2021中国智能网卡研讨会

CHINA SMARTNIC WORKSHOP

软硬件融合

——超大规模云计算架构创新之路

黄朝波

目 录

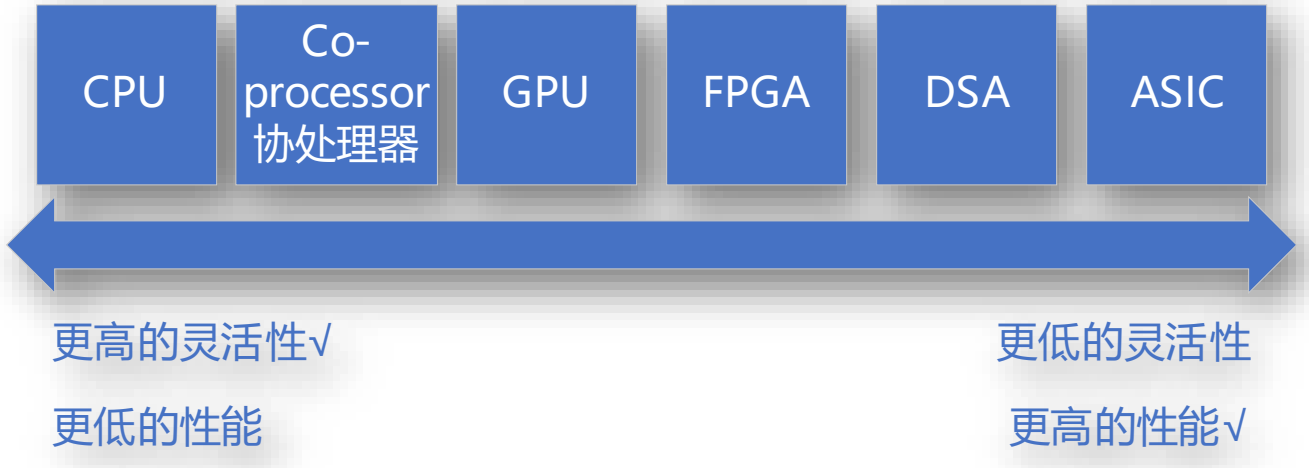
DPU/IPU，Data / Infrastructure Process Unit，数据/基础设施处理器。

0 《软硬件融合》缘起

- 软件热点层出不穷并且快速迭代；CPU性能逐渐瓶颈，摩尔定律失效；图灵奖获得者D&J给出的方案是DSA。
- 云计算是各种复杂场景的叠加，挑战在于：如何把这么多场景优化融汇到一套平台化方案里；既满足灵活性的要求，又满足性能加速的要求。
- 提出了全新的设计理念和方法——软硬件融合，期望实现软件灵活性和硬件高效性的统一。

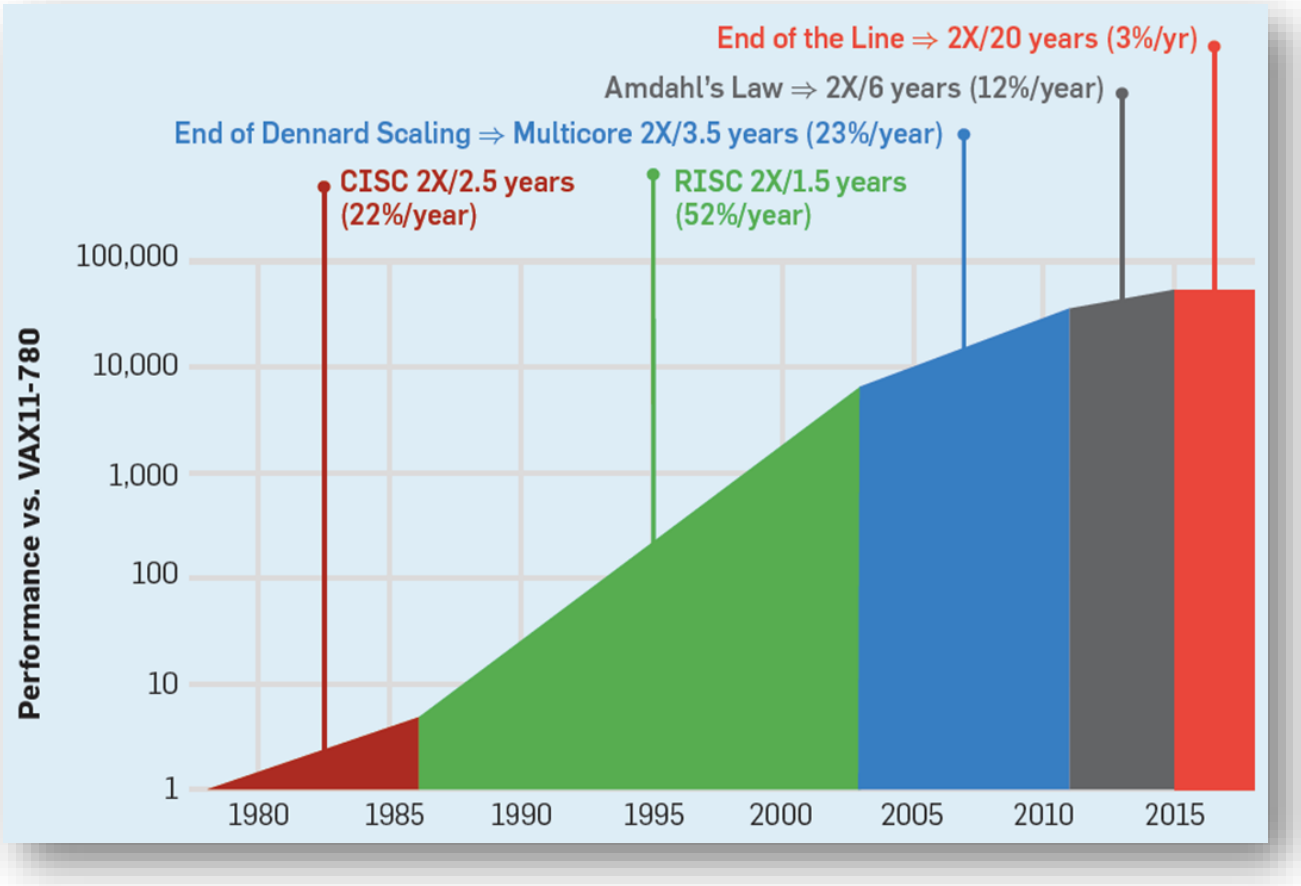
1.1 软件和硬件的定义

- 指令是软件和硬件的媒介，指令的复杂度决定了系统的软硬件解耦程度。
- 按照单位计算(指令)的复杂度，处理器平台大致分为CPU、协处理器、GPU、FPGA、DSA、ASIC。
- 从左往右，单位计算越来越复杂，灵活性越来越低。
- 任务在CPU运行，则定义为软件运行；任务在协处理器、GPU、FPGA、DSA或ASIC运行，则定义为硬件加速运行。



