openEuler 23.03 技术白皮书

1. 概述

欧拉开源操作系统（openEuler，简称“欧拉”）从服务器操作系统正式升级为面向数字基础设施的操作系统，支持服务器、云计算、边缘计算、嵌入式等应用场景，支持多样性计算，致力于提供安全、稳定、易用的操作系统。通过为应用提供确定性保障能力，支持 OT 领域应用及 OT 与 ICT 的融合。

欧拉开源社区通过开放的社区形式与全球的开发者共同构建一个开放、多元和架构包容的软件生态体系，孵化支持多种处理器架构、覆盖数字基础设施全场景，推动企业数字基础设施软硬件、应用生态繁荣发展。

2019 年 12 月 31 日，面向数字基础设施的全场景开源操作系统开源社区 openEuler 正式成立。

2020 年 3 月 30 日， openEuler 20.03 -LTS （Long Term Support，简写为 LTS，中文为长生命周期支持）版本正式发布，为 Linux 世界带来一个全新的具备独立技术演进能力的 Linux 发行版。

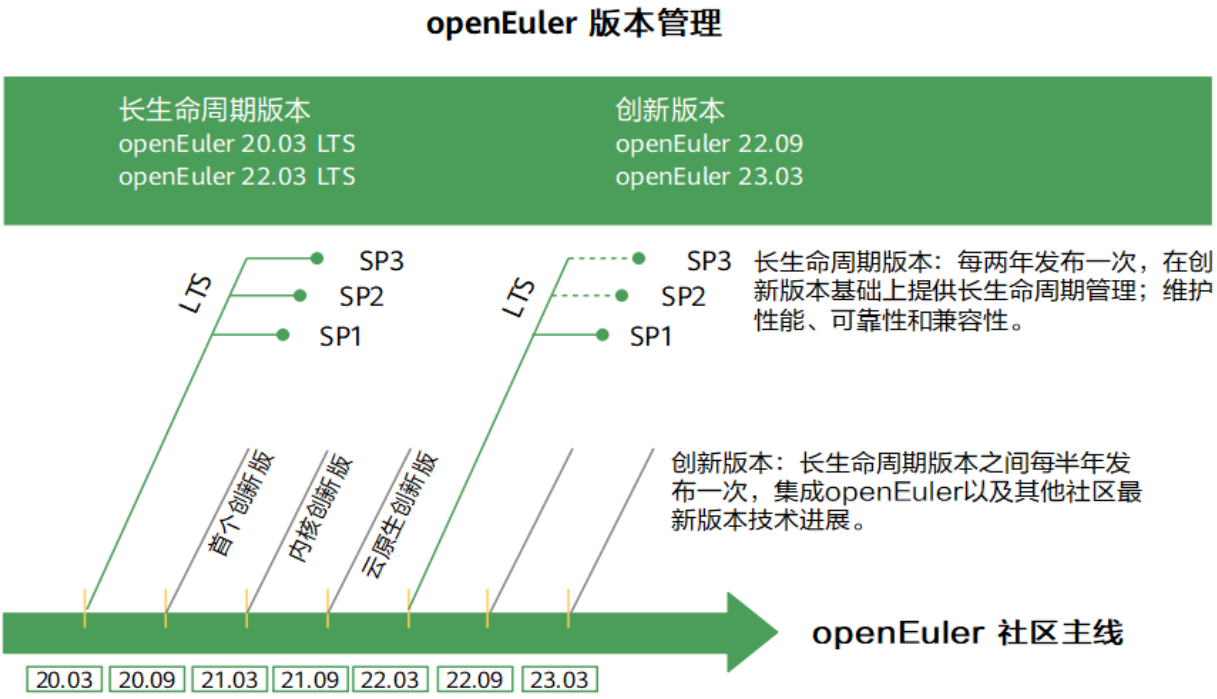
2020 年 9 月 30 日，首个 openEuler 20.09 创新版发布，该版本是 openEuler 社区中的多个企业、团队、独立开发者协同开发的成果，在 openEuler 社区的发展进程中具有里程碑式的意义，也是中国开源历史上的标志性事件。

