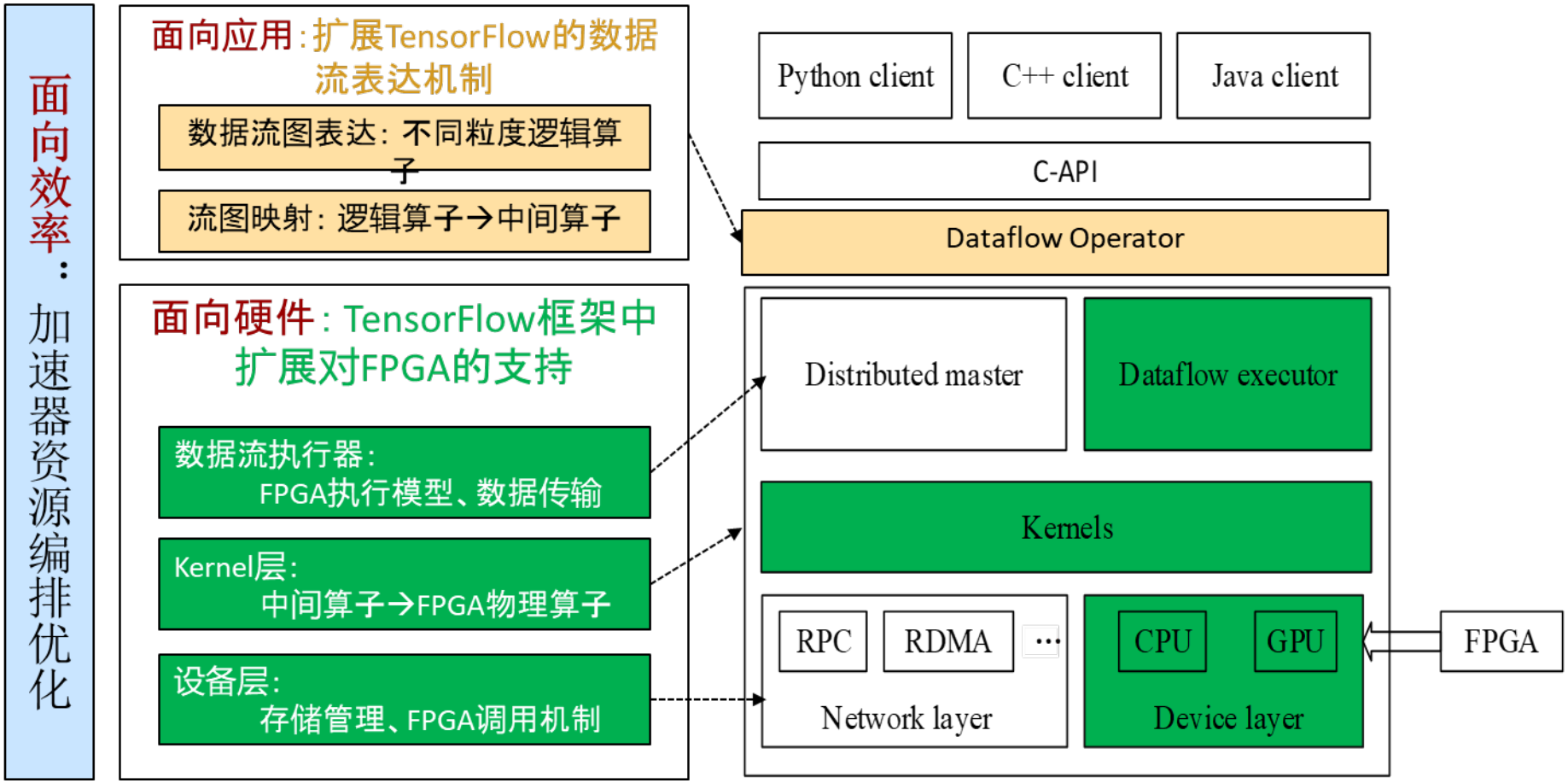
关键技术3 – 异构资源利用

- 以智能视频监控处理为例
  - 包括人脸检测、人脸对齐、人脸特征提取



