基于开放标准OpenCL的深度学习研究和探索

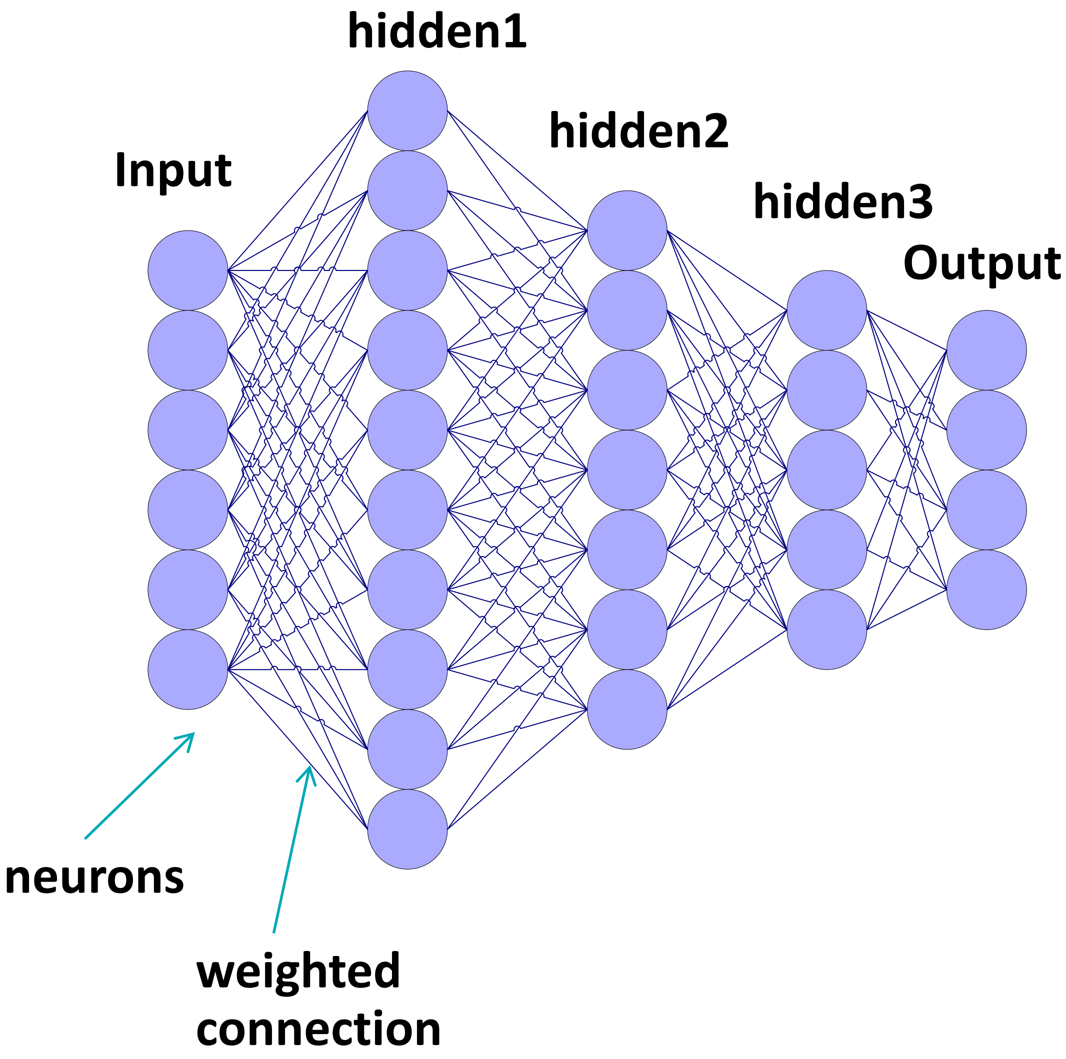
谷俊丽
AMD Research
Collaborated with product team
Junli.Gu@AMD.com

Outline

- 深度学习及其发展状况
- 深度学习对系统实现的挑战
- 基于OpenCL的深度学习探索

DNN 模型

- What is a **D**eep **N**eural **N**etwork (**DNN**)?
  - 3~24 hidden layers, millions to billions of parameters
  - DNN + Big Data is leading recent direction in machine learning
- Rich Varieties of DNN Structures
  - **MLP** (Multi-level Perceptron)/ **AutoEncoder**
  - **CNN** (Convolutional Neural Network)
  - **DBN** (Deep belief network)/**RBM** (Restricted Boltzmann Machine)
- **Deep Learning** on DNN model
  - Random initialized parameters
  - Trained to converge by feeding large scale of data (Big Data)

