

Xenomai Seminar 2024

实时性能调优 - 提高基于KVM虚拟化的混合关键性系统的实时确定性

*Real-Time Performance tuning for KVM based Mixed Criticality System
workload consolidation*

王虎文



内容

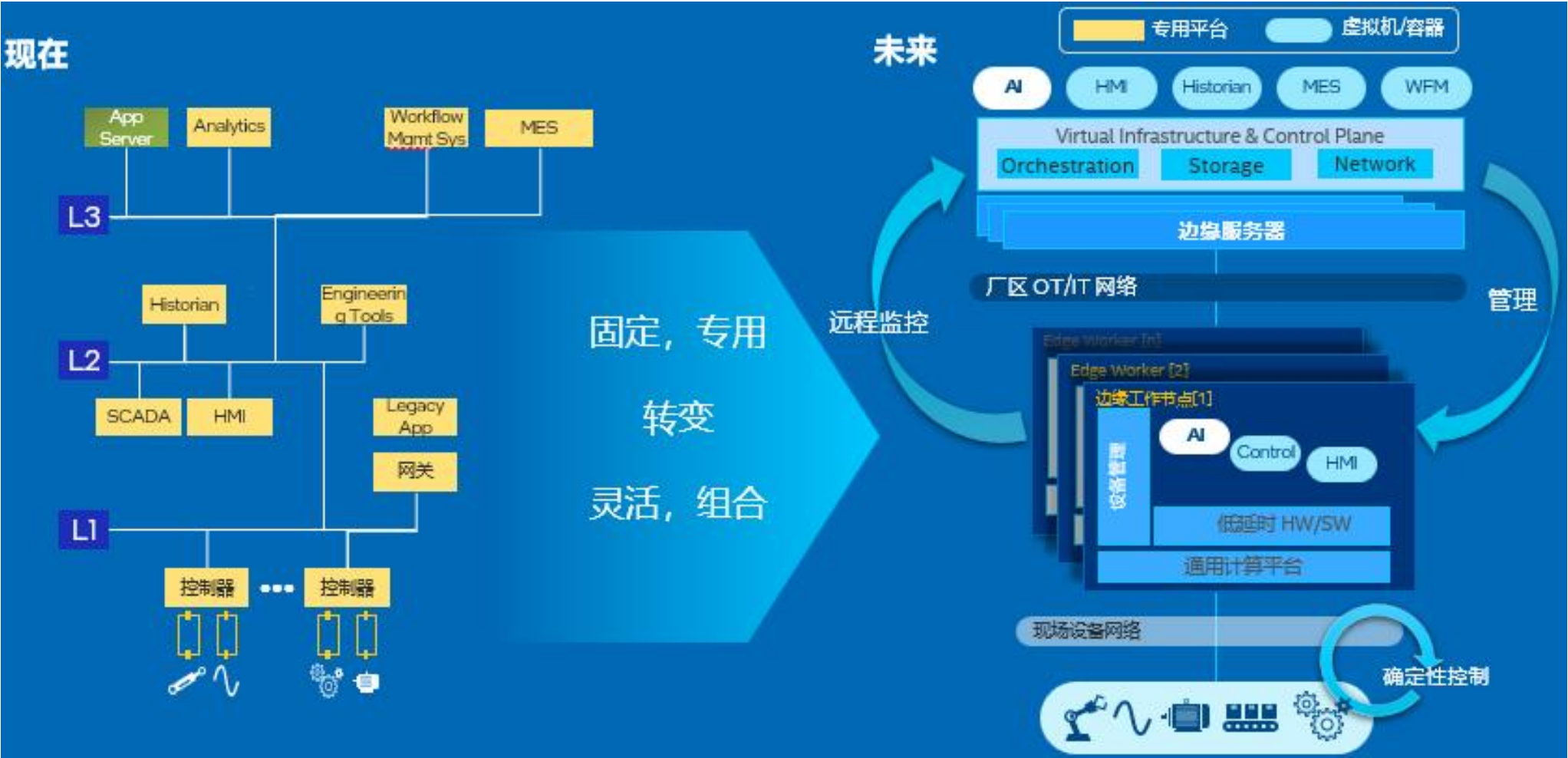
- 软件定义的工控系统
- *Linux KVM* 虚拟化方案
- 实时性能调优