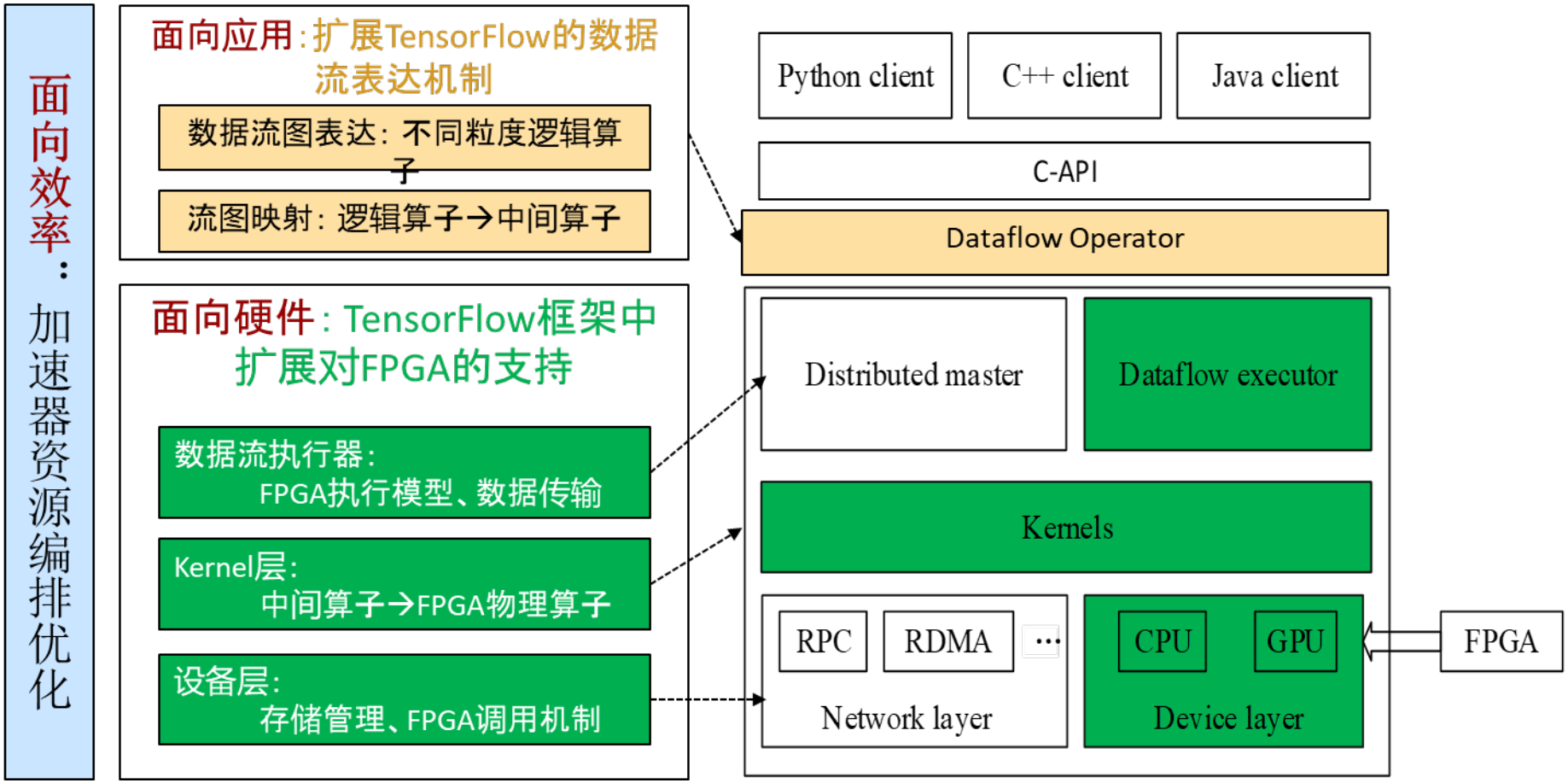
  - 优化一：异构任务部署 – 检测(GPU)、对齐(CPU)、特征提取(GPU)
  - 优化二：GPU上任务进行缓存，增大处理粒度
  - 优化效果：500ms/帧→150ms/帧