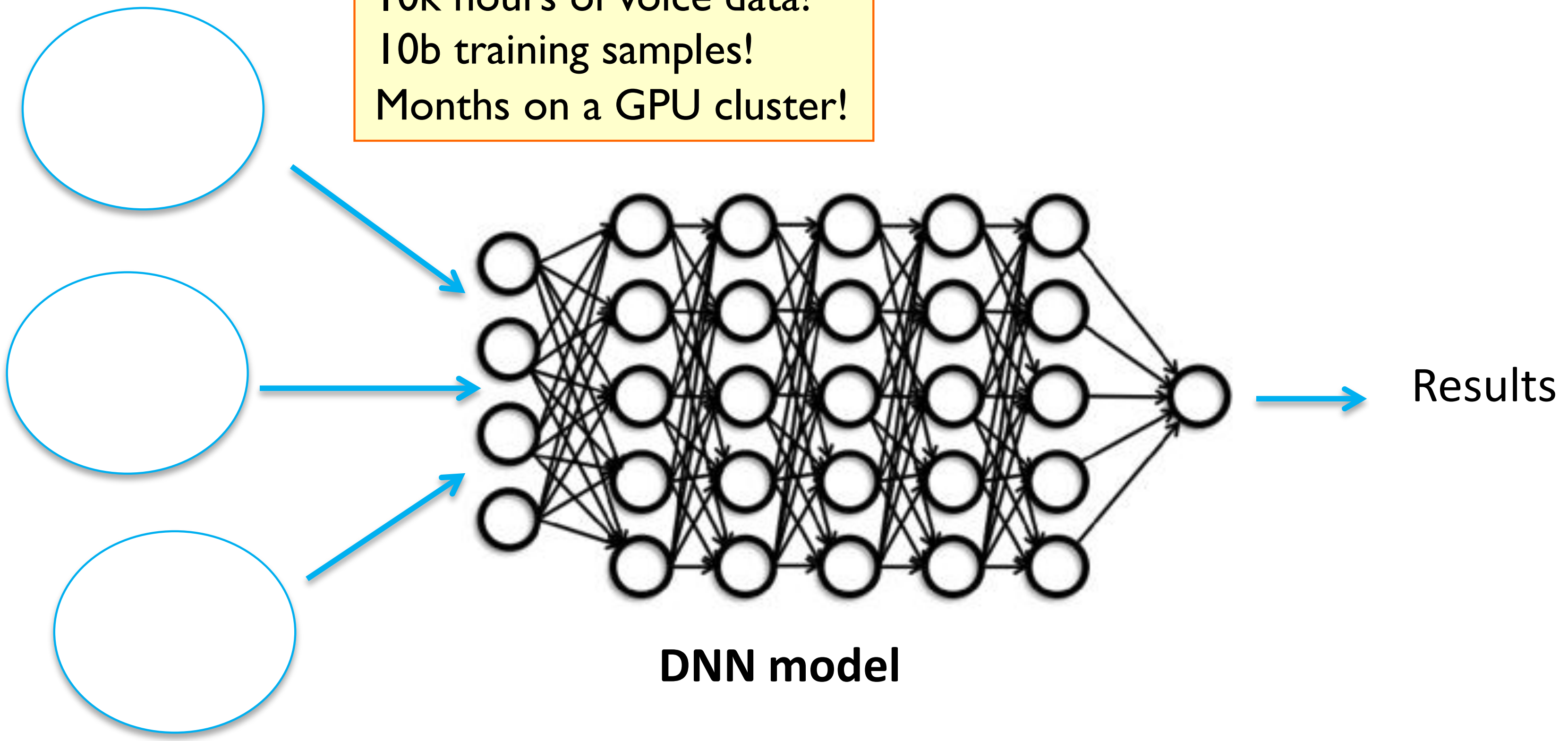


Starting to get really hot after winning 2012 ILSVRC competition

深度学习过程 (DEEP LEARNING)

- Deep Learning: DNN model + Big Data
- Actually human defined features no longer work well for Big Data scenarios with noise.
- All features learnt by training data, without human interference.

DNN for Speech!
10k hours of voice data!
10b training samples!
Months on a GPU cluster!