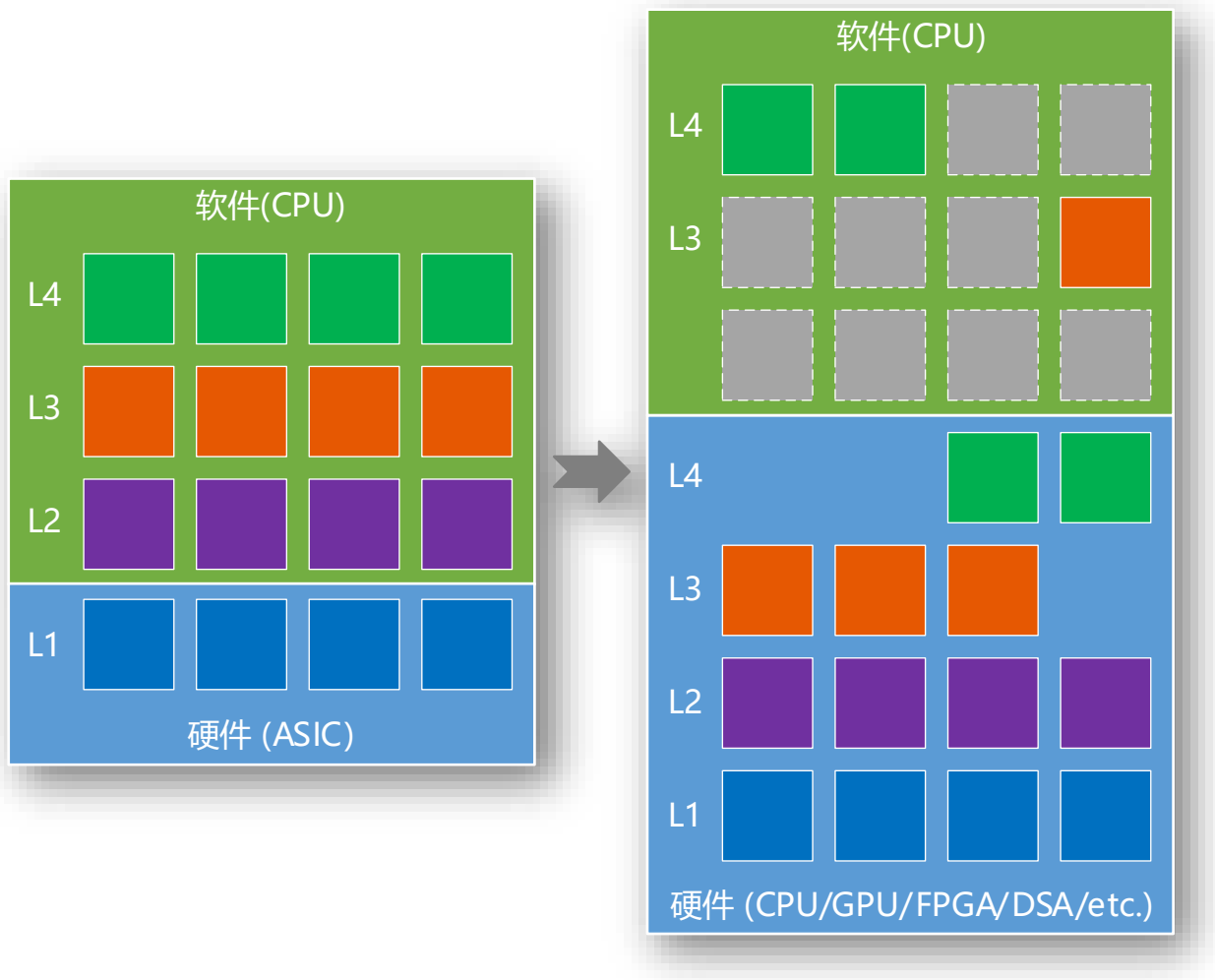
1.2 软硬件融合的背景

- 软件应用层出不穷，并且快速迭代。两年一个热点，已有热点技术仍在快速演进。
- 硬件越来越复杂。工艺持续2D->多层堆叠技术3D->Chiplet多Die互联4D。芯片规模越来越大，一次性成本及研发风险越来越高。
- CPU性能瓶颈。但其所需运行的工作负载数量和单个工作负载CPU资源消耗仍在增加，必须硬件加速。
- 软硬件之间的鸿沟越拉越大。CPU软件性能低下，定制ASIC难以大规模复制。软件迭代越来越快，硬件迭代却越来越慢。芯片高投入高风险，制约着软件的发展。



1.3 软硬件融合

- 哪些任务适合卸载？ ①性能敏感，占据较多CPU资源；②广泛部署，运行于众多服务器。
- ①复杂分层的系统，②CPU性能瓶颈，③宏观的规模，④特定场景服务，使得：软硬件融合的过程其实就是系统不断卸载的过程。

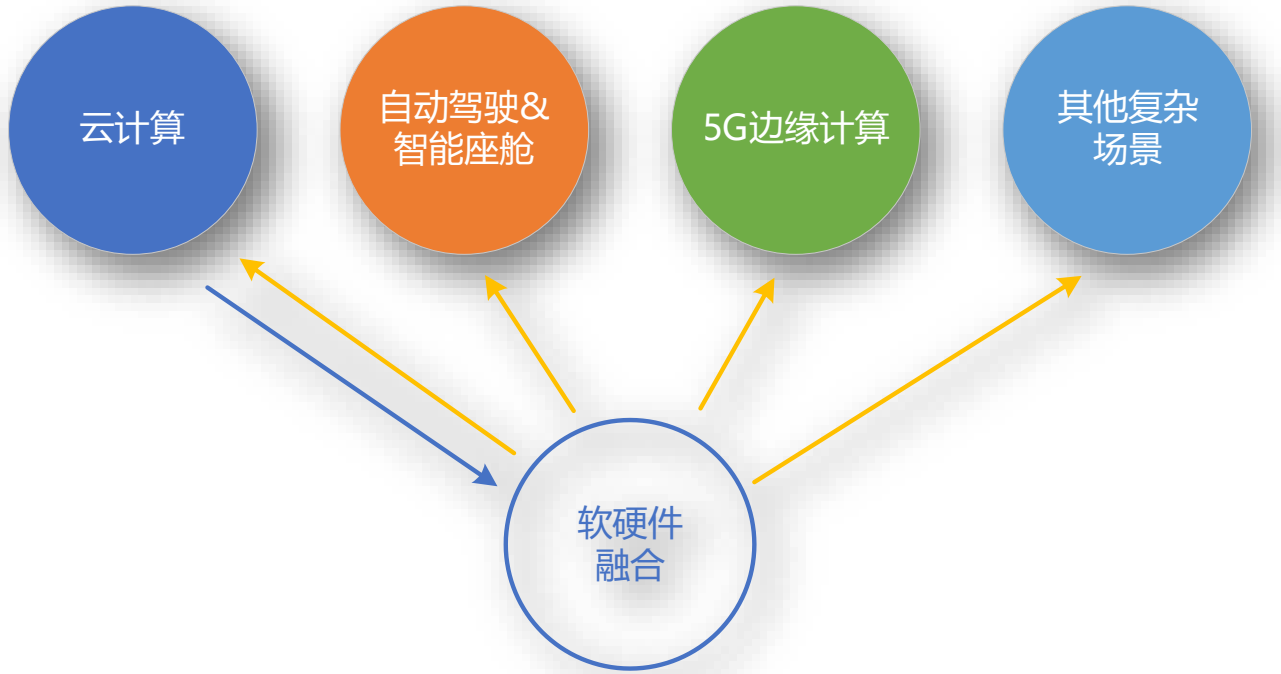


1.3 软硬件融合

- 不改变系统层次结构和组件交互关系，但打破软硬件的界限。
- 传统分层很清晰，下层硬件上层软件；软硬件融合的分层分块，每个任务模块是软件还是硬件，或软硬件协同，都有可能。
- 分层的系统，越上层越灵活软件成分越多，越下层越固定硬件成分越多。
- 庞大的规模以及特定场景服务，使得云计算底层Workload逐渐稳定并且逐步Offload到硬件（被动趋势）。
- 软硬件融合架构，使得“硬件”更加灵活，功能也更加强大，这样使得更多的层次功能向“硬件”加速转移（主动抢占）。

1.4 软硬件融合的应用领域

- 面向未来：复杂系统场景，超异构混合计算，算力需求再上1-2个数量级。
- 云计算，相对性能要求最高，但考虑的因素较少（主要是算力，以及如何更好的提供算力）。
- 软硬件融合从云计算抽象出来，反过来指引包括云计算、自动驾驶等复杂系统场景的芯片及系统设计。



1.5 软硬件融合的意义

- 软硬件融合，既是理论和理念，也是方法和解决方案
- CPU + 协处理器 + GPU + FPGA + DSA + ASIC的超异构混合计算
- 每个Workload都是在软硬件均衡/解耦基础上的再协同
- 连接（软件之间的连接、软硬件的连接、硬件之间的连接）和调用的极致性能和灵活性
- 让硬件更加灵活、弹性、可扩展，弥补硬件和软件之间的鸿沟
- 兼顾软件灵活性和硬件高性能，既要又要
- 应对云计算、大数据及人工智能等复杂应用挑战
- 解决芯片一次性成本过高导致的设计风险
- 等等

