

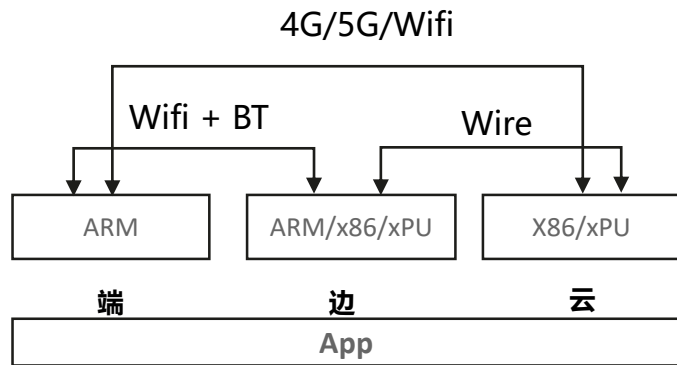
CCF-华为胡杨林基金 系统软件专项 操作系统课题指南

2023年4月7日



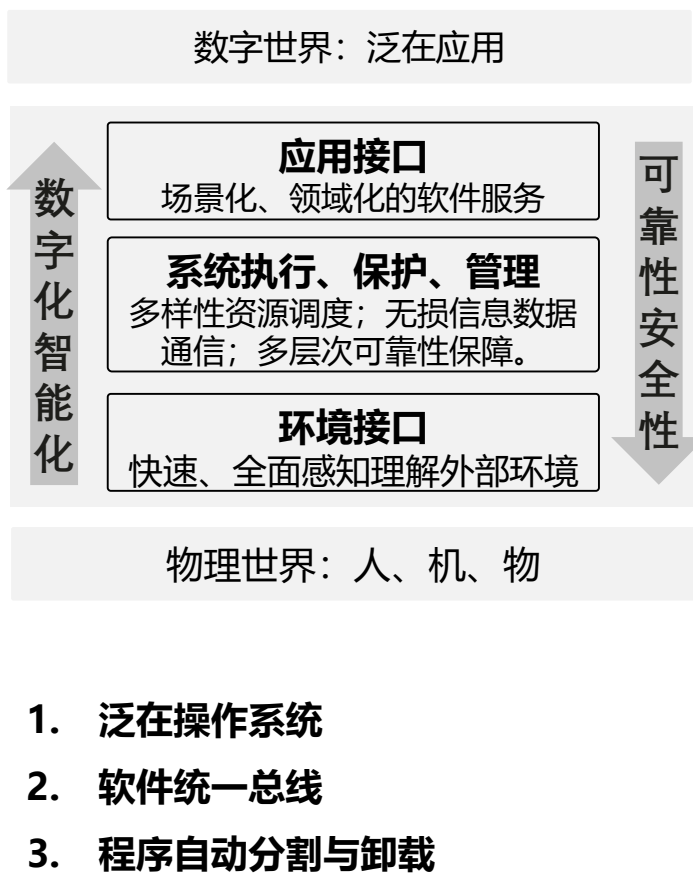
面向云网边端协同的操作系统

产业需求和挑战



1. **云网边端协同的操作系统**: 极轻量的设备也可以具备富生态能力; 以人为中心的多方节点协同操作系统
2. **管理海量异质泛在资源**: 云网边端不同的链接方式与不同的硬件架构, 提供协同的应用开发、执行和维护; 保障以人为核心的系统的安全可靠

潜在技术方向



技术挑战

1. **分布式协同的操作系统架构**
 - OS架构支持组件运行在多个设备, 极轻量设备无需运行完整OS
 - 基于WIFI等近场网络实现 RPC 与内存访问语义
2. **算力运筹**
 - 计算负载的低成本跨节点协作和迁移
 - 根据可用硬件资源与网络状态, 确定分割与迁移的最佳方案
3. **多层次可靠性保障**
 - 以人为中心向外逐层提供冗余备份、隐私加密、攻击防御等安全可靠保障系统