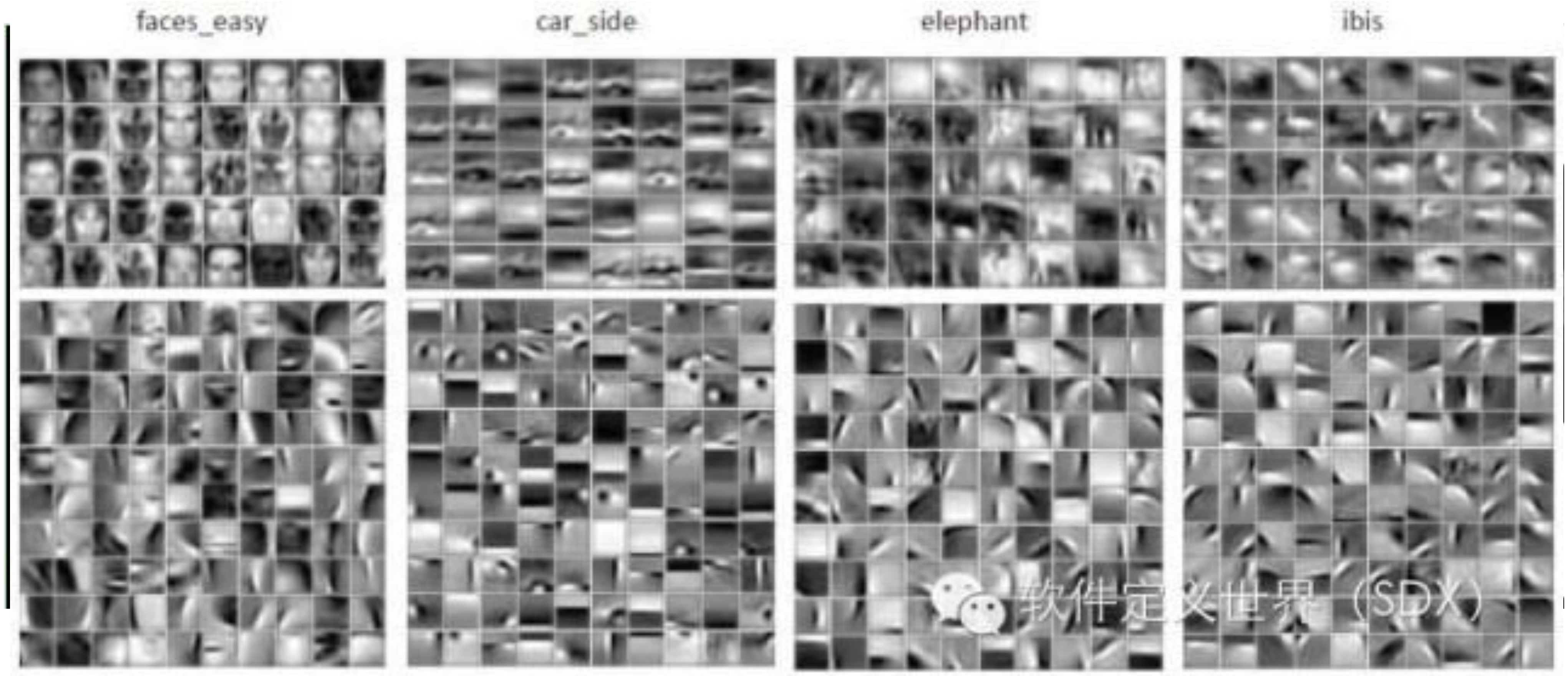


DNN model

深度学习为何强大？

HIERARCHICAL FEATURE EXTRACTION

- Extract features layer by layer from input data, to form hierarchical representation that is **beyond human's definition**
- **Features have semantic meanings**



深度学习正在引领潮流

- Why **internet companies** purse DNN these days?
  - Original human defined algorithms don’t work well for Big Data
  - Competing in machine learning to understand Big Data
- DNN (deep neural networks) is breaking through & leading direction
  - Large scale of image classification/recognition/search, face recognition
  - Online recommendation for electronic business
  - Voice recognition, music search etc.
  - **Eg. Image classification accuracy: 74% in 2011, 93% till today**
- Long-term investment by industry
  - BAT, Google, Facebook, Yahoo, Microsoft, Bank and Finance
  - Google/Baidu/IBM Brain project

DNN + Big Data is believed to be the evolutionary trend for apps & systems.

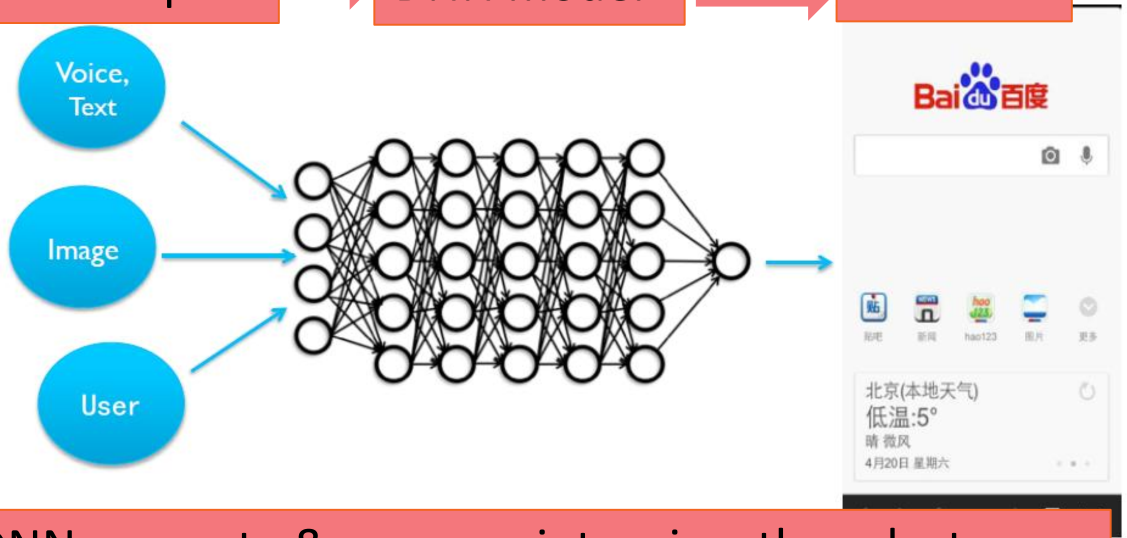
应用示例：以图搜图

深度学习对系统设计的挑战

AMD

- Typical scale of data set
  - Image search: 1M
  - OCR: 100M
  - Speech: 10B, CTR: 100B
- **Projected data to growth 10X per year**
- **DNN model training time**
  - Weeks to months on GPU clusters
  - Trained DNNs then deployed on cloud
- **System is the final enabler**
  - Current platform runs into bottleneck
    - CPU clusters → CPU + GPU clusters
  - Looking at dGPUs, APUs, FPGAs, ASIC, etc.

Big Data input → DNN model → Answer



DNN compute & memory intensive, thus clusters

深度学习将无所不在

- **Deep Learning is applied to tremendous application scenarios and various device platforms**
- Deep learning system should consider
  - Cross platform compatibility, portability
  - People want the same code to run on different platforms

Supercomputers!

Offline training

Datacenters!

Deployed on cloud

Tablets, smartphones!

