

# openEuler LTS版本双内核支持

# 需求背景

## 第一，行业诉求

希望22.03 LTS SPx版本上，引入新内核来实现对新硬件支持，比如互联网行业，有明确诉求；

## 第二，业界实现

- **Ubuntu**: 通过持续发布新内核来获取新特性。

提供了[Hardware Enablement Stack](#)，又叫LTS Enablement Stack，是在LTS版本上提供的新版本内核和X、mesa等包以支持新的硬件平台。其规则为：每个LTS版本发布后，会收到之后4个Ubuntu版本的内核包作为其HWE内核。

以Ubuntu 20.04 LTS版本为例，其发布时的内核为5.4，之后会收到来自20.10 (5.8)，21.04 (5.11)，21.10 (5.13) 和22.04 LTS (5.14) 的内核包更新。类似的，Ubuntu 22.04 LTS发布时内核为5.15，在22.04.02 update版本中HWE内核升级到了来自22.10的5.19版本。并且Ubuntu在发布LTS版本的小幅更新版本的安装ISO时，会默认使用HWE内核，使得用户可以在新硬件上获得更好的安装体验。

- **Anolis OS 23**

Anolis OS 23 现支持 5.10/6.1 两种内核安装，5.10 为默认内核，内核 6.1 系列为 tech-preview 版本，支持滚动升级。

- **RHEL 7 / CentOS 7**: 只有一个内核大版本，通过不断backport来获取新特性

Red Hat官方并没有提供新版本内核机制，新硬件以及容器支持能力依然受到限制；

- **SUSE**

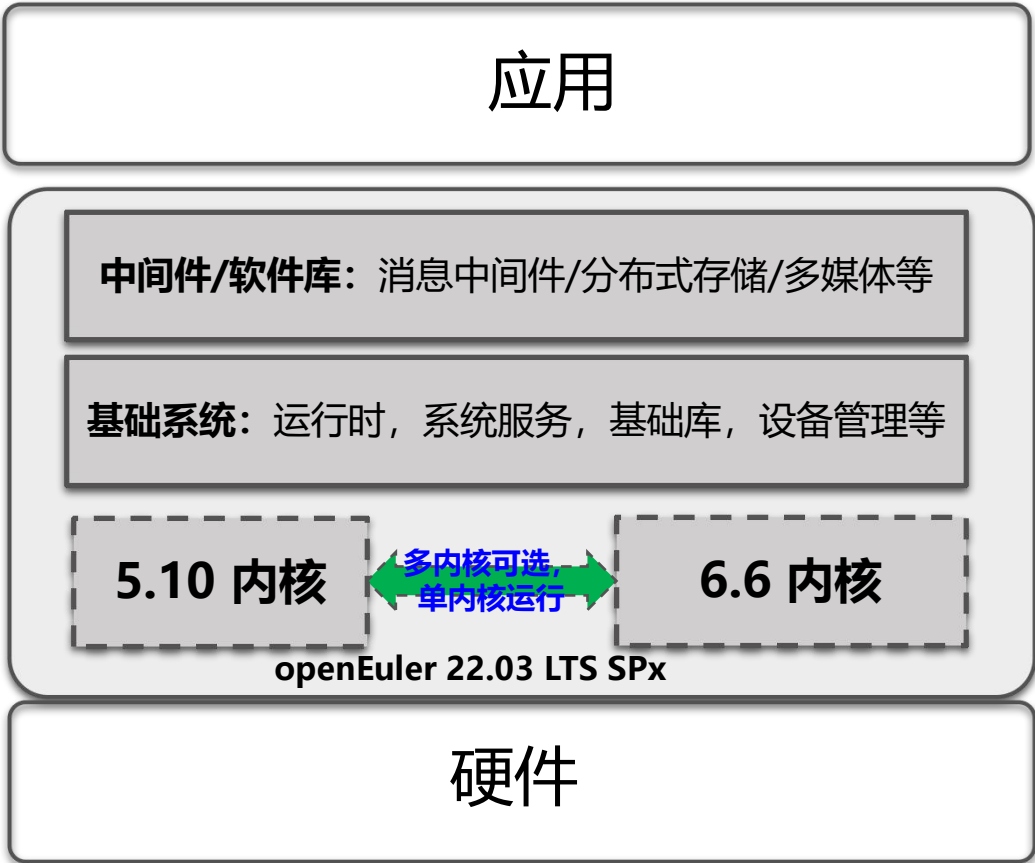
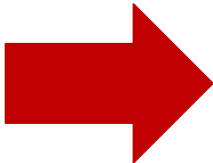
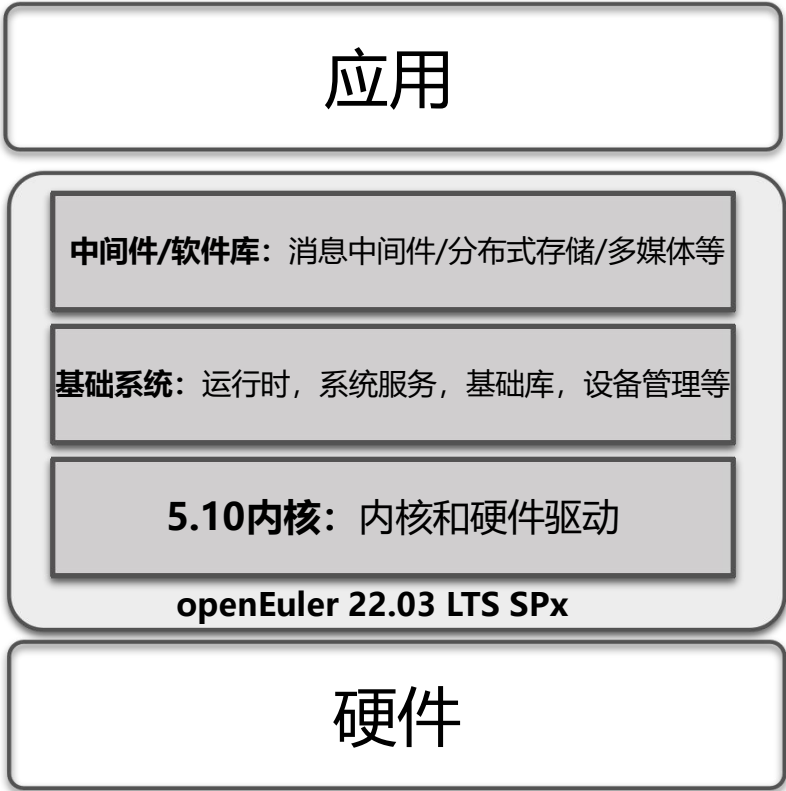
SUSE虽然不允许同时安装多个大版本的内核，但SUSE每隔一个Service Pack都会升级内核大版本，以提供新硬件和新特性支持。