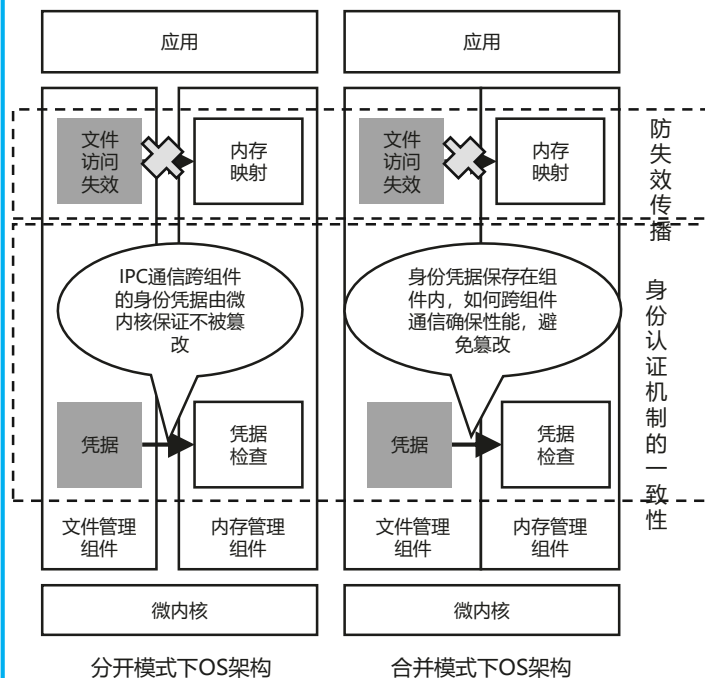
操作系统组件化可分可和技术研究

产业需求和挑战

不同架构的OS可以给其上的最终产品带来不同的性能、内存、安全和可靠性的竞争力。然而每种架构的OS重新开发会带来巨大的开发成本。通过组件化与可分可合，使得同一份OS组件可以拼装成不同的OS架构，为OS架构的选择带来更多的灵活性。

在可分的地址空间独占模式下，不同的组件由不同的地址空间隔离确保安全；然而在可合的地址空间共享模式下，如何能够保持组件之间失效不会互相传递、原有身份认证机制不被绕过，成为支持可分可合在安全上的关键问题。

潜在技术方向



[1] ARMlock(CCS' 14): 使用 ARM 的 Memory Domain 机制隔离组件

[2] Hilps(NDSS' 17): 使用 ARM64 的 TxSz 位来控制组件内存的可见性

技术挑战

1. 组件行为和身份认证方式和功能在可分可合的模式下保持一致，无需设计两套
2. 在可合的共享地址空间模式下，限制失效扩散，身份认证的凭据不被篡改
3. 在跨地址空间和同地址空间内的组件间通信，能够保证高效，且正常执行开销小