2 云计算软硬件融合的相关技术

01 软硬件接口

软件和硬件高效的数据交互

02 高性能的网络

基于RDMA和拥塞控制

03 算法加速和任务卸载

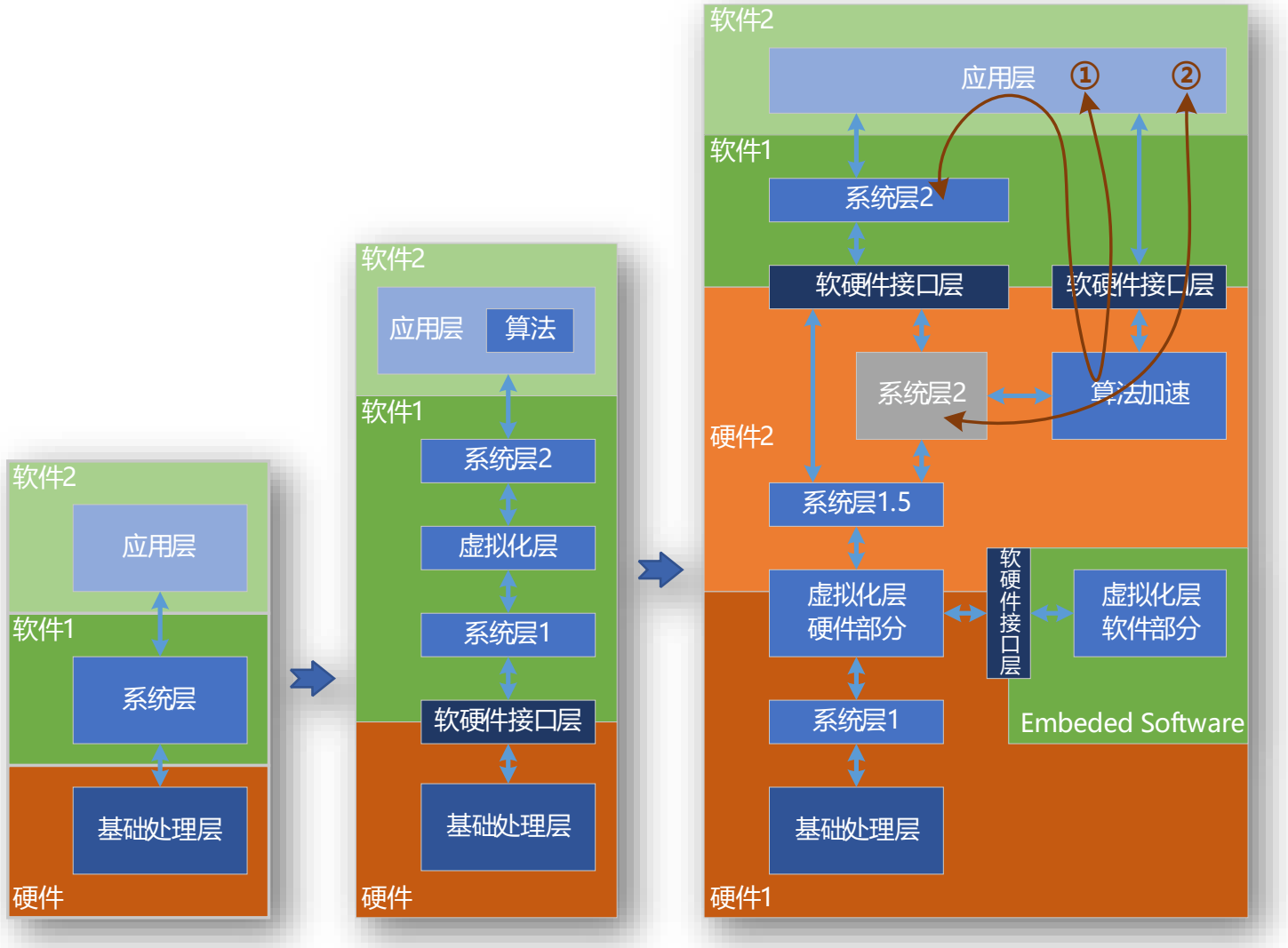
算法硬件实现以及任务卸载框架

04 虚拟化的硬件加速

虚拟化处理的高性能实现

05 异构计算加速平台

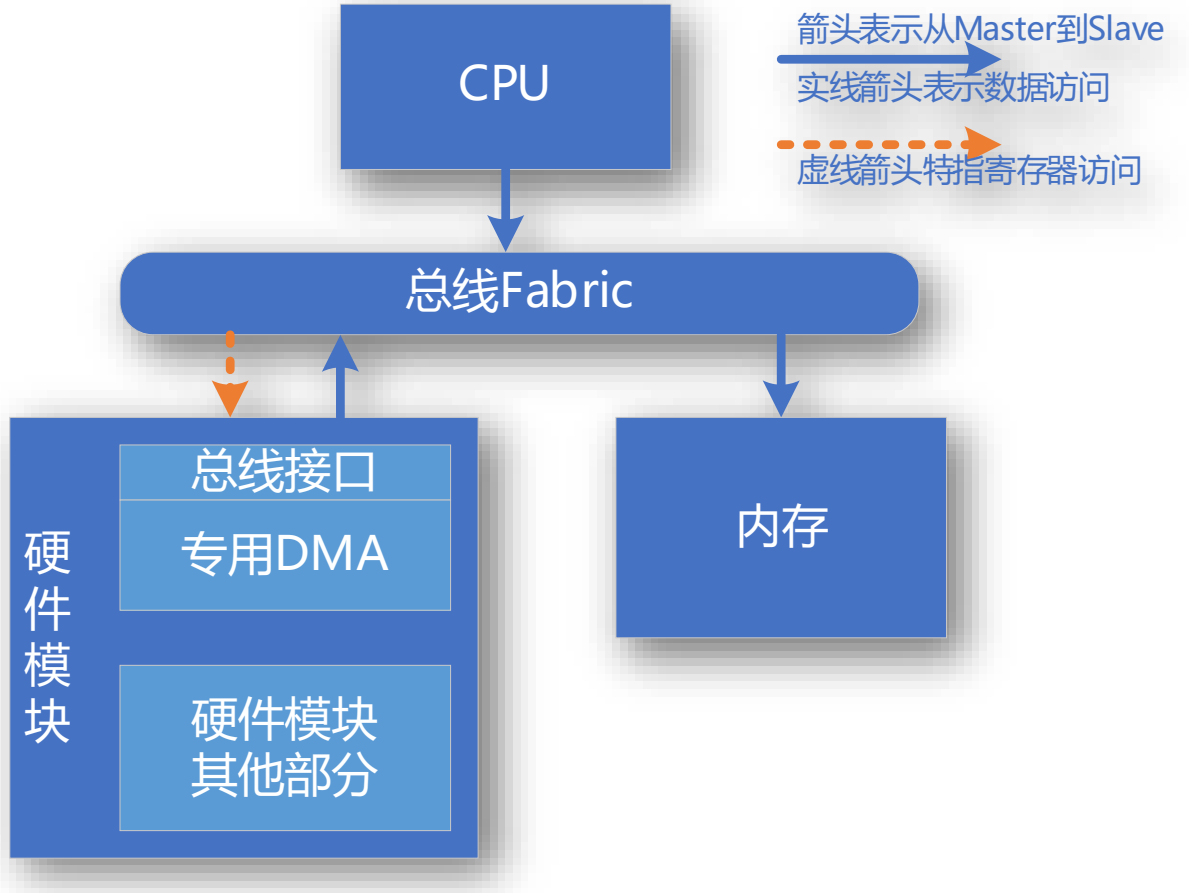
GPU、FPGA以及DSA异构加速平台



2.1 软硬件接口

软硬件接口定义

- CPU软件和其他硬件组件之间数据通信的接口，既包括数据面交互，也包括控制面交互。
- 可以是IO设备接口，可以是加速器接口；更广义的，也可以是封装的高层次服务接口。
- 基于生产者消费者模型，驱动和设备。
- 组成部分：软件驱动、设备的硬件接口子模块、共享队列以及传输的总线。



软硬件接口演进

1. 软件轮询硬件寄存器
2. 硬件中断
3. 硬件DMA
4. 共享队列
5. 用户态软件轮询
6. 多队列并行

2.2 网络高性能

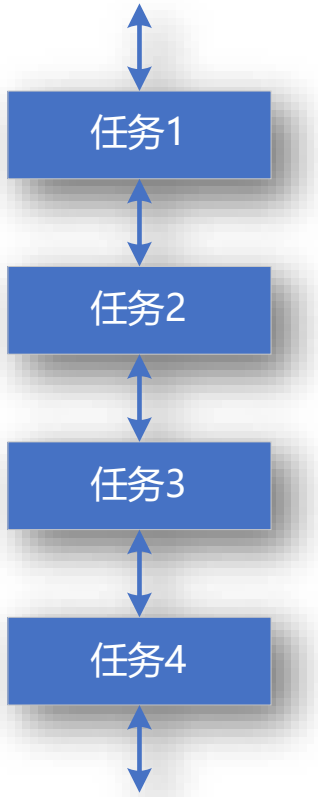
云计算系统持续解构，东西向网络流量激增，服务器堆栈延迟问题凸显。

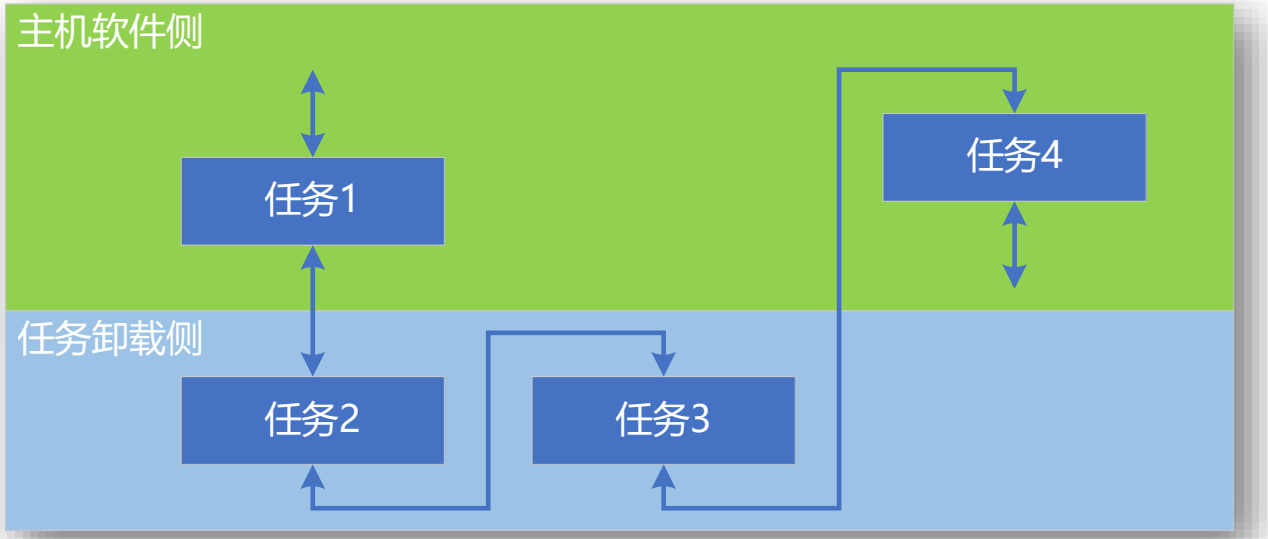
- 网络容量升级，例如整个网络从25Gbps升级到100Gbps；
- 轻量协议栈，数据中心网络是局域网络，距离短/延迟敏感，不需要复杂的用于全球互联的TCP/IP协议栈；
- 网络协议处理硬件加速；
- 高性能软硬件交互：高效交互协议 + PMD + PF/VF/MQ；
- 拥塞控制：低延迟、高可靠性（低性能抖动）、高网络利用率；
- 案例： AWS SRD/EFA和阿里云HPCC。