软件定义的工控系统- 挑战



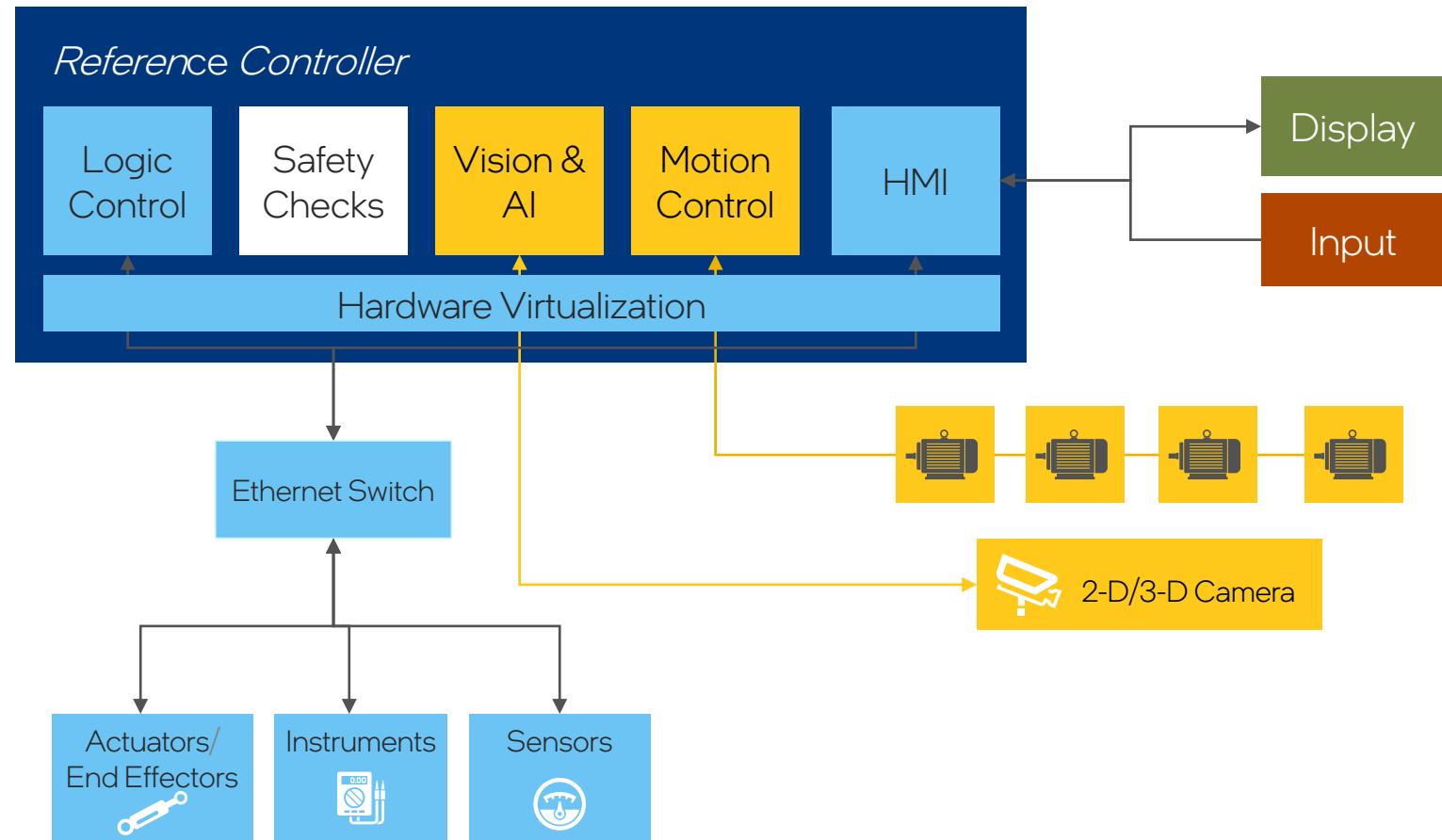
-  缺少具备互操作和可移植的软/硬件设备
-  固定功能，专用的设备
-  灵活性差，升级困难
-  软硬件设备更新导致的产线中断
-  硬件冗余方案价格昂贵
-  既有系统的培训周期长