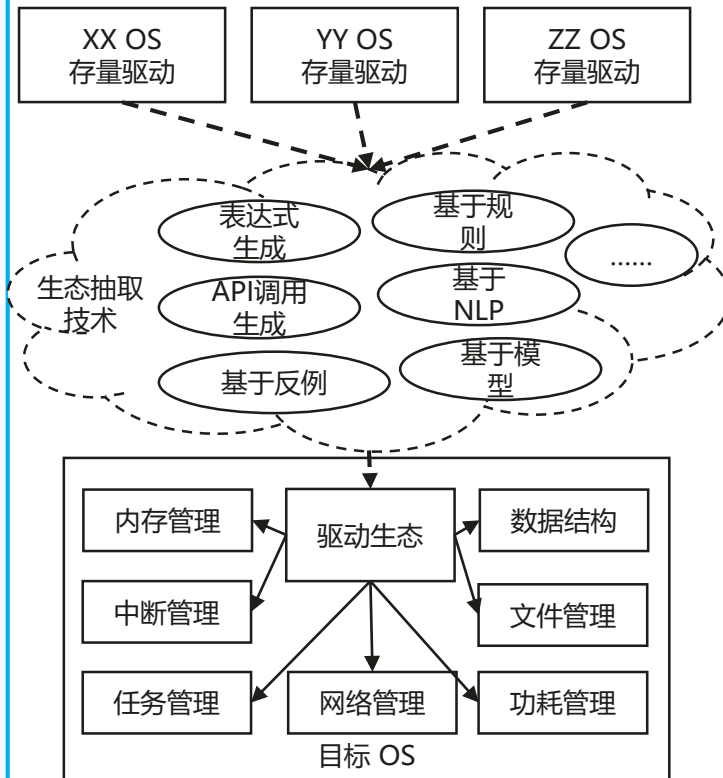
操作系统组件化生态抽取技术

产业需求和挑战

新操作系统的实现通常面临南向的生态问题，即支持海量的设备驱动。传统Linux提供的丰富的南向生态，然而由于KABI的数量众多、种类复杂，并且受限于GPL协议的影响，这些海量的驱动难以直接使用。

如何能够识别出这些海量驱动的运行模式，并且和目标OS相结合，使得这些驱动的设备能够在目标OS上提供相应的功能，成为扩展南向生态的关键问题。

潜在技术方向

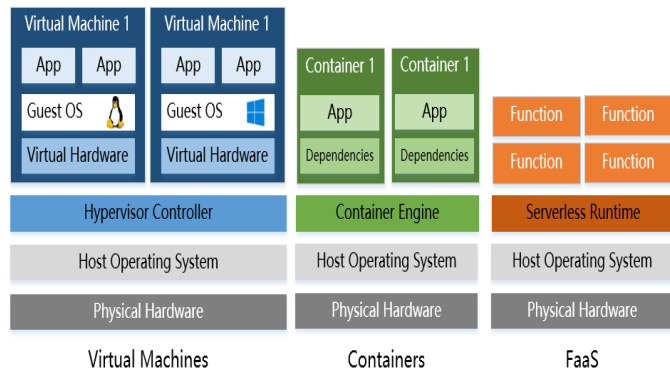


技术挑战

1. 准确识别、区分驱动代码中的硬件逻辑和内核接口的使用。由于同样的功能代码，既可以为硬件逻辑准备数据，也可以为内核接口准备数据。因此需要准确区分这两种目标
2. 准确识别硬件工作模式：硬件有常规模式和失效模式两种，均需要支持才能获得完整的功能
3. 目标OS接口自动适配：生成器需要识别目标OS的接口规范，并基于硬件逻辑生成代码

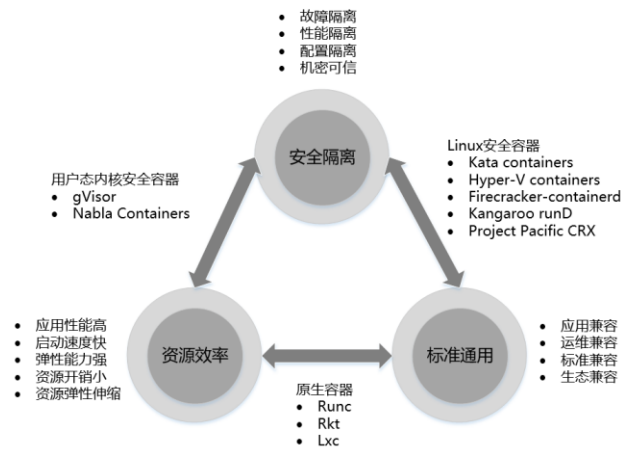
虚拟机&容器的下一跳计算抽象

产业需求和挑战



1. 云计算推动了虚拟机&容器的快速发展，但万物智联的时代呼吁下一跳计算抽象
2. 函数计算、原子化服务等新的编程范式使得虚拟机&容器运行时显得臃肿和迟缓，面向数据中心资源的精细化运营，使得面向服务的更高效运行时技术成为未来焦点
3. 控制和数据的有效隔离要求更高；传统容器难以面向多租户构建安全隔离环境。

潜在技术方向



图片参考网络：《容器技术20年：容器引擎与江湖门派》

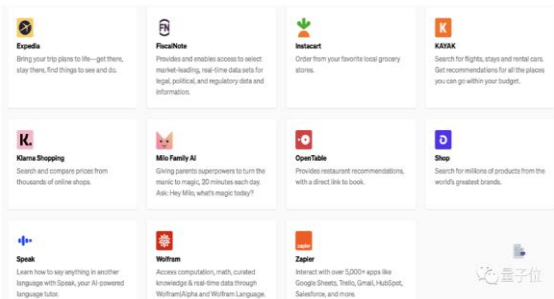
1. 更加极致的低底噪高性能隔离
2. 面向多样性异构算力的统一抽象
3. 面向新服务范式的抽象隔离