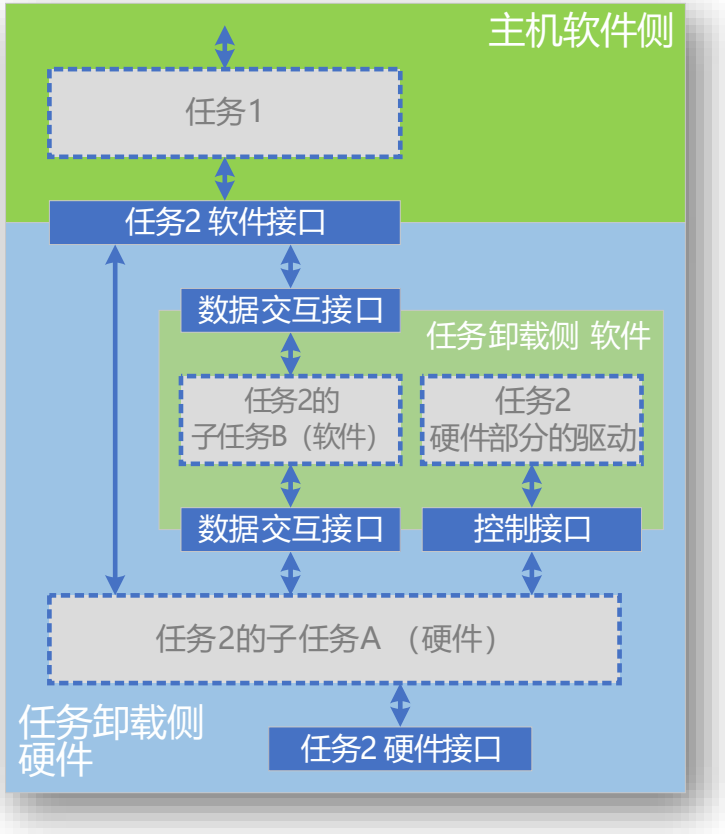
2.3 算法加速和任务卸载

任务卸载通常指：主机软件任务卸载到其他独立硬件；任务卸载以算法“加速器”为核心。

卸载原则：原有调用关系保持不变。软件调用软件转换成软件调用硬件、硬件调用软件或硬件调用硬件。







2.4 虚拟化的硬件加速

抽象是对原有对象的封装，虚拟化是抽象后的复制；虚拟资源通过时间、空间的分割以及抽象模拟共享物理资源。

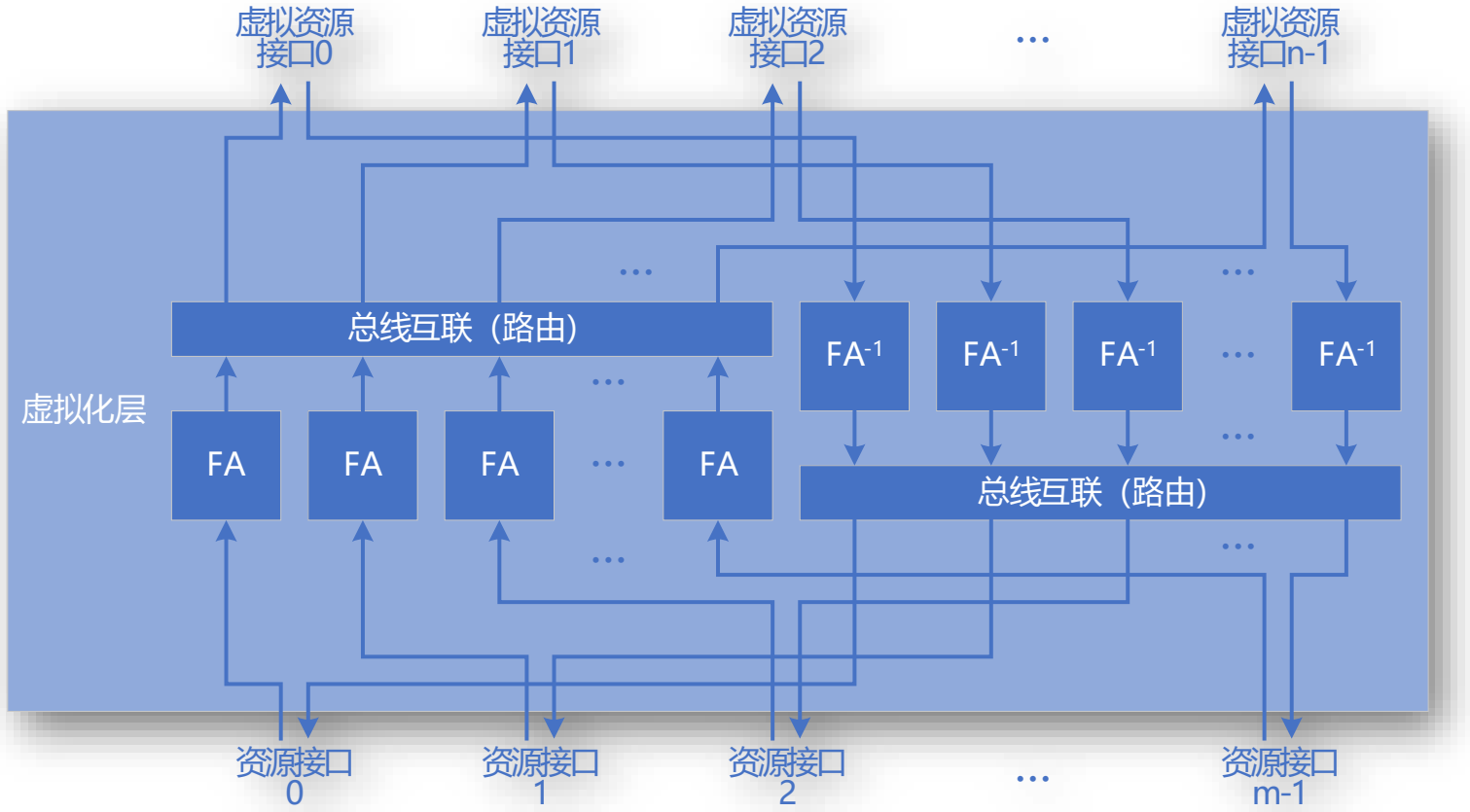
例如，网络虚拟化是虚拟地址到物理地址的映射，存储虚拟化是虚拟块/盘到物理块/盘的映射。