2021 年 3 月 31 日，发布 openEuler 21.03 内核创新版，该版本将内核升级到 5.10，还在内核方向实现内核热升级、内存分级扩展等多个创新特性，加速提升多核性能，构筑千核运算能力。

2021 年 9 月 30 日，全新 openEuler 21.09 创新版如期而至，这是欧拉全新发布后的第一个社区版本，实现了全场景支持。增强服务器和云计算的特性，发布面向云原生的业务混部 CPU 调度算法、容器化操作系统 KubeOS 等关键技术；同时发布边缘和嵌入式版本。

2022 年 3 月 30 日，基于统一的 5.10 内核，发布面向服务器、云计算、边缘计算、嵌入式的全场景 openEuler 22.03-LTS 版本，聚焦算力释放，持续提升资源利用率，打造全场景协同的数字基础设施操作系统。

2023 年 3 月 30 日，发布 openEuler 23.03 内核创新版，采用 Linux Kernel 6.1 内核，为未来 openEuler 长生命周期版本采用 6.x 内核提前进行技术探索，方便开发者进行硬件适配、基础技术创新和上层应用创新。

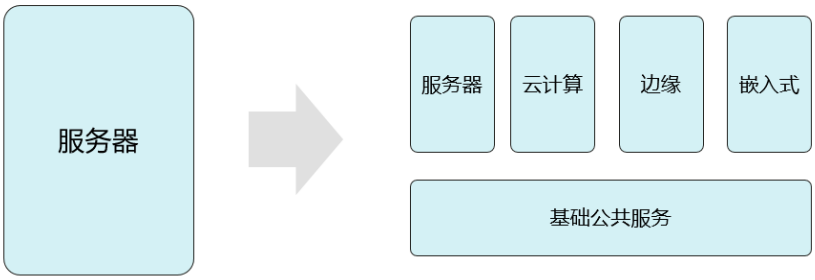


openEuler 作为一个操作系统发行版平台，每两年推出一个 LTS 版本。该版本为企业级用户提供一个安全稳定可靠的操作系统。

openEuler 也是一个技术孵化器。通过每半年发布一次的创新版，快速集成 openEuler 以及其他社区的最新技术成果， 将社区验证成熟的特性逐步回合到发行版中。这些新特性以单个开源项目的方式存在于社区，方便开发者获得源代码，也方便其他开源社区使用。

社区中的最新技术成果持续合入发行版，发行版通过用户反馈反哺技术，激发社区创新活力，从而不断孵化新技术。发行版平台和技术孵化器互相促进、互相推动、牵引版本持续演进。