例如：SLES 15 GA内核为4.12，15 SP1延续此版本；SP2则升级到5.3，SP3延续此版本；SP4升级到5.14，SP5延续此版本。

显著问题：没有为用户提供升级SP但保留旧版本内核继续使用的选项，每两年（两个SP）由于内核大版本升级造成树外模块的不兼容，对用户的升级决策有明显影响。

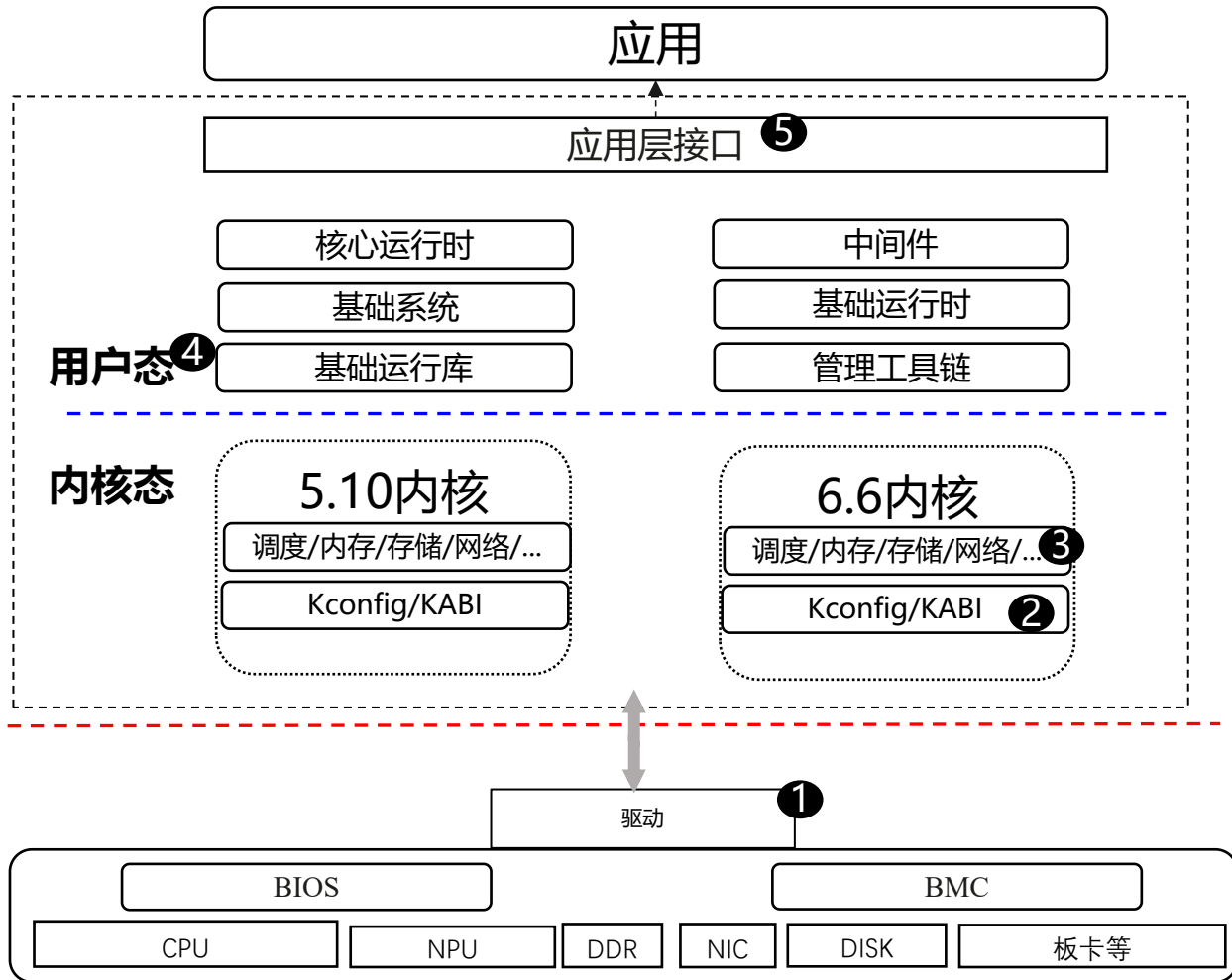
**openEuler多版本内核支持（社区）**：<https://gitee.com/openeuler/TC/blob/master/oEEP/oEEP-0011%20openEuler%E5%A4%9A%E7%89%88%E6%9C%AC%E5%86%85%E6%A0%B8%E6%94%AF%E6%8C%81.md>

# 期望效果



主内核+辅内核

# 影响分析



## ➤ 南向驱动

- 1, 南向肯定会引发兼容性问题 (非内核自带的硬件驱动风险较大、内核自带驱动风险可控);
- 2, 保持各自内核上的驱动兼容, out of tree驱动使用同一套工具链编译 (老版本) 进行编译;

## ➤ 内核KABI

内核KABI不兼容, 对硬件驱动的影响较大;

## ➤ 内核Kconfig

跟随内核, 内核特性根据需要, 配置不同kconfig开关, 影响内核特性, 对用户态影响较小, 有冲突的配置单独分析;

## ➤ 内核新特性

新内核需要明确具体的特性, 哪些特性是需要, 进行按需开启, 以及相关用户态软件包的适配;

## ➤ 北向软件包兼容性

- 1, 大部分基础库/软件包, 基本可以兼容5.10和6.6内核, 跟内核解耦;
- 2, **硬件/OS新特性**: 要单独进行case by case分析适配, 有2种方案:  
方案一, 高版本特性回到低版本, **优势**: 减低对上层应用适配风险, **劣势**: 工作量可能较大;  
方案二, 多版本支持, 对应用层存在适配风险;

## ➤ 应用层接口

- 1, 目标: ABI完全兼容, 应用程序不用改动, 就可以兼容运行;
- 2, 应用层针对新的内核可能不兼容, 例如iostate/top等, 需要在内核/proc里边获得信息, 新内核如果与老内核不兼容这个就需要修改系统应用层的东西进行适配;