Online on mobiles

Wearable devices! IoTs!

wearables and IoTs

Credit to Baidu Ren Wu

OPENCL开放标准

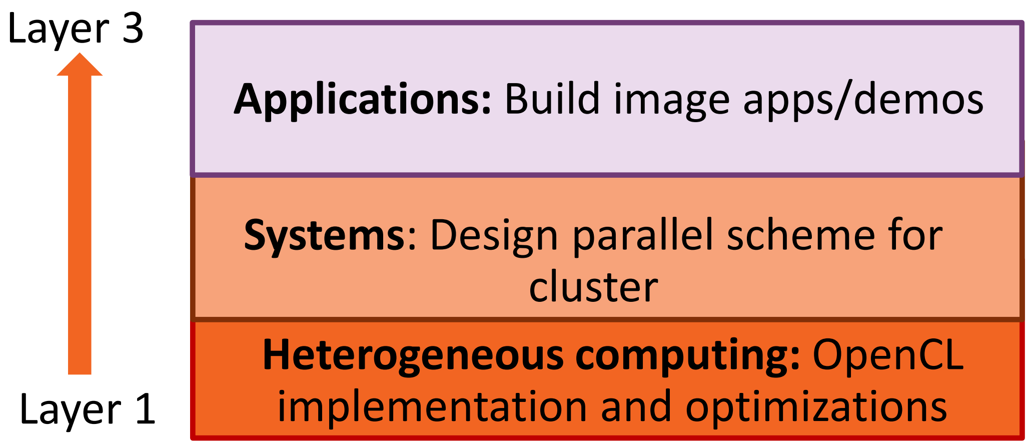
- OpenCL is industry’s open standard for heterogeneous computing
  - Support cross platform compatibility, portability
- Broad support from different companies
- We believe deep learning system should be built based on OpenCL
  - One version of codes, you can run on CPU, GPU, APU, accelerators from all vendors

AMD DNN: 基于OPENCL的深度学习实现

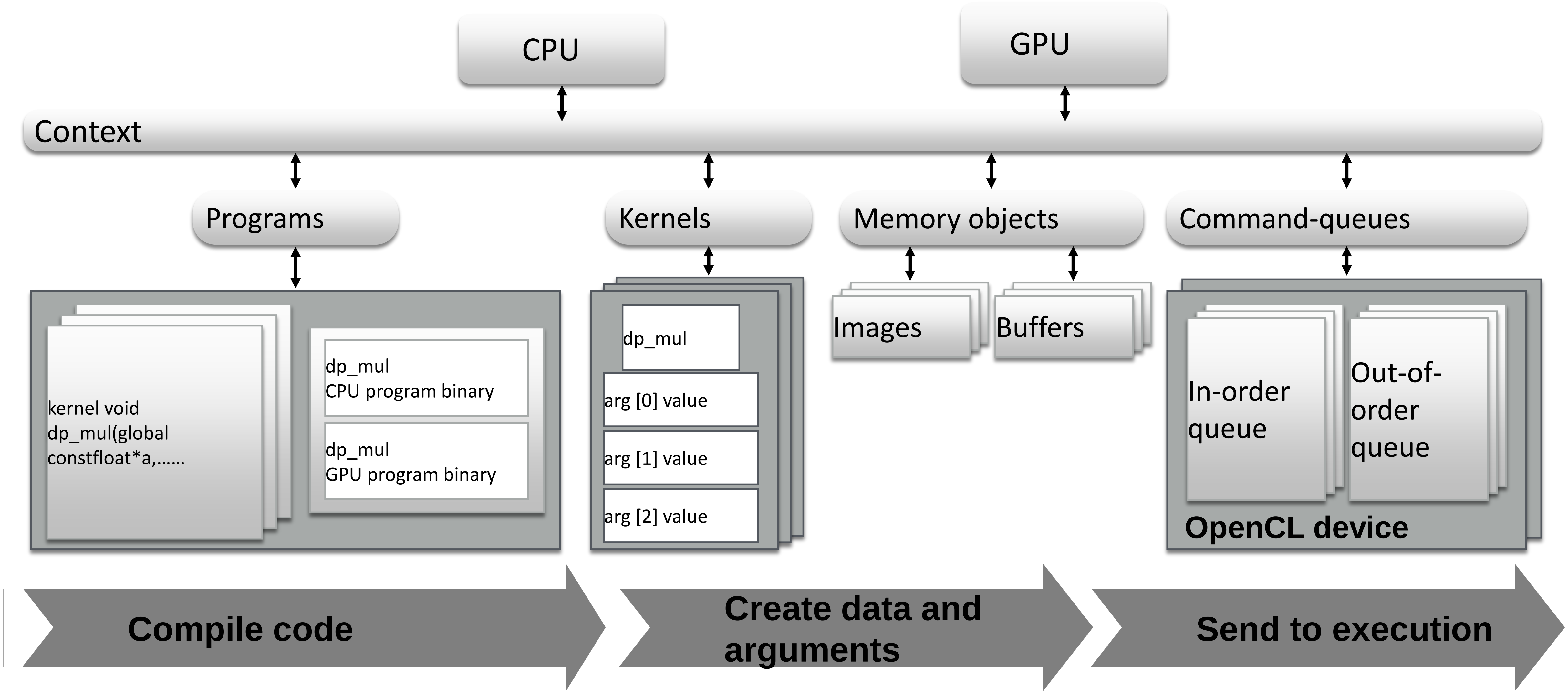
AMD

- **Project Goal: tackle DNN challenges from H/W to System to Applications**

- **Layer1 H/W: Heterogeneous platform implementation and speedup**
  - **OpenCL implementation and performance optimizations**
- **Layer2 Systems: Scale out to distributed systems**
- **Layer 3 App.: DNN + Big Data applications**
  –