openEuler 覆盖全场景的创新平台

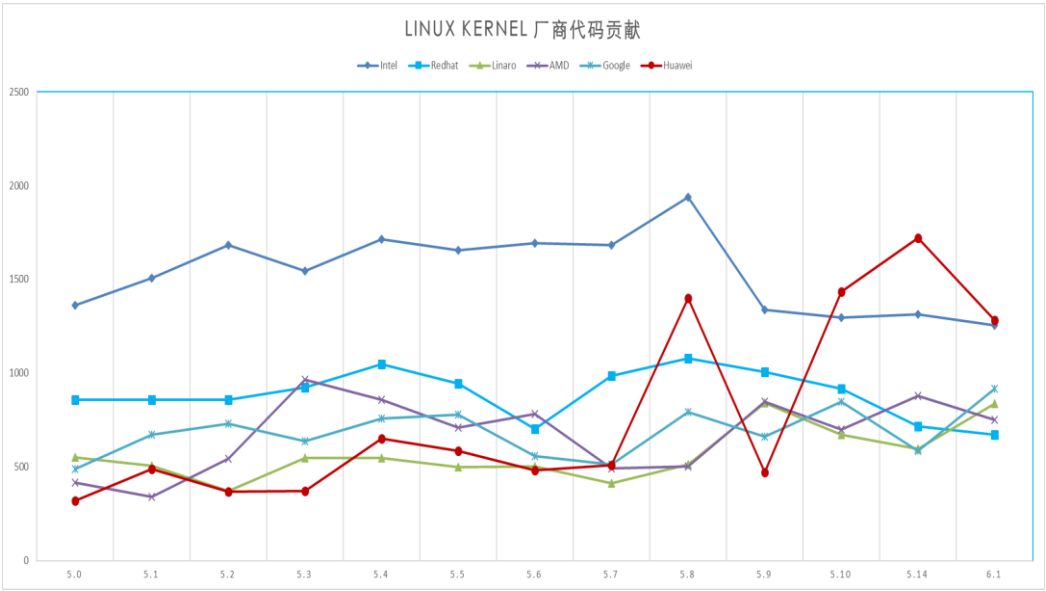


openEuler 23.03 创新版本已支持 x86、Arm 多处理器架构，未来还会基于 LTS 版本扩展 SW64/LoongArch/RISC-V/PowerPC 等更多芯片架构支持，持续完善多样性算力生态体验。

openEuler 社区面向场景化的 SIG 不断组建，推动 openEuler 应用边界从最初的服务器场景，逐步拓展到云计算、边缘计算、嵌入式等更多场景。openEuler 正成为覆盖数字基础设施全场景的操作系统，新增发布面向边缘计算的版本 openEuler 23.03 Edge、面向嵌入式的版本 openEuler 23.03 Embedded。

openEuler 希望与广大生态伙伴、用户、开发者一起，通过联合创新、社区共建，不断增强场景化能力，最终实现统一操作系统支持多设备，应用一次开发覆盖全场景。

openEuler 对 Linux Kernel 的持续贡献



openEuler 内核研发团队持续贡献 Linux Kernel 上游社区，回馈主要集中在：芯片架构、ACPI、内存管理、文件系统、 Media、内核文档、针对整个内核质量加固的 bug fix 及代码

重构等内容。

openEuler 开放透明的开源软件供应链管理

开源操作系统的构建过程，也是供应链聚合优化的过程。拥有可靠开源软件供应链，是大规模商用操作系统的基础。openEuler 从用户场景出发，回溯梳理相应的软件依赖关系，理清所有软件包的上游社区地址、源码和上游对应验证。完成构建验证、分发、实现生命周期管理。开源软件的构建、运行依赖关系和上游社区，三者之前形成闭环且完整透明的软件供应链管理。

2. 平台架构

系统框架

openEuler 是覆盖全场景的创新平台，在引领内核创新、夯实云化基座的基础上，面向计算架构互联总线、存储介质发展新趋势，创新分布式、实时加速引擎和基础服务，结合边缘、嵌入式领域竞争力探索，打造全场景协同的面向数字基础设施的开源操作系统。