整体方案


面向效率：加速器资源编排优化
面向应用：扩展TensorFlow的数据流表达机制
数据流图表达：不同粒度逻辑算子
流图映射：逻辑算子→中间算子
面向硬件：TensorFlow框架中扩展对FPGA的支持
数据流执行器：
FPGA执行模型、数据传输
Kernel层：
中间算子→FPGA物理算子
设备层：
存储管理、FPGA调用机制
Python client
C++ client
Java client
C-API
Dataflow Operator
Distributed master
Dataflow executor
Kernels
RPC
RDMA
...
Network layer
CPU
GPU
Device layer
FPGA

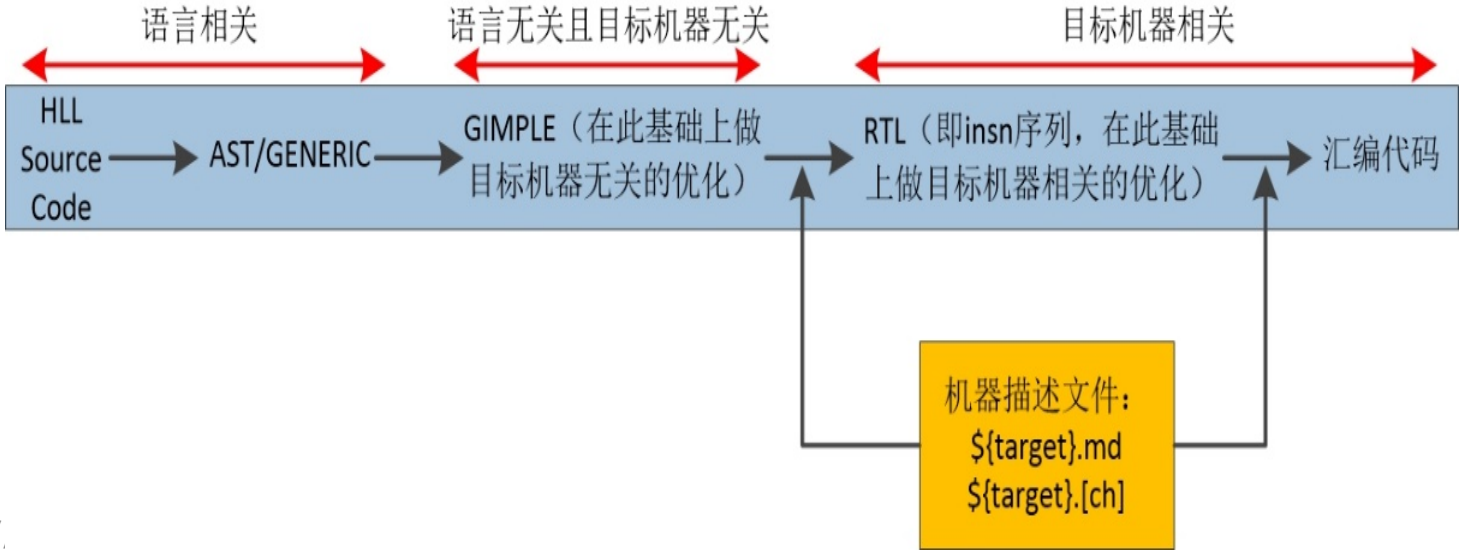
下一步：
From Programming Framework To Infrastructure