## ➤ 性能影响、兼容性完整测试

待评测

# 方案细化

分类	变化点	传统单一内核架构	双内核架构方案	方案影响/工作量预估
核心工具链	编译器选型	5.10上GCC 编译器是10.3; 6.6上GCC 编译器是12.3;	GCC 编译器是10.3, 用老版本的编译器	如果有编译问题, 进行适配
南向驱动	out of tree驱动	内核KABI, 不同内核无法兼容;	方案一, 保持各自内核上的驱动兼容, out of tree驱动使用同一套工具链编译(老版本)进行编译; 方案二, 一个驱动适配2个内核; 方案三, 引导厂家, 做成核内驱动;	这部分驱动需专项识别, 并跟厂家沟通, 存在风险; 用户视角: 不感知使用哪些内核及配套驱动, 自动获取对应驱动版本;
内核特性	使能新硬件	Inbox驱动随内核编译; out of tree随内核演进; 硬件特性在内核使能;	out of tree随内核演进; 硬件特性在内核使能;	硬件特性使能按需合入
	内核新特性支持	6.6相比5.10, 新增xfs/ebpf/cgroup V2/Folio/EEVDF/IOMMUFD等特性,	5.10 内核 + 6.6 内核, 内核特性跟单一内核架构保持一致	新内核特性, 需要用户态的新特性支持的场景, 通过按需回合
	kconfig配置不一致	不同内核, kconfig配置不一致	跟随内核, 内核特性根据需要, 配置不同kconfig开关, 影响内核特性, 对用户态影响单独分析	有冲突的kconfig, 个别进行调整适配, 比如page大小
基础软件包/库	部分跟内核配套	lrpoute/ethtool跟内核版本相配, 需要升级为6.6版本;	部分基础库, 可能依赖新内核的特性, 要单独进行适配, 高版本特性回合到低版本/做多版本, 减低对上层应用适配风险	需专项分析, 工作量case by case单独评估
ABI兼容	应用层接口	跟基础软件包强强相关	用户态的软件包基本保持兼容, 确保对用户态ABI接口兼容	对应用层的ABI保持兼容
虚拟化	kvm	内核模块, 跟随内核演进	保持跟随内核演进	保持跟随内核演进
	qemu	6.6内核: qemu 8.2; 5.10内核: qemu 6.2; 5.10上的已全部继承到6.6	qemu跟kvm直接的配套; 方案: 1, 配套内核, 出2个qemu版本, 应用层兼容存在风险; 2, 高版本特性回合, 保持应用层兼容, 工作量较大;	做过简单验证, 基本功能可以兼容;
	libvirt	6.6内核: libvirt 9.10; 5.10内核: libvirt 6.2; 5.10上的已全部继承到6.6	libvirt跟qemu直接的配套; 方案: 1, 配套内核, 出2个版本, 应用层兼容存在风险, 如OpenStack; 2, 高版本特性回合, 保持应用层兼容, 工作量较大;	做过简单验证, 基本功能可以兼容;
	新特性支持	按需合入	按需合入	按需合入, 工作量单独评估
容器	基础功能	5.10和6.6, isula是同一套代码	22.03 LTS SP3上isula, 可以在6.6内核上正常运行, 基本功能验证过	有问题单独适配, 风险低
	新特性, cgroupV2/V1特性支持	内核支持, 即可使用	6.6内核默认是V2, 5.10和6.6, isula是同一套代码; Docker在2403 LTS 做了适配, 解决V2选型问题, 切回V1, 改动不大	内核具备特性, 容器就可以正常使用新特性, Docker场景需做少量适配
工程能力	构建, 用户使用方式	单一内核构建	1, ISO中自带1套用户态、2个内核, 用户选内核安装; 2, 2个ISO, 1套用户态+1个内核; 3, 现有ISO不变, 第二个内核放在repo仓, 定义个单独目录存放, 单独rpm下载, yaml安装;	EulerMaker需配套支持, 制作镜像要支持2个内核
测试	兼容性测试	单个内核+用户测试, 首版本发布后, 基本稳定, ~50人月	需增加老版本用户态软件包+新内核完整测试验证, 性能基线等	对性能基线存在影响, 需单独评估, ~100人月

# 待进一步交流的问题

## 第一，维护分工问题

增加双内核版本，维护工作量会增大，在双内核版本上维护和分工策略，待讨论；

## 第二，交付件问题

社区只做工程能力支持，还是要做正式双内核版本发布，后续如何维护和收编，待讨论；

## 第三，南向硬件驱动配套问题

整机厂家/板卡厂家提供驱动时，当前配套的是版本，后续如何配套/认证？

## 第四，性能

策略待明确，是否有主内核版本和辅内核版本的策略，性能怎么验收？



# THANKS