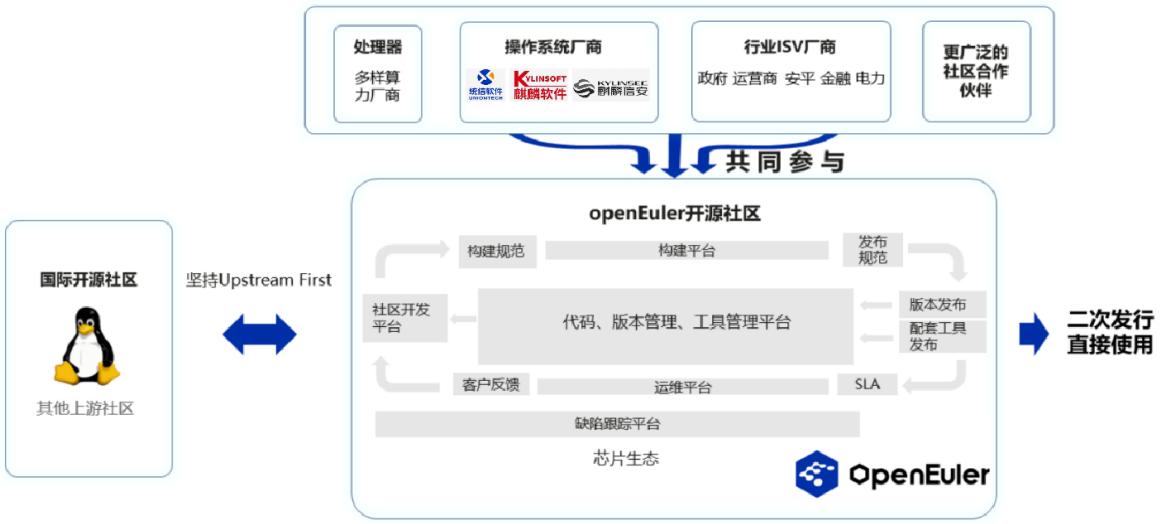
面向未来，社区将持续创新、社区共建、繁荣生态，夯实数字基座。

繁荣社区生态：

- 友好桌面环境：UKUI、DDE 、Xfce、Kiran-desktop、GNOME 桌面环境，丰富社区桌面环境生态。
- 欧拉 DevKit：支持操作系统迁移、兼容性评估、简化安全配置 secPaver 等更多开发工具。

平台框架

openEuler 社区与上下游生态建立连接，构建多样性的社区合作伙伴和协作模式，共同推进版本演进。



硬件支持

openEuler 23.03 创新版本基于 x86/Arm 两个平台进行验证，验证结果如下。

-机型：TaiShan200 2280、2288H V6

-部件芯片：I350、SAS3408、1822、Intel X710

-板卡：SP210、SR150-M、SP580、SP330

更广泛的兼容性欢迎硬件厂家一起参与，相关流程：

详细的列表可在支持网站查询：https://www.openeuler.org/zh/compatibility/。

3. 运行环境

服务器

若需要在物理机环境上安装 openEuler 操作系统，则物理机硬件需要满足以下兼容性和最小硬件要求。

硬件兼容支持请查看 openEuler 兼容性列表：https://openeuler.org/zh/compatibility/

部件名称	最小硬件要求
架构	Arm64、x86_64
内存	为了获得更好的体验，建议不小于 4GB
硬盘	为了获得更好的体验，建议不小于 20 GB

虚拟机

openEuler 安装时，应注意虚拟机的兼容性问题。

部件名称	最小虚拟化空间要求
架构	Arm64、x86_64
CPU	2 个 CPU
内存	为了获得更好的体验，建议不小于 4GB
硬盘	为了获得更好的体验，建议不小于 20 GB

边缘设备

若需要在边缘设备环境上安装 openEuler 操作系统，则边缘设备硬件需要满足以下兼容性和最小硬件要求。

部件名称	最小硬件要求
架构	Arm64、x86_64
内存	为了获得更好的体验，建议不小于 4GB
硬盘	为了获得更好的体验，建议不小于 20 GB

嵌入式

若需要在嵌入式环境上安装 openEuler 操作系统，则嵌入式硬件需要满足以下兼容性和最小硬件要求。

部件名称	最小硬件要求
架构	Arm64、Arm32
内存	为了获得更好的体验，建议不小于 512MB
存储	为了获得更好的体验，建议不小于 256MB

4. 场景创新

云原生高性能服务网格数据面（Kmesh）

随着 AI/直播等大应用的发展及传统应用云化改造的深入，数据中心集群规模越来越大、应用类型也越来越丰富，如何实现集群内服务间的高效互通、满足应用 SLA 诉求已成为数据中心面临的关键问题，对云基础设施提出了很高的要求。

基于 k8s 的云基础设施能够帮助应用实现敏捷的部署管理，但在应用流量编排方面有所欠缺，服务网格的出现很好的弥补了 k8s 流量编排的缺陷，与 k8s 互补，真正实现敏捷的云应用开发运维；但随着对服务网格应用的逐步深入，当前服务网格的代理架构，数据面引入了额外的时延底噪开销，已成为业界共识的性能问题：