整体方案

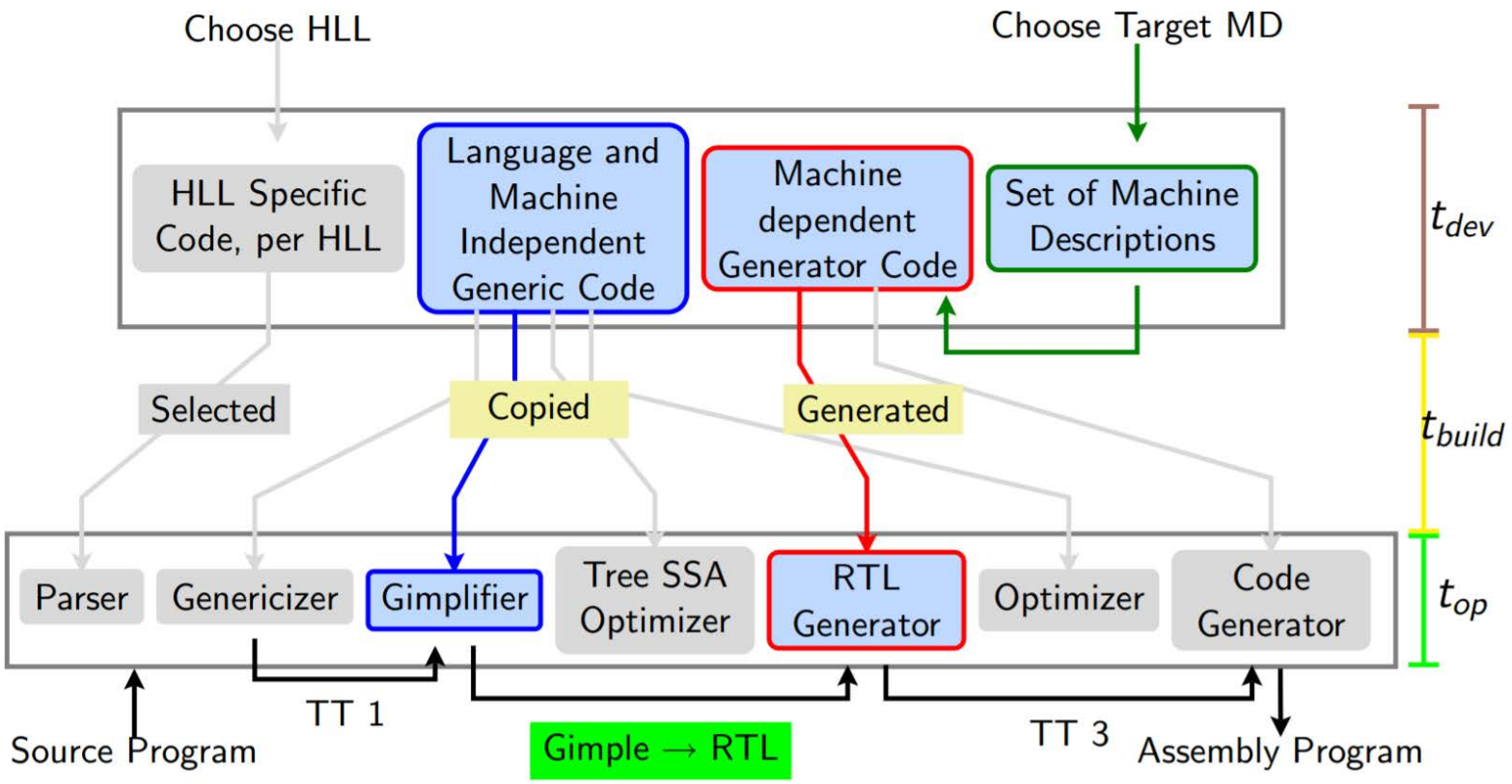

面向效率：加速器资源编排优化
面向应用：扩展TensorFlow的数据流表达机制
数据流图表达：不同粒度逻辑算子
流图映射：逻辑算子→中间算子
面向硬件：TensorFlow框架中扩展对FPGA的支持
数据流执行器：
FPGA执行模型、数据传输
Kernel层：
中间算子→FPGA物理算子
设备层：
存储管理、FPGA调用机制
Python client
C++ client
Java client
C-API
Dataflow Operator
Distributed master
Dataflow executor
Kernels
RPC
RDMA
...
Network layer
CPU
GPU
Device layer
FPGA