技术挑战

1. 面向 LLM 等技术，是否会产生新的抽象隔离技术
 - 如果模型即服务成为趋势，如何有效做好租户间的隔离
2. 虚拟化和容器直接结合，更安全但性能不理想：
 - 基于虚拟机或用户态内核构建安全容器，减少容器逃逸的风险；但也导致应用性能下降和底噪升高
 - 虚拟化层导致IO性能显著下降。若通过设备直通提升性能，则大幅增加服务成本；
 - 虚拟机的资源使用底噪对于短时容器过于臃肿。
3. 需要创新的软、硬件内存安全隔离技术
 - 成熟技术存在较多限制
 - SFI通过编译插桩实现内存隔离，不支持动态新增内存隔离区；
 - Intel MPX支持的内存分区数量有限，只支持16个keys标识隔离区；
 - KASAN会引入额外大量开销，占用额外的CPU资源。
 - 新技术的隔离能力和结合方式需要探索
 - 包括但不限于CHERI, TDX, CCA 等

基于人工智能的操作系统架构探索

产业需求和挑战



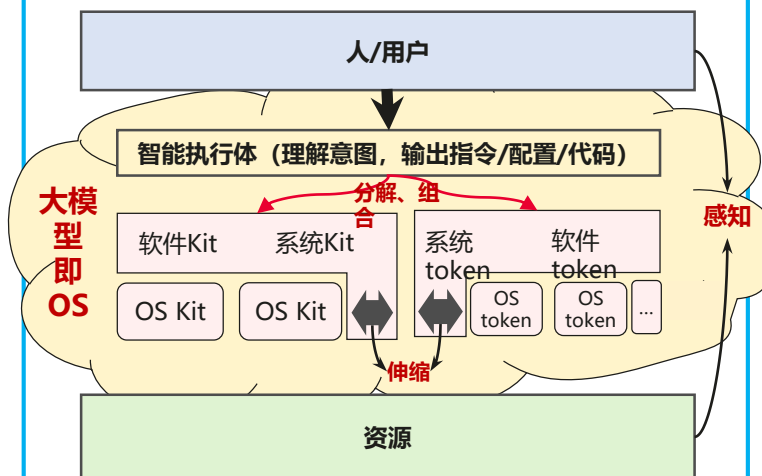
ChatGPT目前已支持的第三方插件，包括餐厅推荐、旅行计划、购物、AI语言老师、网上店铺，以及学术界知识应用Wolfram、应用库（与5000多个应用程序交互，包括Google表格），初步展现构建操作系统的可能。

```
➔ heycli git:(master) *
➔ heycli git:(master) * hey list open ports in my machine

Answer: sudo lsof -i -P -n | grep LISTEN
Execute command? (yes/no): yes
Password:
COMMAND  PID    USER  FD  TYPE  DEVICE SIZE/OFF  NODE NAME
launchd   1      root   30u  IPv4  0x6c5c01b5664003e7 0t0  UDP *:137
launchd   1      root   31u  IPv4  0x6c5c01b5664000ff 0t0  UDP *:138
syslogd   89     root   10u  IPv4  0x6c5c01b566fb3ae17 0t0  UDP *:51997
UserEvent 90     root   56u  IPv6  0x6c5c01b56d8f2c97 0t0  TCP [fe80::a92->[fe80::a92:48ff:fe33:4455]:59602 (ESTABLISHED)
UserEvent 90     root   68u  IPv6  0x6c5c01b56d8f32b7 0t0  TCP [fe80::a93 (LISTEN)
93 (LISTEN)
UserEvent 90     root   74u  IPv6  0x6c5c01b56d8f38d7 0t0  TCP [fe80::a93 (LISTEN)
OK (LISTEN)
```