虚拟化模型

- 数学模型：y=FV(x)
- 架构模型，通过接口呈现虚拟化特性

虚拟化的硬件处理

- 流水线：指令流驱动 vs 数据流驱动
- 映射机制：映射算法 vs 映射表
- 缓存机制：主动 vs 被动



2.5 异构计算加速

云计算异构加速主要用于业务应用，权衡：既要保证加速的性能，还要考虑加速的弹性。

01 基于GPU的异构加速

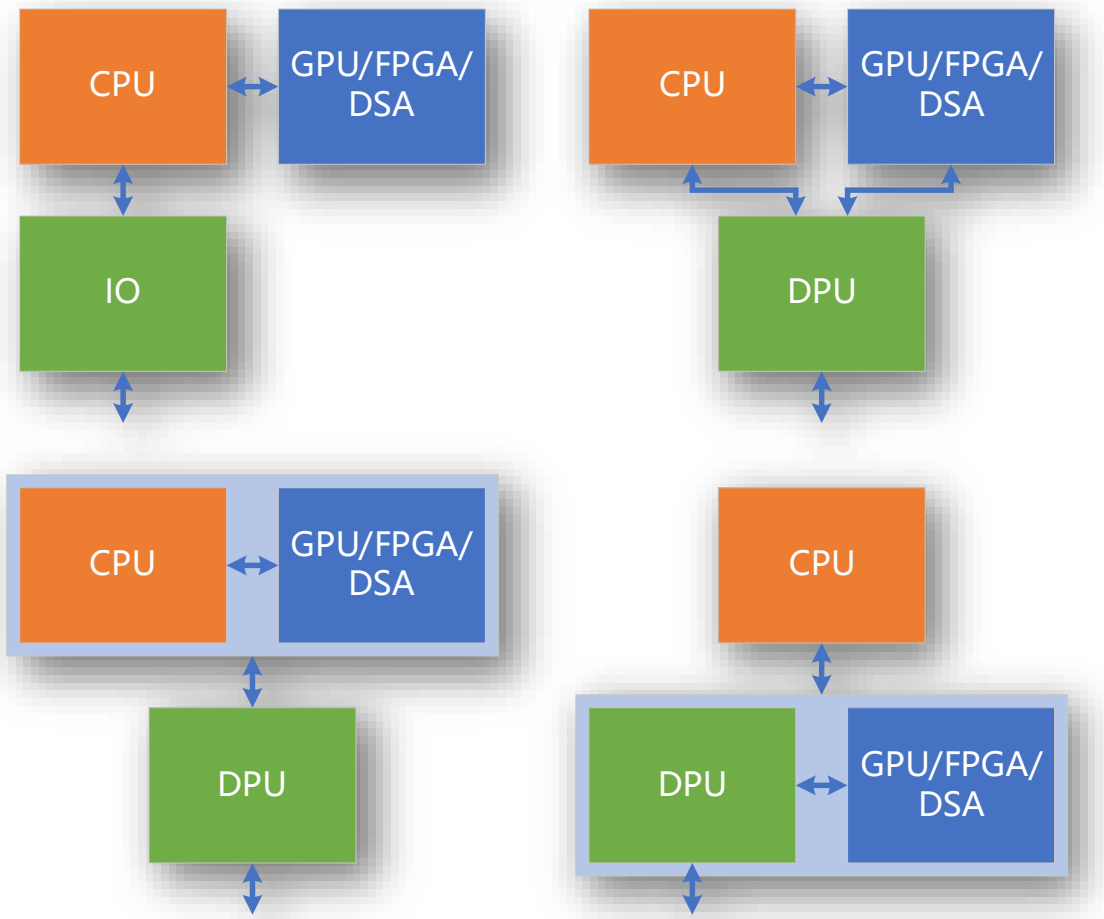
GPGPU+CUDA， GPU异构加速本质是众多并行的高效能通用处理器，CUDA编程友好性。

02 基于FPGA的异构加速

FPGA的硬件弹性跟云非常契合，加速框架Shell/引擎Kernel，运行时RT，开发Stack等。

03 基于DSA的异构加速

DSA是从ASIC回调，相比ASIC具有一定通用性，可以覆盖较多的场景。

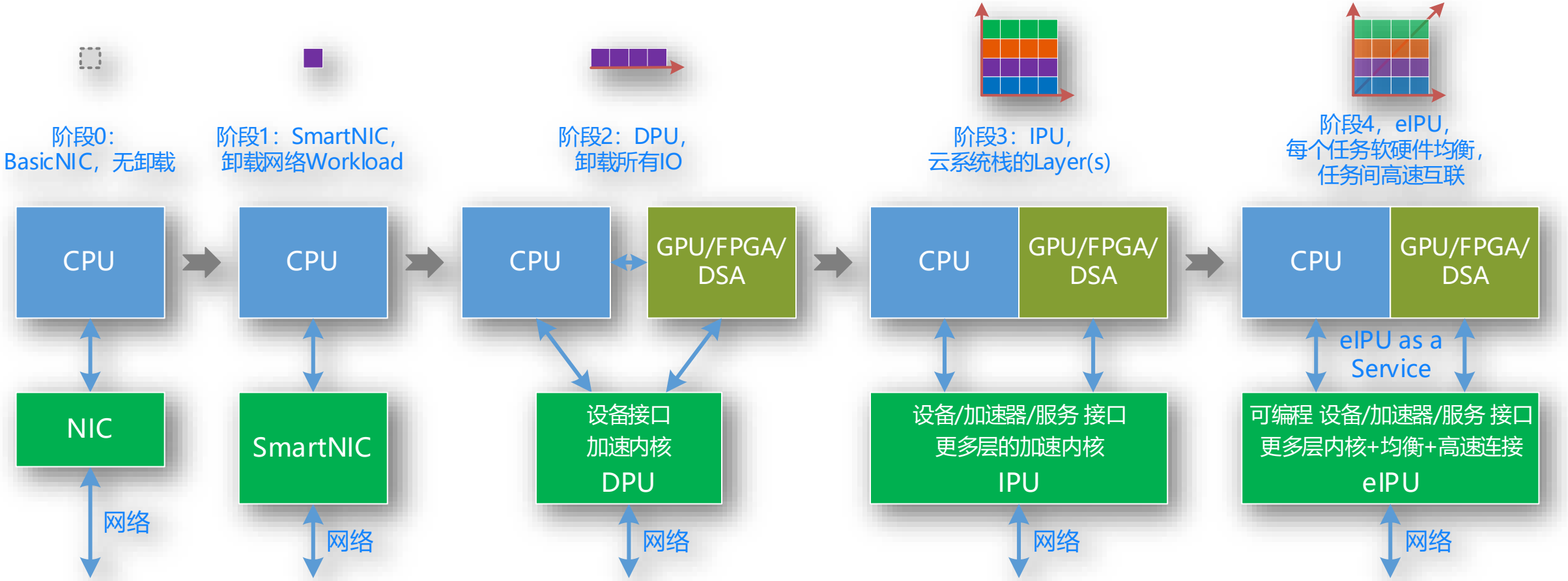


3 DPU/IPU：数据中心的通用加速平台

无规模，不DPU/IPU。需要有独立的加速平台，可以不断的把CPU的软件Workload卸载到硬件加速。

DPU/IPU主要用于应用层之下的通用任务加速，独立GPU/FPGA/DSA主要用于应用层业务加速。

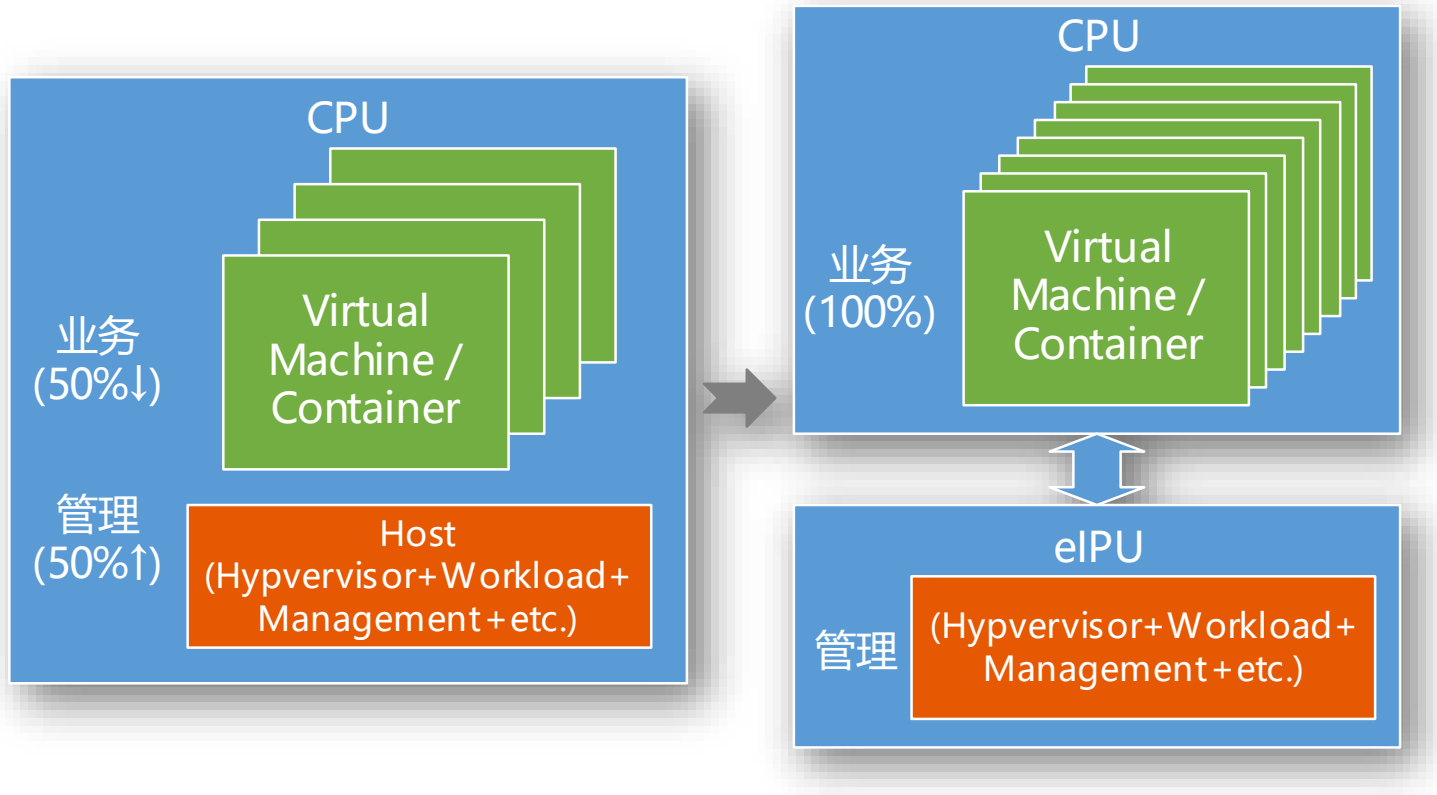
NVIDIA 2020年5月发布DPU，10月大张旗鼓宣传；作者2020年8月提出四阶段论；Intel 2021年6月发布IPU。