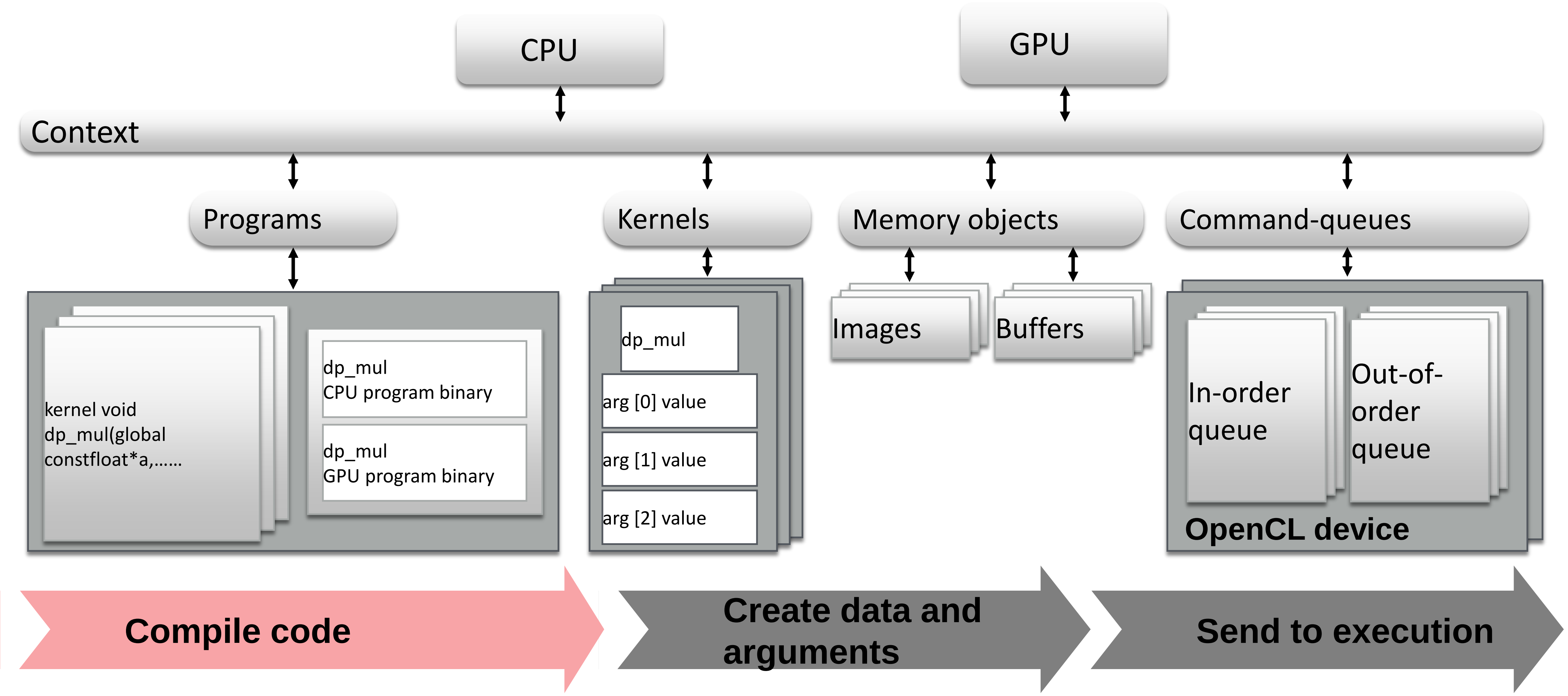
OPENCL 实现详细



CPU
GPU
Context
Programs
Kernels
Memory objects
Command-queues
kernel void
dp_mul(global
constfloat*a,……
dp_mul
CPU program binary
dp_mul
GPU program binary
dp_mul
arg [0] value
arg [1] value
arg [2] value
Images
Buffers
In-order queue
Out-of-order queue
OpenCL device
Compile code
Create data and arguments
Send to execution