基于自然语言的命令行已经出现雏形，人机交互与系统组织方式有望改变

潜在技术方向



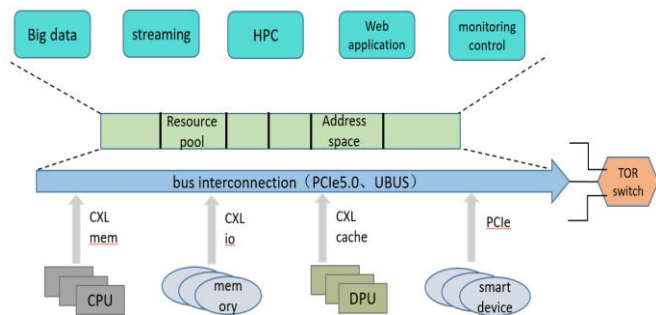
1. 大模型即操作系统本身
2. 改变人机交互范式
3. 改变系统运维和应用调优范式

技术挑战

- 操作系统是否能够根据外界指令调整自身结构，不断重新分解组合，**软件kit化 → token化**；
- 人机物三元世界的高效交互，大模型能否成为人与机、物交互的有效界面；
- 人工智能算法结果具备不确定性，如何保证这样一个操作系统的数据安全性；
- 如何将专家经验和模型相结合，以降低计算成本。

多样性算力的任务调度

产业需求和挑战



硬件结构的变化:

- 单节点从host-device结构, 演变为更对等的结构
- 集群从分布式结构, 演变为机柜级池化共享的结构

算力多样性对系统设计和任务调度产生挑战:

- 影响因素多
- 决策目标多
- 使用限制多

潜在技术方向

1. 协同调度

- **新型调度框架:** 实现可扩展到更多内存层次、更多样化算力和更大互连节点规模的调度框架。
- **新型调度算法:** 实现复杂度为 $O(1)$ 的调度算法, 支持不同硬件形态和互联结构。

2. 异构OS

- **系统抽象和接口:** 自动将业务模型转化为运行实体。
- **多样化算力管理:** OS自身可以充分利用异构算力, 优化性能; OS功能平滑扩展到越来越大的异构算力规模

技术挑战

1. 多样性算力的统一描述

- 通过统一的规格支持描述多样性算力, 支持不同性能的CPU、GPU、NPU

1. 收益-成本模型复杂

- 多样性算力的对等架构, 复杂组网和异构能力, 使得调度收益的建模、估计、分析极其复杂。当前通用模型和策略不再适用。

2. 融合调度

- 如何实现数据调度和业务调度的协同; 实现不同XPU间调度的协同, 充分发挥算力以及拓扑的局部性优势。

融合数据驱动的系统设计与运行管理

产业需求和挑战

长期以来，OS只建立了从资源信息到资源管理的单一链接。实际上，作为“中间层”，OS也能从它的视角看到数据、计算等要素，却极少收集和使用它们。

目前，OS的角色仅是在来自上层的数据和计算，以及来自下层的资源都已经确定的情况下，完成好自己的管理衔接工作。

那么当三方面的复杂度皆呈爆炸性增长时，OS是否应该保持“袖手旁观”？它是否应该利用能够看到三方面要素的独特性质，走出现有的职能范式，整合这些信息，对上层的数据及计算模式和下层的资源结构主动提出改良建议，形成“外溢”效应？