软件定义的工控系统

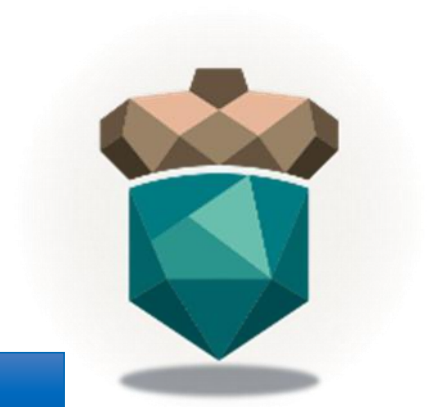


参考架构: 视觉, AI 辅助的控制

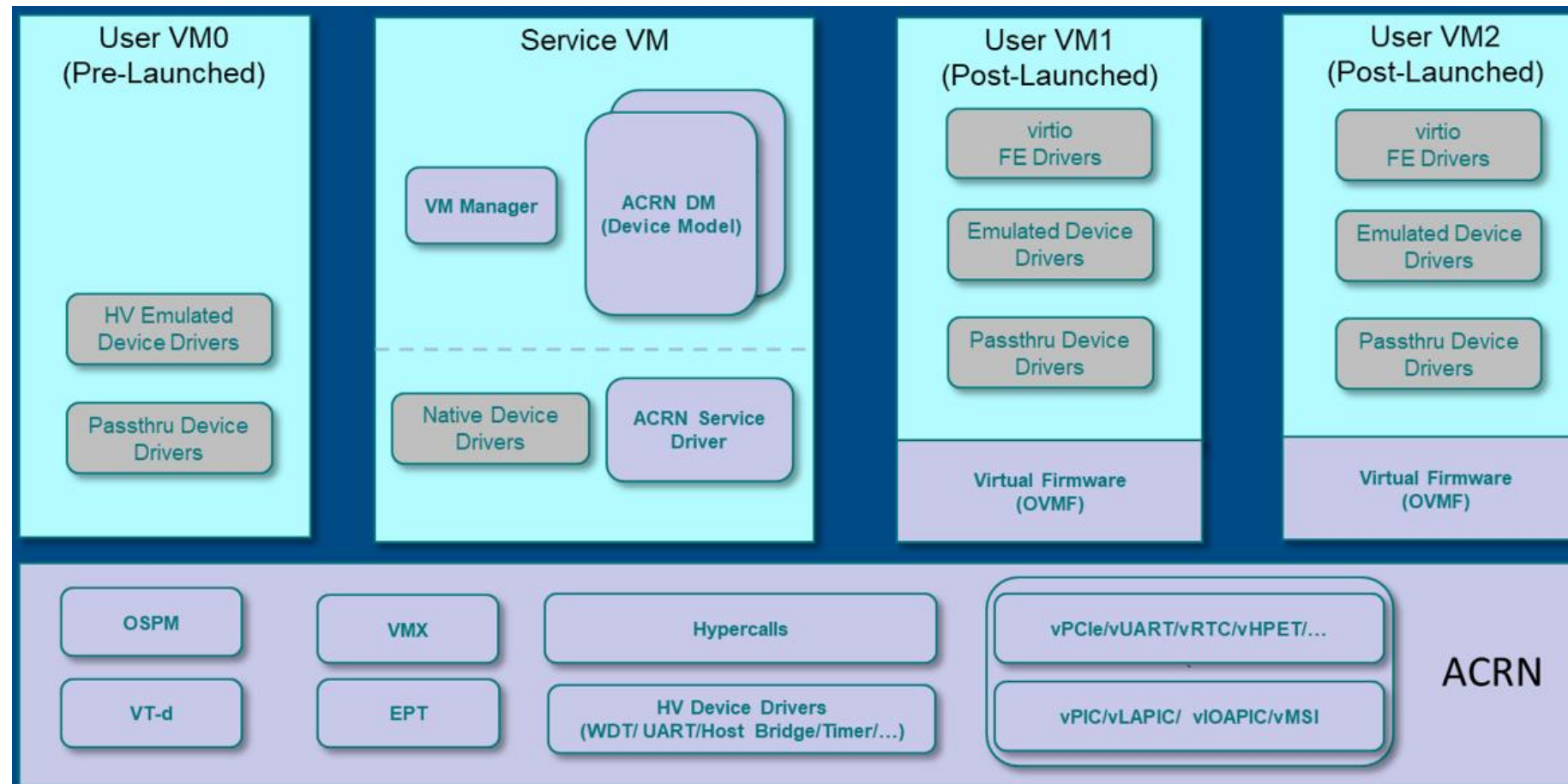
- 结合视觉/人工智能和实时控制
- 通过在单个CPU上整合视觉和控制来降低系统成本
- 视觉/人工智能支持多种应用场景, 包括检测和拣选放置
- 更快的部署和重新利用