OPENCL 实现的挑战

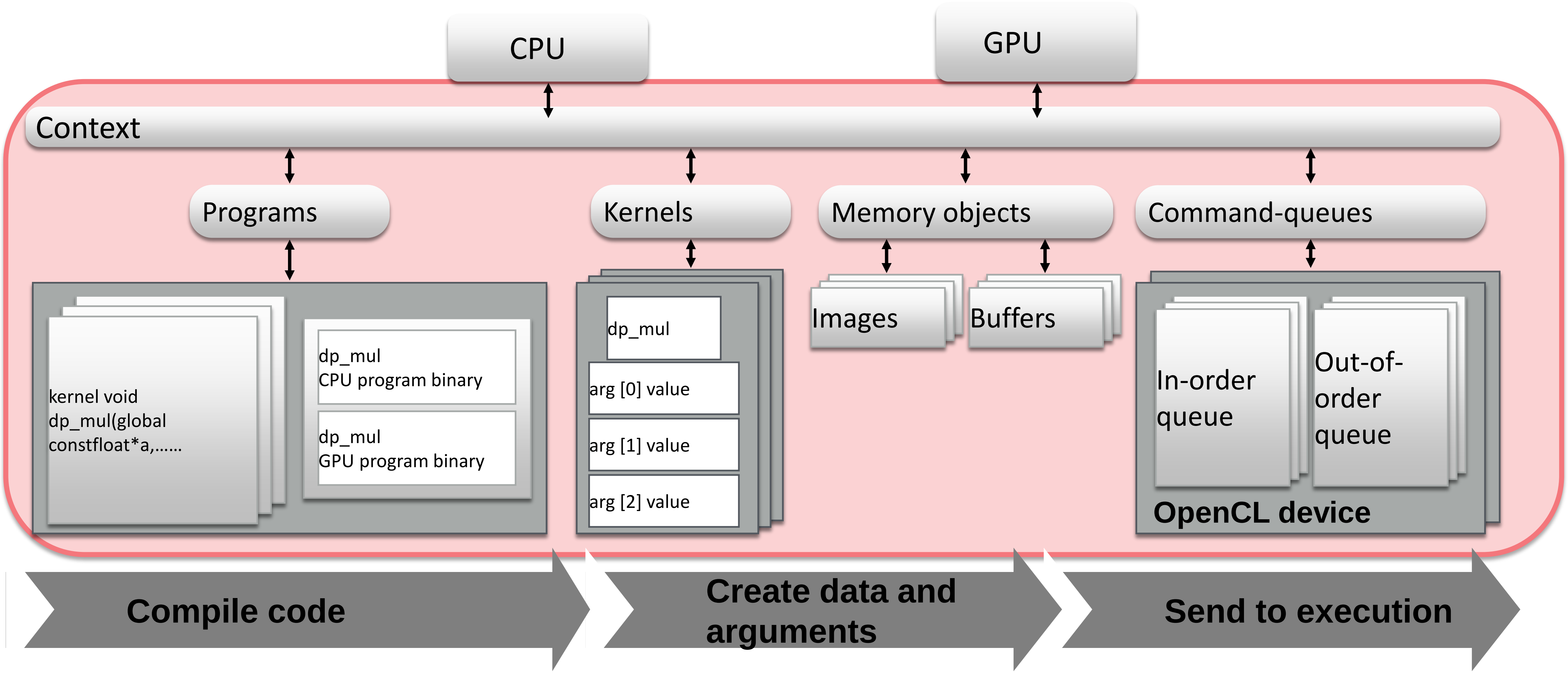
AMD



OpenCL uses runtime compiling: To allow various H/W devices and optimize kernels accordingly

Tradeoff: runtime compiling takes computation time

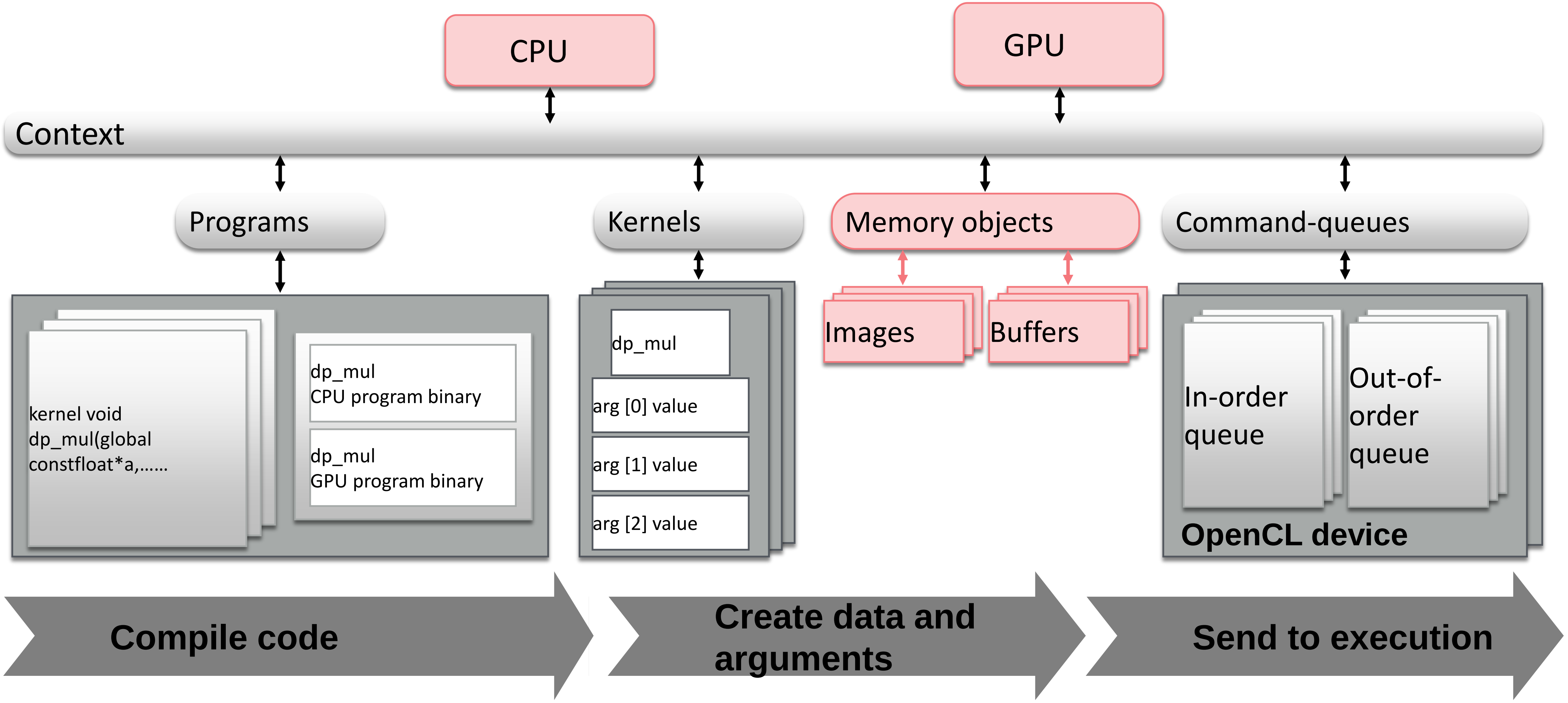
OPENCL 实现的挑战



Heavy OpenCL H/W details

Domain expert usually don't appreciated hardware details (devices, cache, memory, etc.)