3.1 额外的价值：业务和管理分离

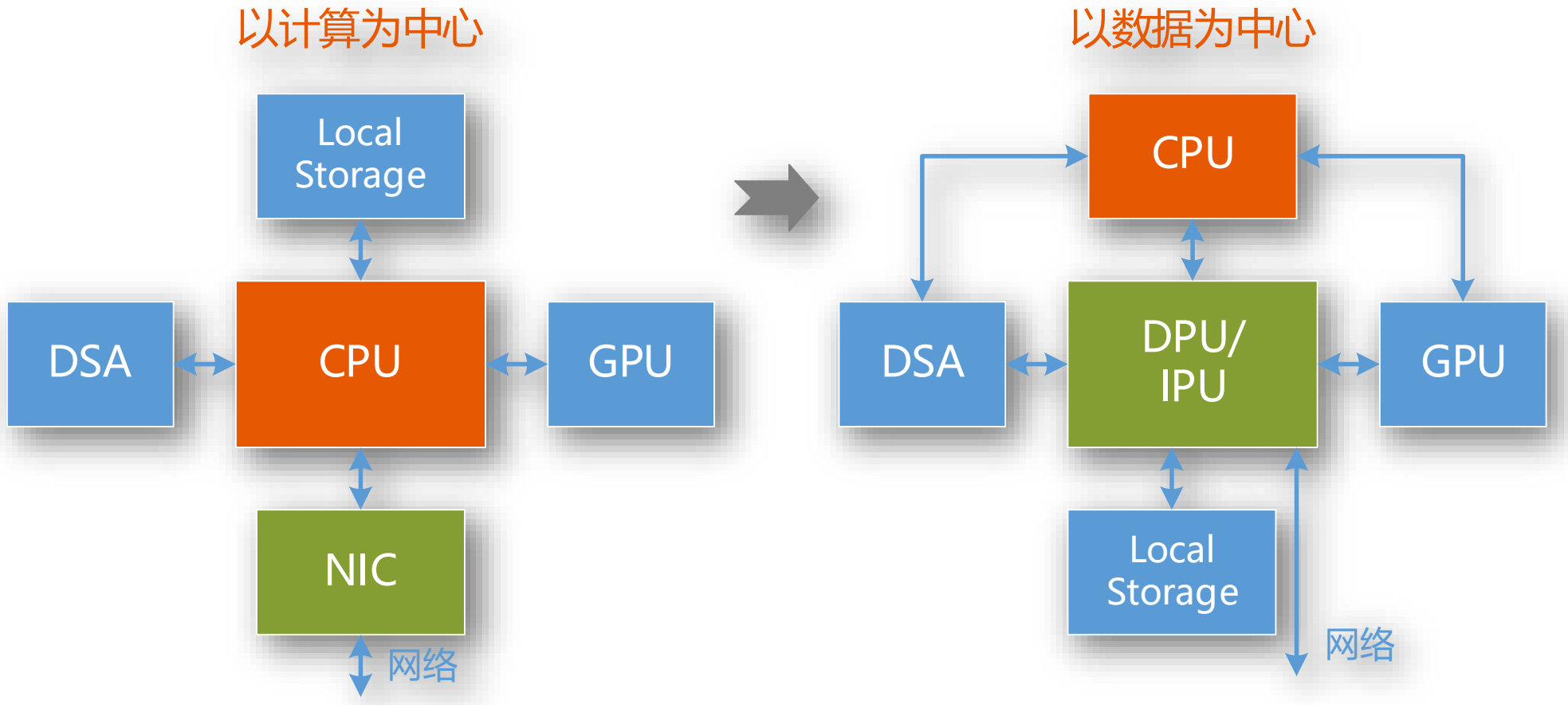
典型价值

- CPU资源完全交付
- 更高可扩展性，灵活主机配置
- 传统客户方便上云（虚拟化嵌套）
- 主机侧安全访问
- 物理机的性能 + 虚拟机的可扩展性及高可用
- 统一公有云和私有云运维