回顾ACRN 虚拟化方案



LINUX FOUNDATION PROJECT。专为IOT设计的灵活的，完全开源的，轻量级的HV。





Linux KVM

虚拟化解决方案

- 适用于包含虚拟化扩展的Linux x86硬件的完整虚拟化解决方案
- 可加载内核模块
- 处理器特定模块

安全设计

每个虚拟机都有私有的虚拟化硬件：网卡、磁盘、图形适配器

灵活性

- 提供设备抽象，但不进行处理器仿真
- 使用SeaBIOS作为开源实现
- 支持热插拔vCPU、动态内存管理、实时迁移

图形化管理工具

Kimchi, Virtual Machine Manager, Proxmox Virtual Environment, OpenQRM, GNOME Boxes, oVirt

实现方式

硬件虚拟化

- *Allows you turn Linux into a hypervisor that allows a host machine to run multiple, isolated virtual environments*
- *Requires a processor with hardware virtualization extensions such as Intel-VT*

开源软件

- *The kernel component is included in mainline Linux*
- *The userspace component is included in mainline QEMU*

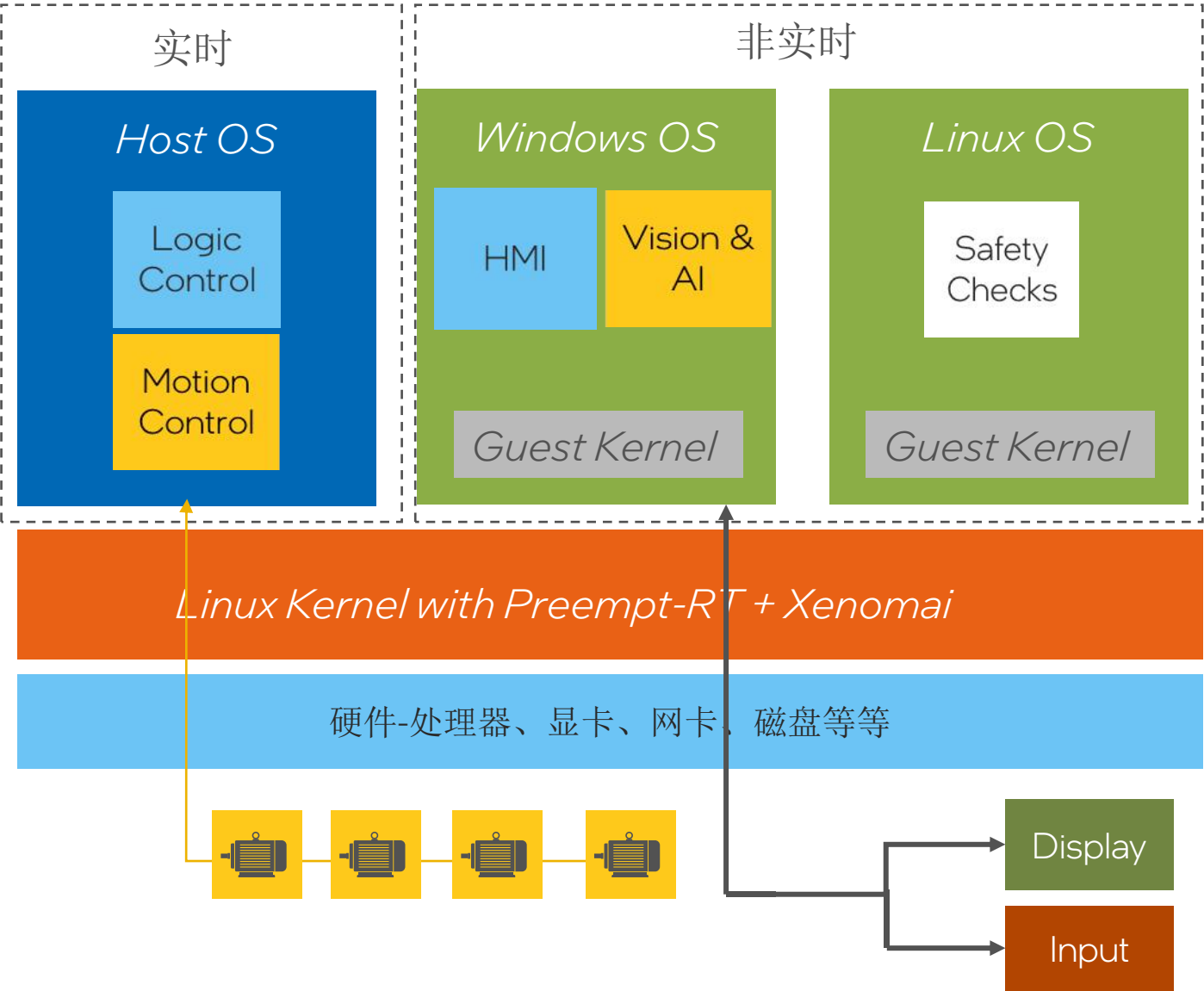
多操作系统

- *Run multiple virtual machines running unmodified Linux or Windows images*
- *Can be ported to other OS such as FreeBSD and Illumos*
- *Provides additional para virtualization support for Linux, BSD, Solaris, Windows, Haiku, ReactOS, Plan 9*

KVM – 硬件虚机化、灵活、功能丰富

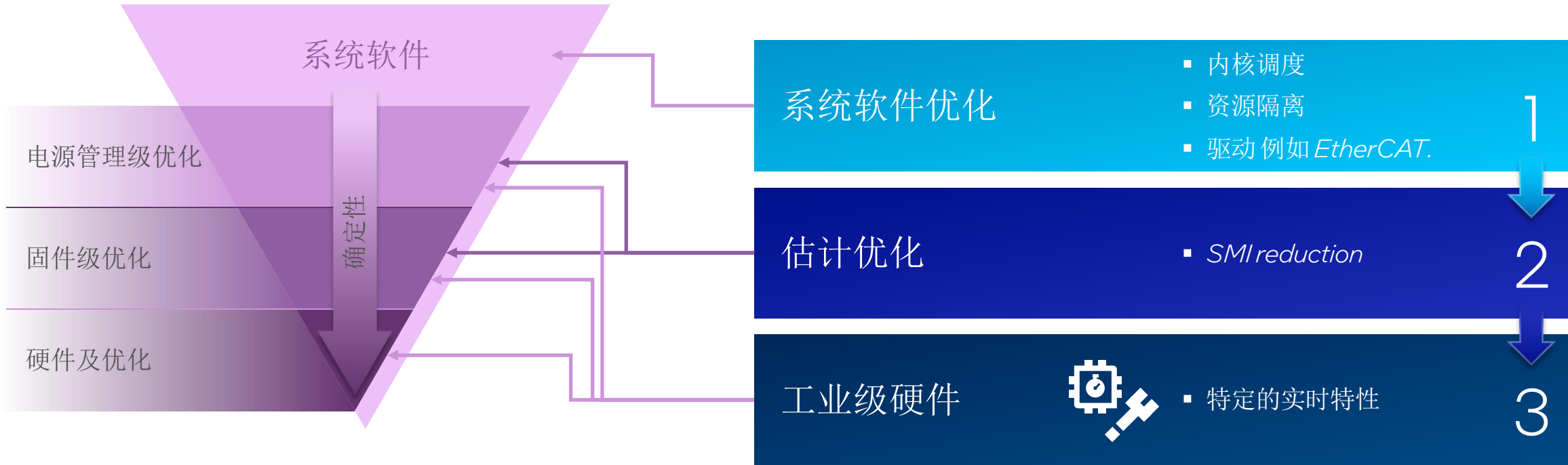
- 开源且具有成本效益。拥有庞大的开发者社区。
- 灵活性和多功能性。支持多种来宾操作系统（如*Linux*、*Windows*等）。与*Linux*的集成。
- 性能。利用硬件虚拟化扩展。
- 生态系统和工具。丰富的部署和管理工具。
- 安全性。