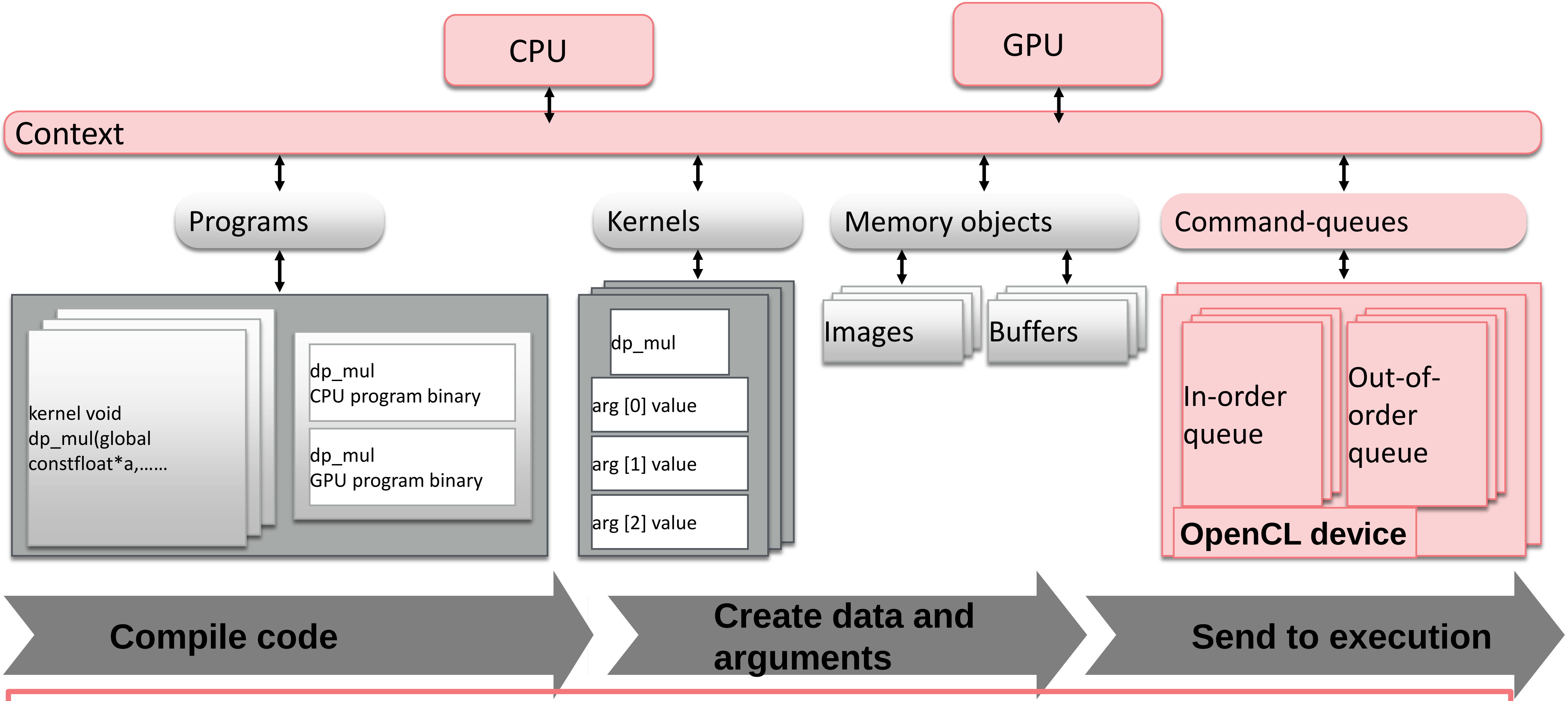
OPENCL 实现的挑战



Memory layout and data coherence

we have both CPU and GPU mem objects. How to maintain the coherence?