潜在技术方向

业务应用

系统调优

根据负载自动调优，数据辅助芯片设计

系统安全

数据驱动运行安全

定位定界

故障诊断与预测

学习型操作系统

数据驱动、动态调整、持续演进

硬件 & 芯片

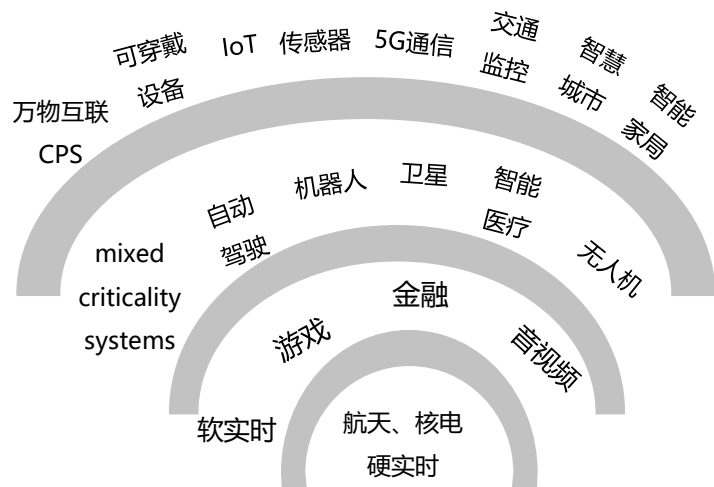
1. **学习型操作系统**：基于数据驱动，重新思考系统设计，保证动态调整，持续演进；
2. **系统智能运维**：准应用灰度故障模型与立体故障树实时构建技术；
3. **系统调优**：根据业务负载系统自动调优，获取的性能数据进一步辅助芯片设计；

技术挑战

- 业务多样化，数量海量，资源布局复杂化，作为联结的操作系统如何帮助上层应用与自身，动态调整自身设计，以达成状态最优？
-
- **基于应用、系统KPI的异常检测无法识别灰度故障**：软件设计的模块化原则导致应用/系统之间必然存在观测差异，所以现有技术（有/无监督类异常检测算法等）均只能片面评估应用/系统健康度。
- **基于相关性、知识图谱构建调度故障树准确性、可解释性不足**：现有技术（包括知识图谱类、事件相关性类）无法综合平衡准确性、可解释性；进程/线程间隐式动态故障扩散，进一步对故障树构建实时性提出更高要求。

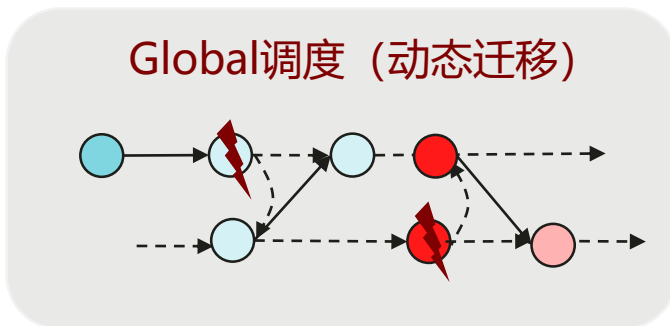
面向多核异构硬件的CPS操作系统

产业需求和挑战



1. **下一代CPS系统：**随着硬件技术不断提升和软件功能飞速发展，CPS系统渴求突破原有的固定硬件和封闭软件，以适应现有的软硬件环境；但因其直接影响物理世界和人身安全，其可靠性、确定性和正确性必须保证。
2. **如何打造一个CPS系统，使得它既可以使用高性能硬件和多功能软件，又可满足CPS对可靠性、确定性和正确性的要求？**

潜在技术方向



- **多核多任务的硬实时调度：**嵌入式高性能硬件往多核方向发展；软件功能复杂化，往多任务方向发展；需要能够保障多核多任务的硬实时调度能力。
- **支持异构硬件的混合关键性调度：**在传统的实时调度的基础上，结合不同场景的实时、功能安全诉求差异，支持在单一硬件上保障不同关键性等级OS的指标达成。