可重定向 – Learnt from compilers

- 多种前端语言适配多种后端机器
  - 灵活适配与扩展，语言（机器）改变不影响机器（语言）
  - 机器描述抽象目标机器
  - IR抽象程序，并与语言无关



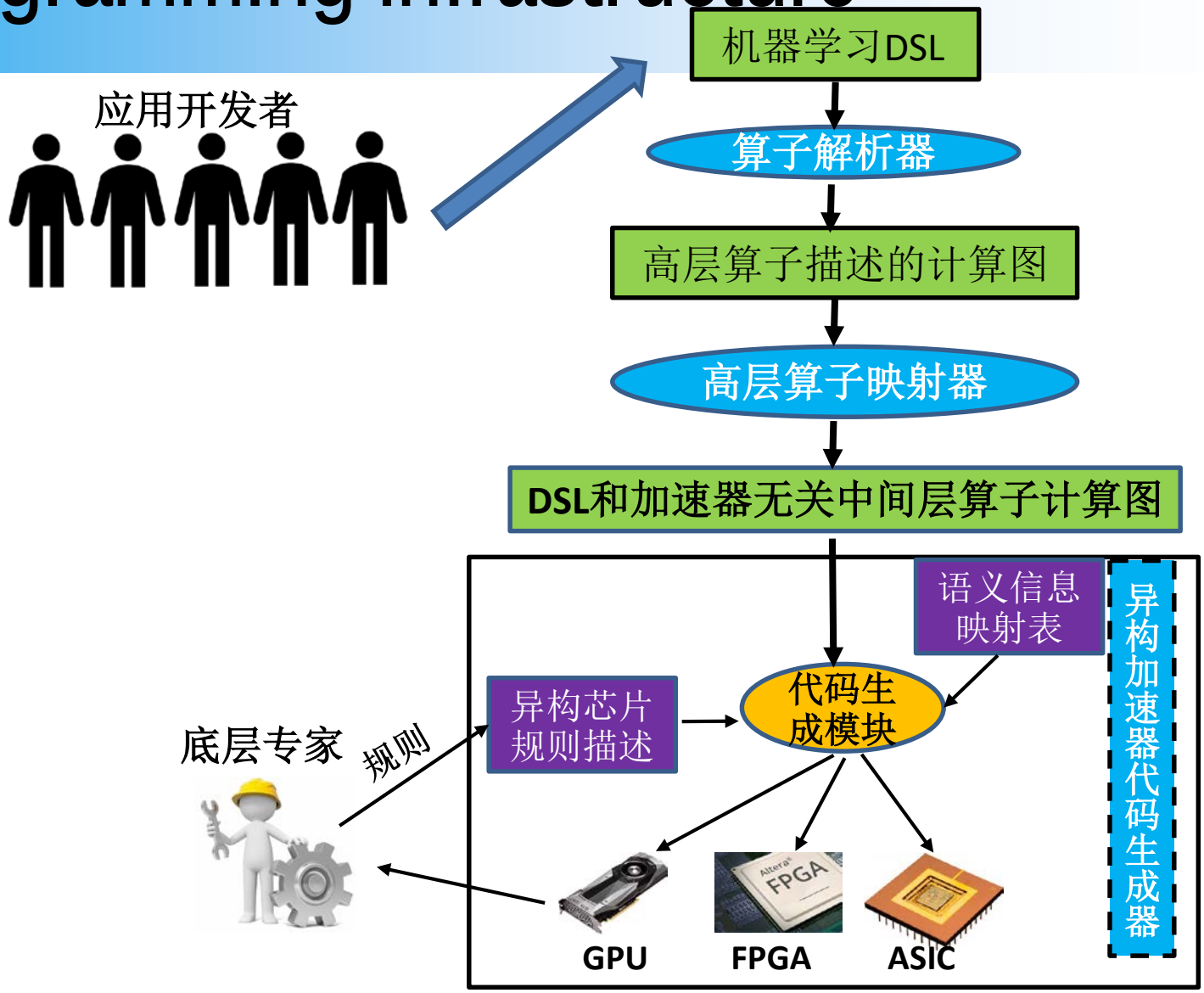
可重定向 – Learnt from compilers



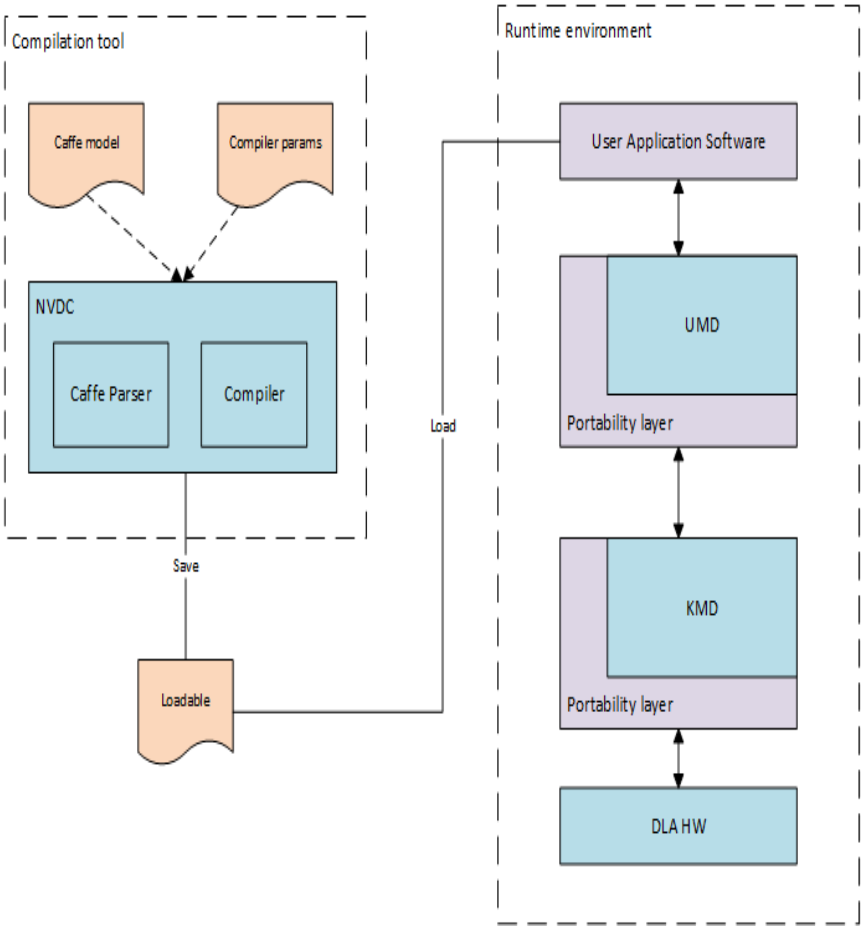
无需改动编译器

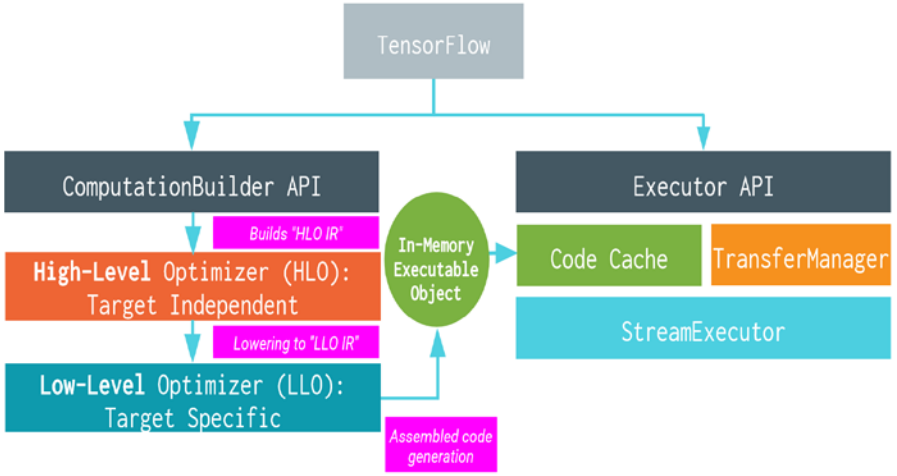
无需手工实现程序到底层硬件的映射

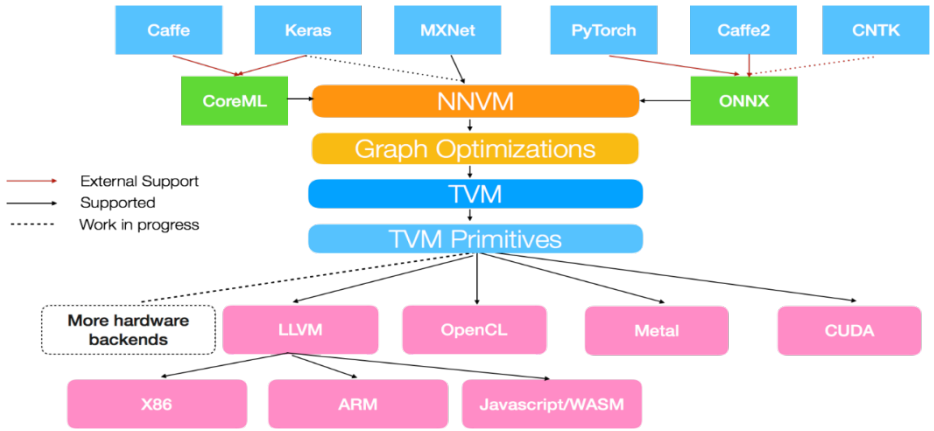
AI Programming Infrastructure



近期热点


Compilation tool
Caffe model
Compiler params
NVDC
Caffe Parser
Compiler
Save
Loadable
Load
Runtime environment
User Application Software
UMD