OPENCL实现的挑战



Task and device synchronization

Critical when you have multi-tasks and multi-devices

STRATEGIES AND SOLUTIONS WE USE

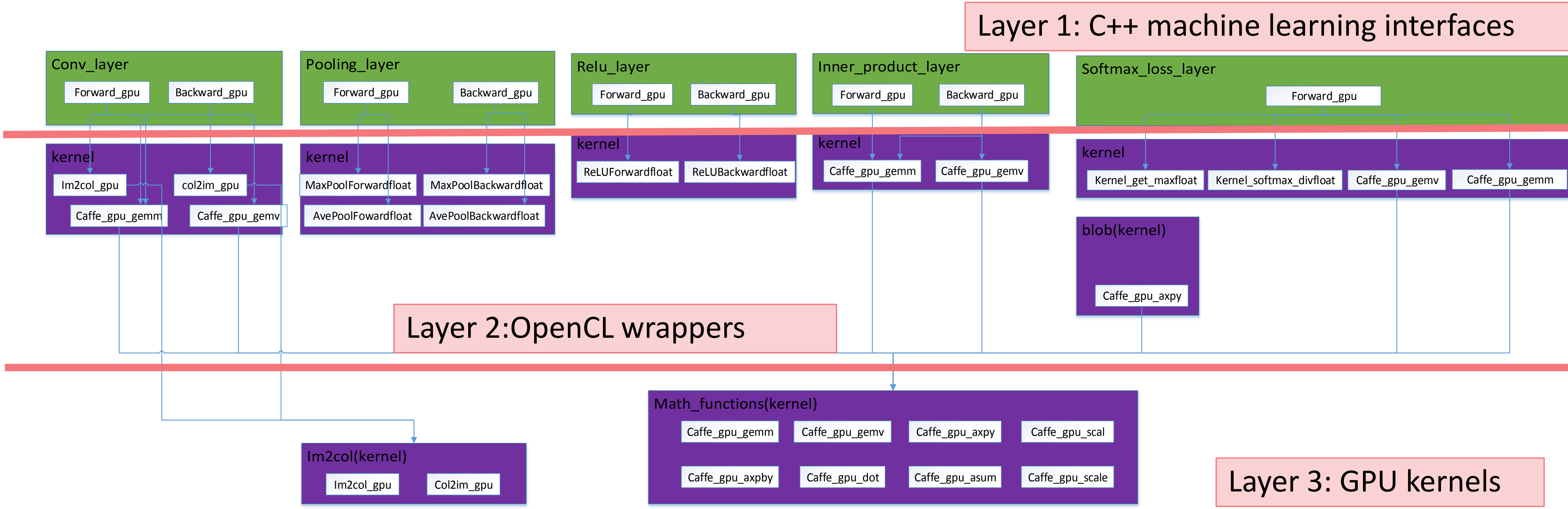
- **OpenCL runtime compilation**
  - We solved this by optimizing both library and OpenCL runtime
- **H/W wrap-up layer**
  - Deals with H/W details and optimizations
- **Data coherence between CPU and GPU**
  - We designed S/W level coherence protocols; Hopefully HSA's features will enable more effective solution
- **Task and device synchronization**
  - We designed synchronization protocols using context, command queues and events

OPENCL DNN HIERARCHY DESIGN

AMD

- Layer1: C++ interfaces (for domain experts)
- Layer2: OpenCL wrapper hides hardware details (for systems)
- Layer3: Underlying GPU kernels (for deep optimizations)

- GPU kernels
  - Hand coded kernels
  - OpenCL APIs

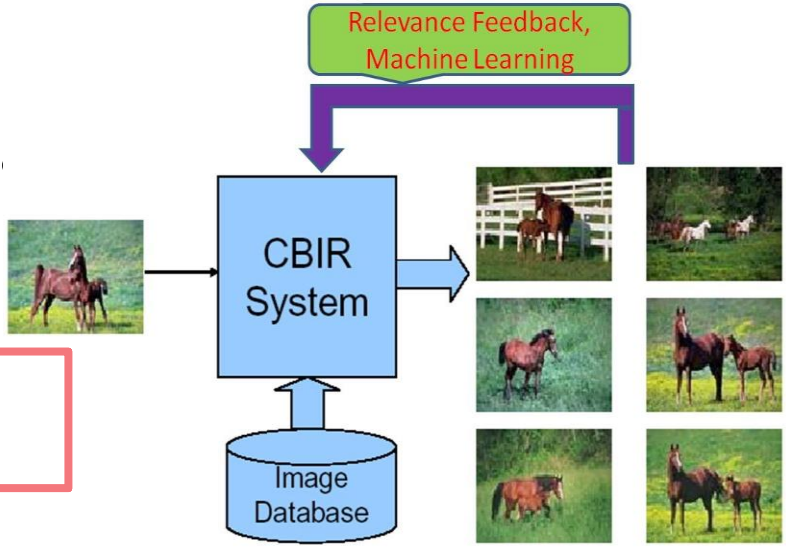


搭建深度学习的大数据应用场景

- Classification and recognition based on MLP model
  - Optical Character Recognition (OCR)
  - Driver license plate recognition
  - Voice recognition with industry scale of data
- Image/object classification based on CNN
  - Small images are done, next scaling to industry size images

Large scale object recognition

- Content based image retrieval (CBIR)
  - Retrieve images that are similar in content to the query imag
  - Model used: Autoencoder + RBM



Using our kernels your application is able to run on CPU/GPU/APU/accelerators etc.

AMD之深度学习解决方案

- **H/W solutions: Parallel implementation on systems and system level evaluation**
  - **CPU + GPUs cluster**
  - **APU server**

- **S/W solutions: OpenCL solution of deep learning applications**
  - **Applicable to general heterogeneous platforms**

- **Set up real world application scenarios with external company’s involvement and apply AMD solutions to industry**

Note: Collaboration from both academia and industry is welcomed

人工智能与系统相结合：机遇与挑战并存

AMD

机遇

- 人工智能的新浪潮将引领未来20年的技术和系统革命，这个浪潮首先在互联网公司掀起，正在如火如荼的进行研究。
- **光有算法是解决不了最终问题的，硬件系统是大数据+算法的enabler**。硬件领域也需要抓住此时机，回答硬件系统如何设计具有人工智能的本领，这是系统研究人员面临的机遇。
- IBM的沃森处理器是一个好的研究成果，并且已经投入使用解决一些大数据的金融分析、实时语音翻译等应用。

挑战

- 现有的分布式系统上的实现方法，节点间需要传输大量数据和参数，通信代价太高，当节点数目超过一定数量时，不能获得持续的加速比。多个节点间训练不同数据时如何协调和同步，可能需要从算法角度重新设计。
- **分布式系统如何设计，需要DNN算法专家和系统专家共同协同解决**，解决的方法可能既要修改算法使之跟底层硬件架构匹配，又要求系统专家设计计算能力强大的单机器，又要设计高密度整合、高效通信的服务器。