技术挑战

■ 多核硬实时支持：

- 任务拆分至多核并发缩短执行周期，提升单位时间算力，多核任务编排满足硬实时约束的量化复杂度急剧上升。

■ 混合关键性系统需要平衡资源利用率与不同关键性等级隔离：

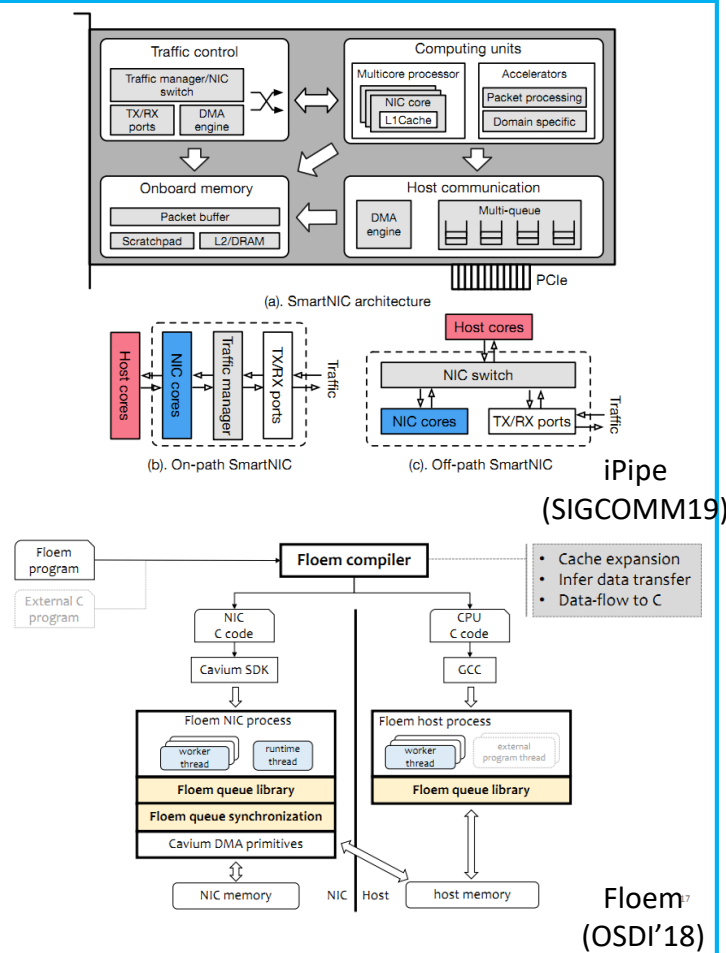
- **CPU之外的算力支持混合关键性：**通过软件和硬件协同的技术，支持非CPU的算力也可以混合共享
- **资源利用率有待提高：**在满足嵌入式场景约束的前提下，如何通过机制、算法实现资源共享，提升OS部署密度
- **难以满足不同关键性等级的需求：**实现资源共享后，如何通过以OS为单位的混合关键性调度，保障不同安全性等级达成，特别是高实时、高功能安全OS

面向可编程网络加速器的通用计算框架

产业需求和挑战

1. SmartNIC等智能网卡逐渐成为网络芯片的标配，除了用来进行网络包处理加速外，是否还能承担其他相关的任务？什么类型的任务适合通过智能网卡来卸载加速？
2. 如何对网络加速器计算能力进行抽象，让普通业务能更方便的使用网络加速器，提高业务并行度和整体业务性能规格？

潜在技术方向



技术挑战

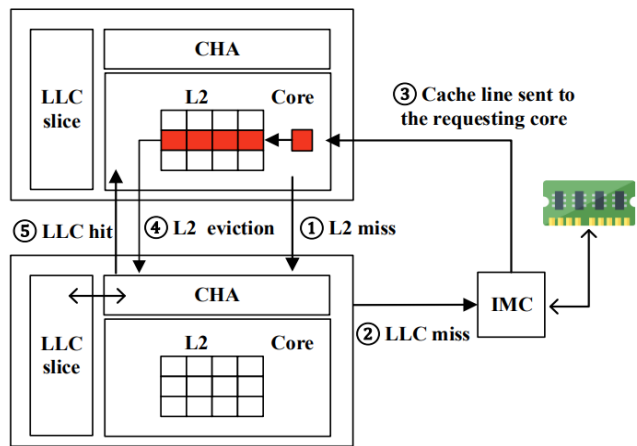
1. 对当前网络加速器的能力进行抽象建模，提取适合网络加速器的计算模型，并实现匹配的编程框架；
2. 实现网络加速器资源的池化和通用化接口，达成加速器资源在管控面和数据面的动态分配；
3. 基于包处理的流程逻辑实现系统资源的软硬件协同调度，实现整体系统吞吐最优；