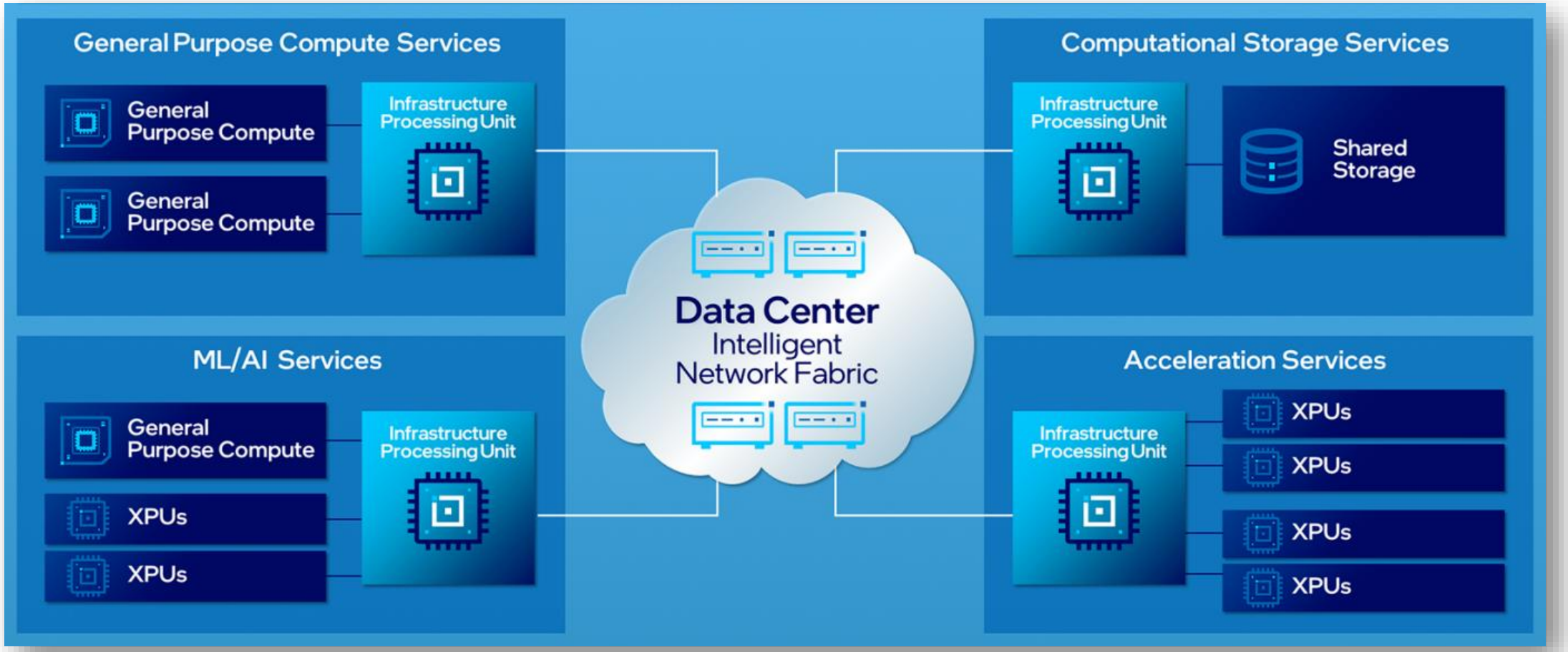
3.2 以数据为中心

CPU性能瓶颈，IO带宽持续增大，IO成为系统瓶颈。DPU/IPU逐渐吞噬CPU和GPU的通用工作任务。

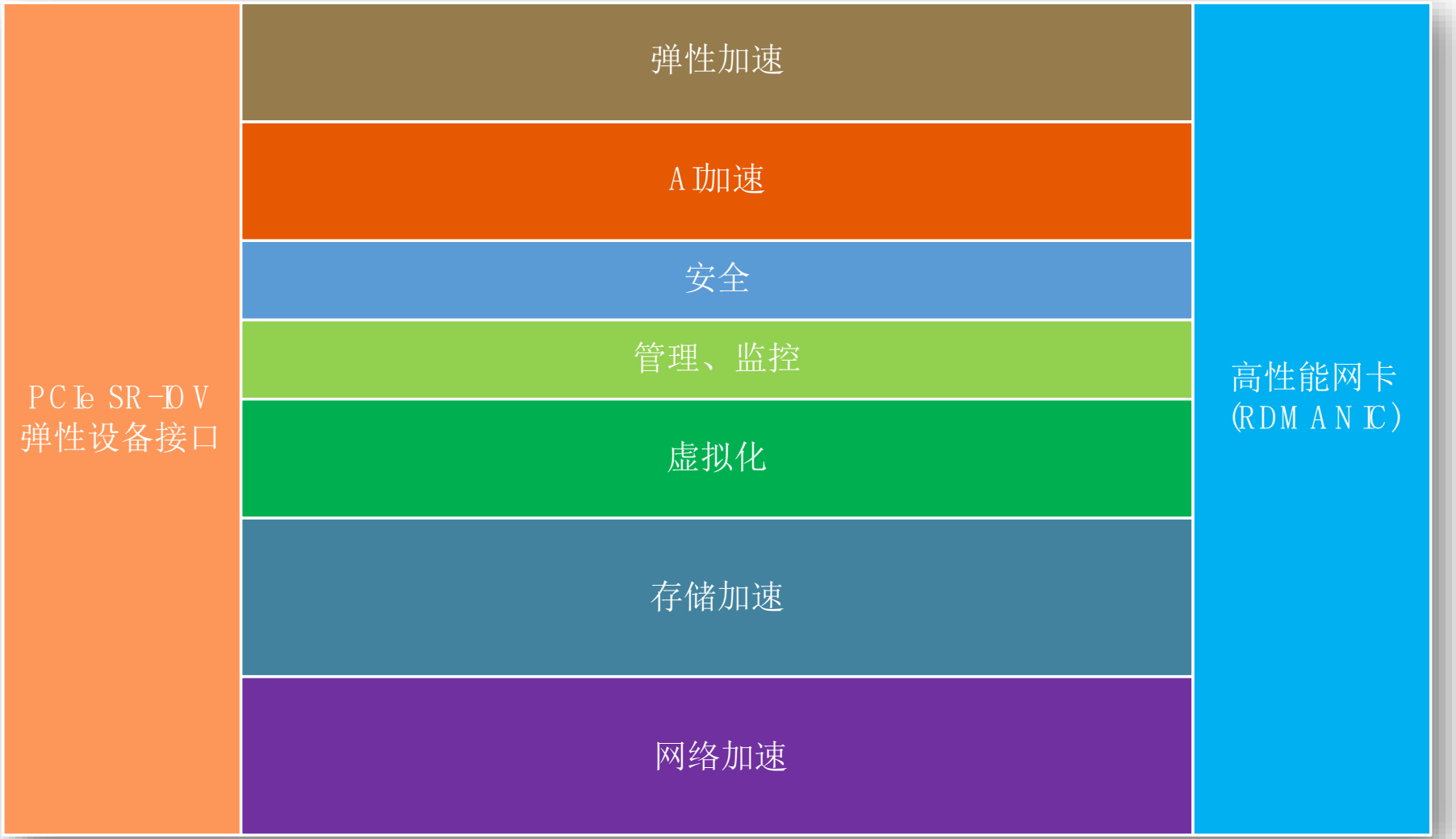


3.3 Intel's View: 未来数据中心架构

Intel IPU是一种可编程网络设备，可通过安全地加速数据中心的功能来智能地管理系统级基础设施资源。



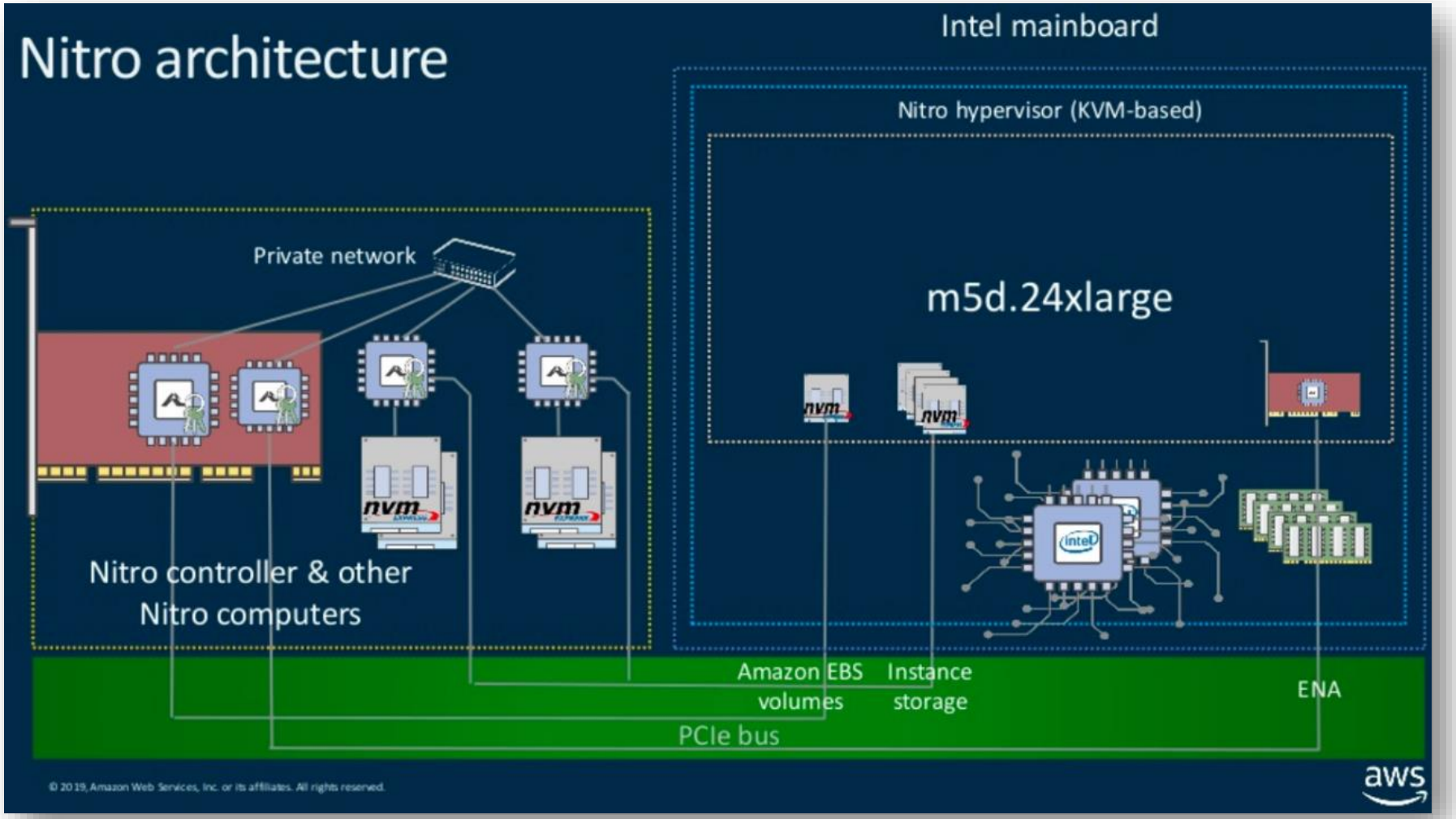
3.4 DPU/IPU的常见功能



3.5 AWS Nitro系统

AWS Nitro系统包括:

- VPC加速卡
- EBS加速卡
- 本地存储加速卡
- Nitro控制器
- 安全芯片
- Lite Hypervisor



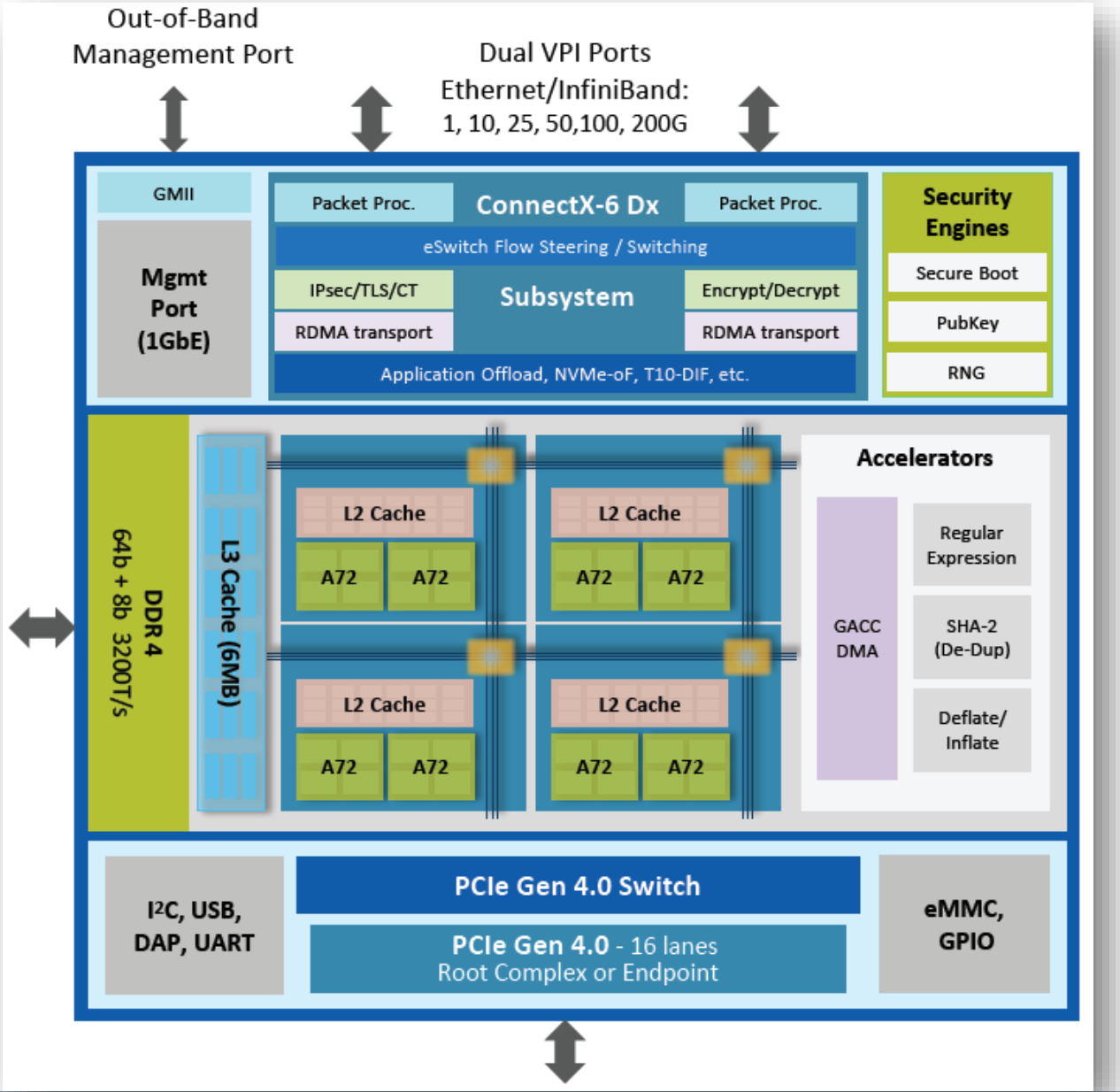
3.6 NVIDIA DPU

优势

- 硬件的网络虚拟化
- RDMA高性能网卡
- 安全类Feature
- 单芯片SOC
- 公开市场唯一落地方案

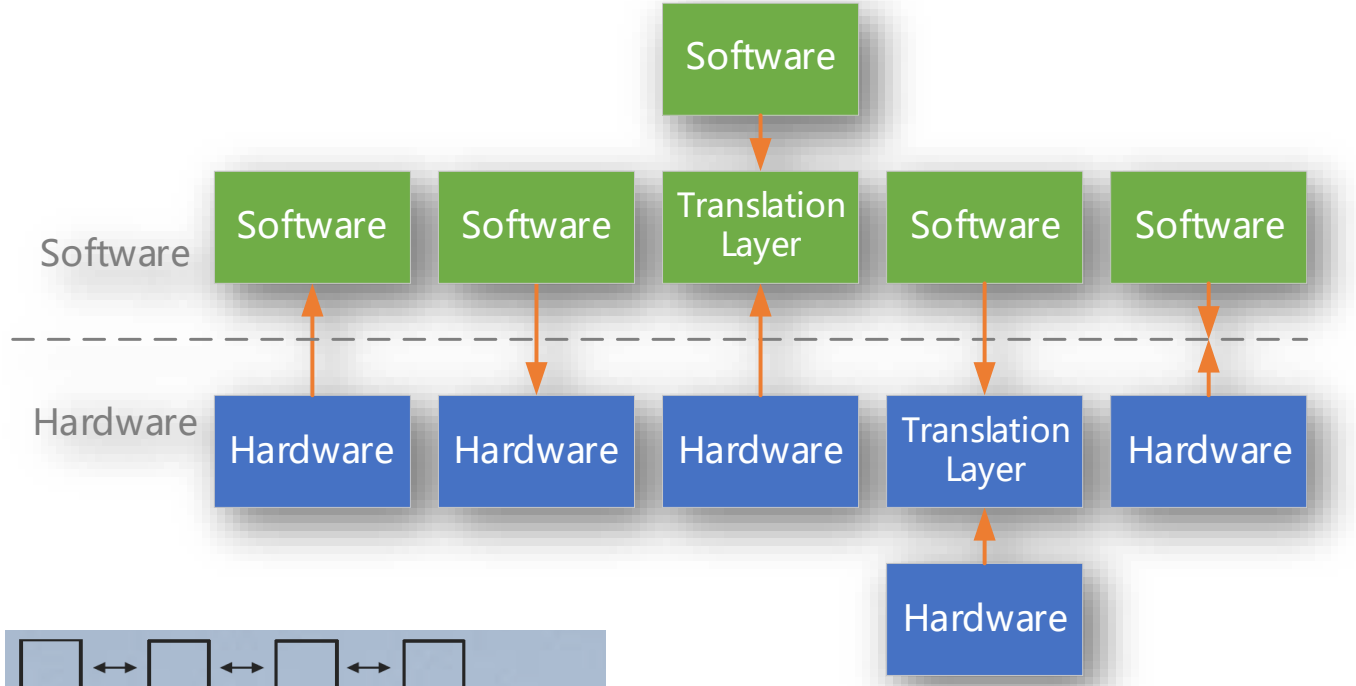
待改进

- 存储软件Offload
- 非标准接口
- 功能扩展性和差异化
- 网络数据面可编程



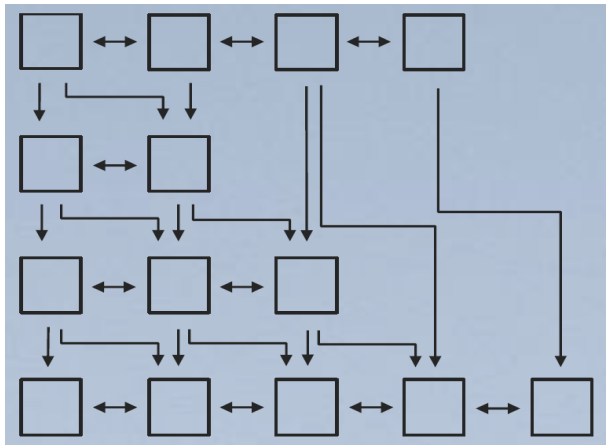
4.1 软硬件接口的形态

1. 硬件定义接口，软件适配
2. 软件定义接口，硬件适配
3. 硬件/软件定义接口，软件接口适配层
4. 硬件/软件定义接口，接口适配层卸载
5. 软件硬件设计遵循标准接口（标准、高效、开源、迭代）



接口泛化：

- 层和层之间，块和块之间的接口；
- 软件和软件之间的接口，硬件和硬件之间的接口，软硬件之间的接口。



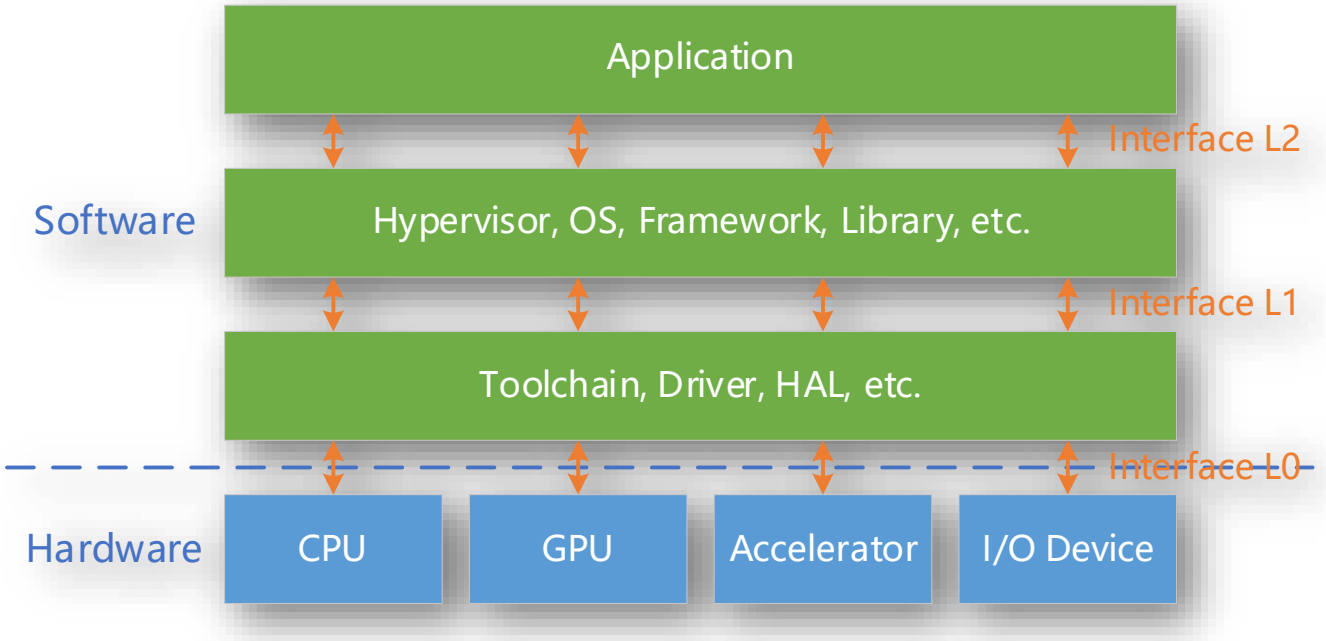
4.2 软硬件融合·乌托邦

1 超异构混合计算架构

- CPU+GPU+FPGA+DSA等多种架构处理引擎组成的混合计算
- 既要又要：接近CPU的灵活性，接近ASIC的性能

2 平台化&可编程

- 软件定义一切，硬件加速一切
- 完全可软件编程的硬件加速平台
- 业务逻辑完全由软件编程决定
- 能够满足多场景、多用户的需求，能够满足业务的长期演进



3 标准&开放

- 接口标准、开放，持续演进
- 拥抱开源生态
- 融入云原生
- 用户无（硬件、框架等）平台依赖

THANKS