参考架构: 视觉, AI 辅助的控制



- Host OS 运行实时负载
- 外设透传
- 处理器绑定

实时性优化思路



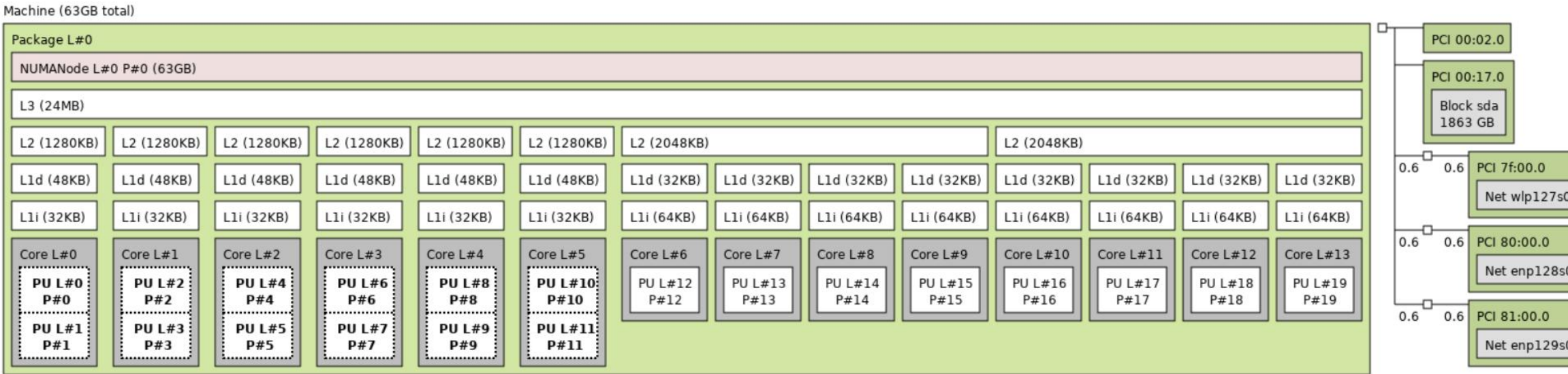
CPU 绑定 – 拓扑结构

- ✓ NUMA Awareness
- ✓ CPU/Core Complex Awareness

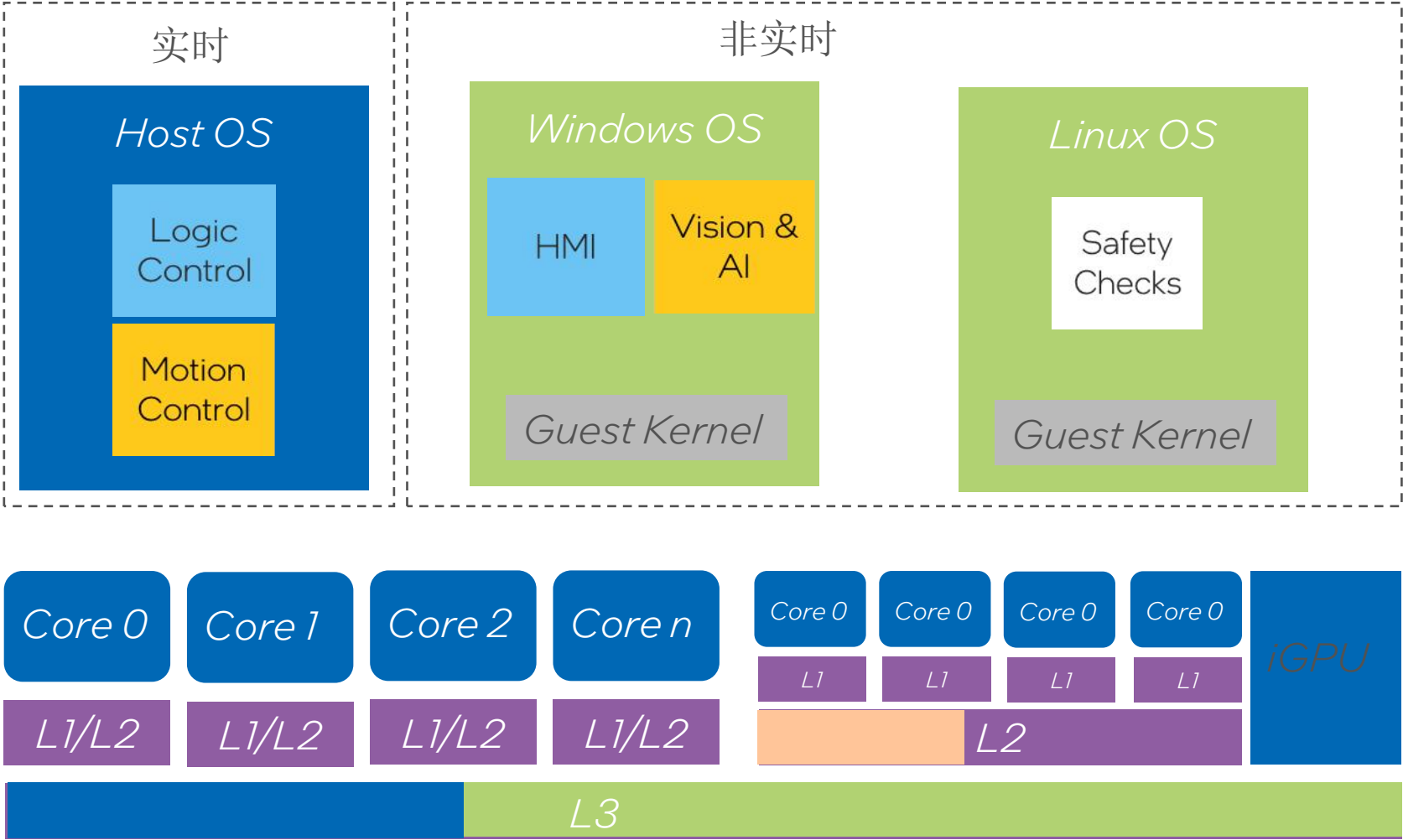
CPU Affinity in KVM

```
1 <vcpu placement='static'>4</vcpu>
2
3 <cputune>
4   <vcpupin vcpu='0' cpuset='0' />
5   <vcpupin vcpu='1' cpuset='1' />
```

CPU Topology (example via lstopo)



CPU Cache 分配



■ 更高的实时性能

```
BASH
1  #!/bin/bash
2
3  # Verify RDT support
4  echo "Verifying RDT support..."
5  sudo pqos -s
6
7  # Reset RDT configuration
8  echo "Resetting RDT configuration..."
9  sudo pqos -R
10
11 # Define cache allocation class (50% of LLC)
12 echo "Defining cache allocation class..."
13 sudo pqos -e "llc:1=0x0F"
14
15 # Associate core 1 with the cache allocation class
16 echo "Associating core 1 with cache allocation class..."
17 sudo pqos -a "llc:1=1"
18
19 # Verify configuration
20 echo "Verifying configuration..."
21 sudo pqos -s
```