微架构感知的众核并发控制技术研究

产业需求和挑战

多核下同步原语（包括spinlock，原子操作等同步原语）由于cache/总线等资源冲突，使CPU有效算力未充分用于业务，消耗在同步原语开销上。但目前一般同步原语优化工作从同步逻辑本身出发，优化同步流程，从而提升性能，目前该优化工作基本已经达到极致，比如Linux内核的Qspinlock，以及 CLoF(SOSP' 21)方案。如何感知微架构+cache同步路径，最小化路径长度，来降低同步原语带来的cache同步开销，从而提升系统性能，是一个值得探索的方向。

潜在技术方向



NUMA aware qspinlock, Oracle在Linux内核社区推的技术方案，考虑了NUMA节点内以及NUMA节点外cache同步开销，优先节点CPU程获取到锁，降低NUMA节点间cache line 乒乓，从而提升整体吞吐，在128核下锁操作有30%+提升。
利用CacheDirector(EuroSys' 19),使用微架构感知的方法对NFV进行优化加速。

技术挑战

1. 现有微架构感知方案的问题主要是多数方案仅针对特定应用进行优化，不具备普适性。具体来说，各个优化方案仅感知对其目标应用性能影响最大的微架构状态，不能推广至所有应用。
2. 在超多核CPU上，Spinlock等同步原语的性能（Lock per Second）可提升30%以上

欢迎基于 OpenHarmony 和 openEuler 的研究工作

OpenHarmony

OpenHarmony是由开放原子开源基金会（OpenAtom Foundation）孵化及运营的开源项目，目标是面向全场景、全连接、全智能时代，搭建一个智能终端设备操作系统的框架和平台，促进万物互联产业的繁荣发展。

<https://openharmony.io> contact@openharmony.io

已关注 29588

🏠 概览

📦 仓库 236

📌 任务 604

🔗 Pull Requests 353

📰 动态

👤 成员 106

精选

appexecfwk_standard

Application execution and management framework | 用户程序运行管理框架

🔗 6 ⭐ 49 📄 40

communication_dsoftbus

DSoftBus capabilities, including discovery, networking, and transmission | 软总线发现、组网、传输功能实现

🔗 17 ⭐ 83 📄 60

docs

OpenHarmony documentation | OpenHarmony开发者文档

🔗 644 ⭐ 3840 📄 966

kernel_liteos_a

LiteOS kernel for embedded devices with rich resources | 适用于资源较丰富嵌入式设备的LiteOS内核

🔗 260 ⭐ 1688 📄 739

ace_engine_lite

ACE JS lite framework | 轻量级JS核心开发框架

🔗 88 ⭐ 672 📄 302

community

OpenHarmony community governance, developer contribution guide, contribution agreement, and community ...

🔗 80 ⭐ 365 📄 183

gitee.com/openharmony

openEuler

通过社区合作，打造创新平台，构建支持多处理器架构、统一和开放的操作系统openEuler，推动软硬件生态繁荣发展。

<https://openeuler.org> contact@openeuler.io

已关注 1.7K

🏠 概览

📦 仓库 152

📌 任务 1302

🔗 Pull Requests 200

📰 动态

👤 成员 123

+ 新建仓库

精选

iSulad

iSulad is a light weight container runtime daemon which is designed for IOT and Cloud infrastructure.

🔗 69 ⭐ 306 📄 142

A-Tune

A-Tune is an OS tuning engine based on AI.

🔗 52 ⭐ 226 📄 104

stratovirt

StratoVirt is an opensource VMM(Virtual Machine Manager) which aims to perform next generation virtualization...

🔗 80 ⭐ 287 📄 94

bishengjdk-8

Bisheng JDK 8 is a high-performance, production-ready distribution of OpenDK 8.

🔗 66 ⭐ 347 📄 113

kernel

openEuler 内核是 openEuler操作系统的核心，是系统性能和稳定性的基础，是链接芯片、设备与业务的桥梁。...

🔗 200 ⭐ 635 📄 304

community

Community governance is listed in the repository.

🔗 104 ⭐ 173 📄 468

gitee.com/openeuler

把数字世界带入每个人、每个家庭、每个组织，
构建万物互联的智能世界

Bring digital to every person, home and organization
for a fully connected, intelligent world



Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and
organization for a fully connected,
intelligent world.

**Copyright©2018 Huawei Technologies Co., Ltd.
All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