Portability layer
KMD
Portability layer
DLA HW


TensorFlow
ComputationBuilder API
Builds "HLO IR"
High-Level Optimizer (HLO): Target Independent
Lowering to "LLO IR"
Low-Level Optimizer (LLO): Target Specific
Assembled code generation
In-Memory Executable Object
Executor API
Code Cache
TransferManager
StreamExecutor


Caffe
Keras
MXNet
PyTorch
Caffe2
CNTK
CoreML
NNVM
ONNX
Graph Optimizations
TVM
TVM Primitives
External Support
Supported
Work in progress
More hardware backends
LLVM
OpenCL
Metal
CUDA
X86
ARM
Javascript/WASM

So far, we have

- 一个AI Programming Infrastructure
  - 运行于数据中心
  - 可以支持多种数据流图
  - 可以适配多种加速芯片

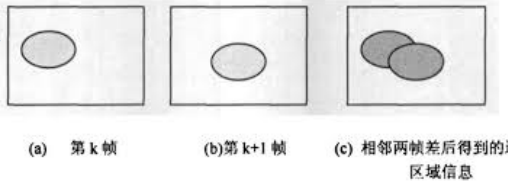
- Go one step further
  - 端云融合

端云融合编程


APK
模式1：应用运行于端设备
模式2：UI运行于端设备
处理发送到数据中心完成
08:08
模式4：应用部分运行于端
设备，部分运行于数据中心
APK
4-1:
缺省
运行
于端，
切片
上载
数据
中心
4-2:
缺省
运行
数据
中心，
切片
下载
到端
模式3：用户使用云ML
平台进行AI计算

代表性场景 - 平安城市

快速物体检测算法在
端设备初步实施筛选



端完成深度学习
前X层特征提取

[2.41, 0.87]
[7,92, 0.87]

数据中心完成深
度学习后续层次

端无法处理的帧
传输到数据中心

Thanks

http://www.carch.ac.cn/~huimin/

cuihm@ict.ac.cn