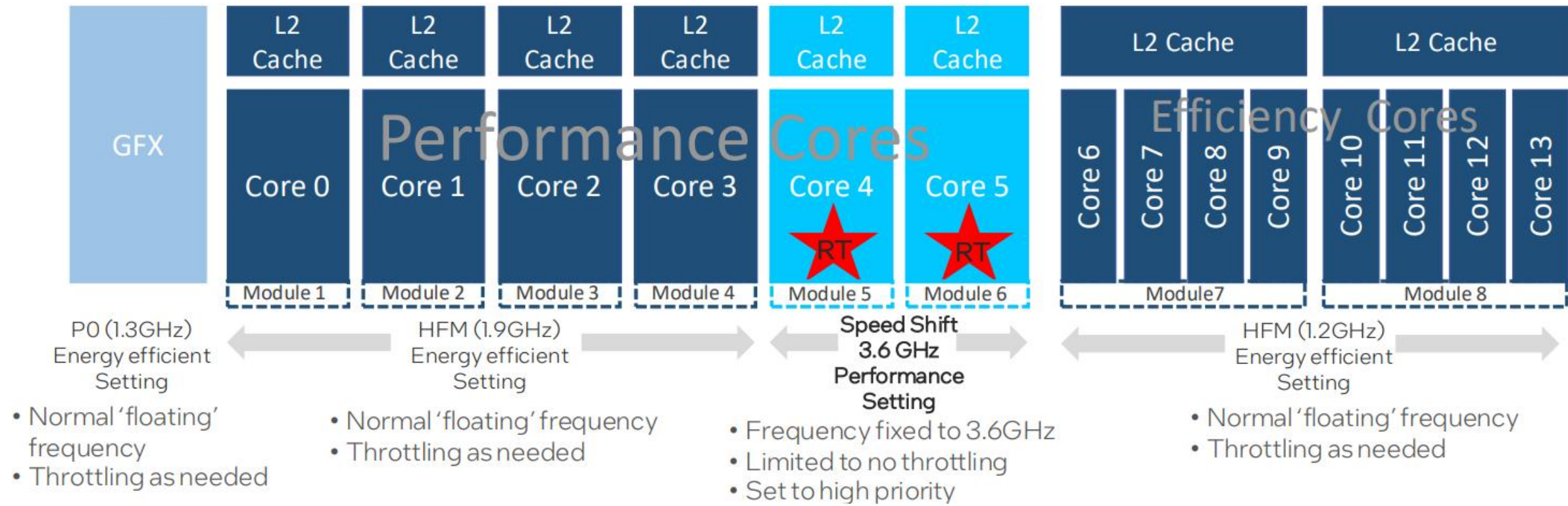
英特尔® Speed Shift 技术

How to optimize compute performance and balance power for mixed-criticality RT systems?

Fix CPU Frequency in BIOS		
Intel(R) SpeedStep	Disabled	Intel Advanced Menu → Power & Performance → CPU - Power Management Control
Turbo Mode	Disabled	Intel Advanced Menu → Power & Performance → CPU - Power Management Control

- *Why are Intel® Speed Shift and Turbo Boost technology not recommended for mixed-criticality RT designs?*
- *Increase of core count is directly reducing the advertised base frequency!*
- *We are worried about the true single-thread performance!*
- *How can we fine tune or balance power consumption to stay within TDP limits?*

示例 - 用于最大化单线程实时性能的配置



Notices and Disclaimers

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

The Intel logo is centered on a solid blue background. It features the word "intel" in a white, lowercase, sans-serif font. A small, light blue square is positioned above the first vertical stroke of the letter 'i'. To the right of the word "intel" is a small white registered trademark symbol (®).

intel®