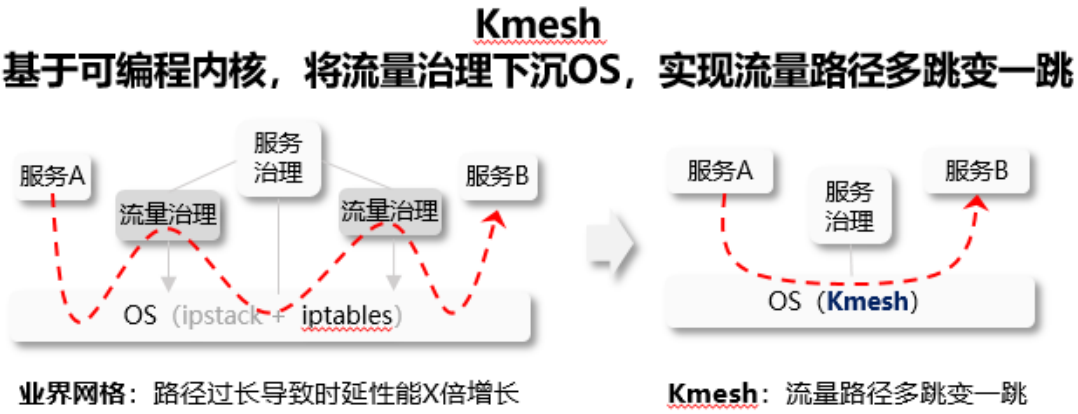
* 时延

以服务网格典型软件 istio 为例，网格化后，服务访问单跳时延增加 2.65ms；无法满足时延敏感型应用诉求。

* 底噪

istio 中，每个 sidecar 软件占用内存 50M+，CPU 默认独占 2 core，对于大规模集群底噪开销太大，降低了业务容器的部署密度。

Kmesh 基于可编程内核，将服务治理下沉 OS，实现高性能服务网格数据面，服务间通信时延对比业界方案提升 5 倍。



功能描述

- 支持对接遵从 XDS 协议的网格控制面（如 istio）
- 流量编排能力

  -负载均衡：支持轮询等负载均衡策略

  -路由：支持 L7 路由规则

  -灰度：支持百分比灰度方式选择后端服务策略

应用场景

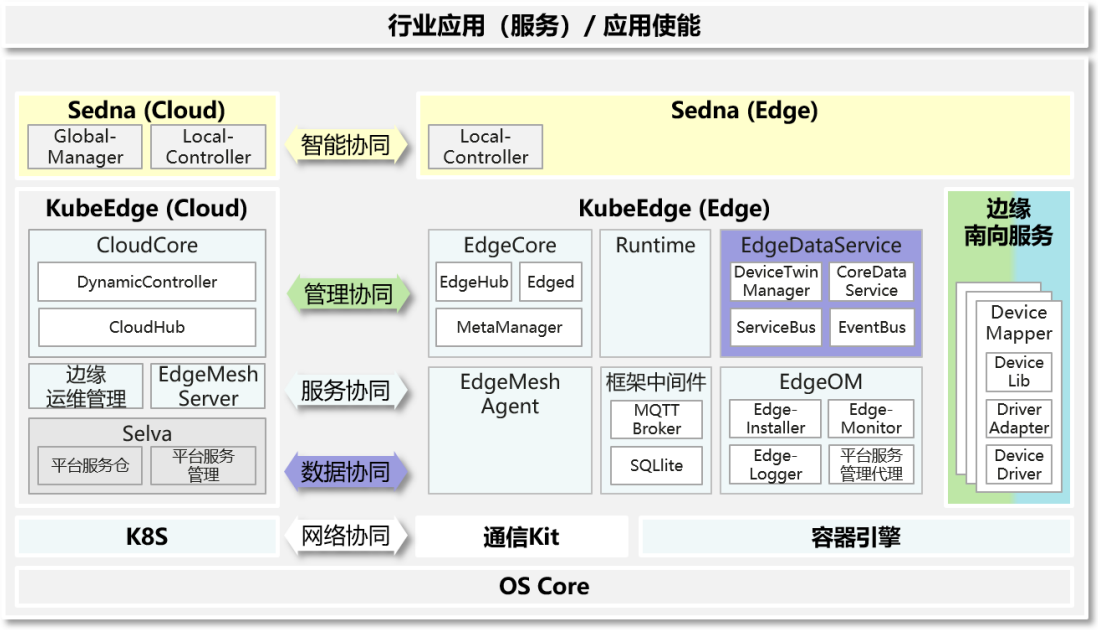
服务网格场景：优化云原生服务网格内服务通信性能。例如电商、计费、金融、物流、短视频、在线会议、云游戏等对服务时延敏感的应用。

边缘计算

边缘计算是未来 10 大战略技术趋势。随着智慧城市、自动驾驶、工业互联网等应用落地，海量数据将在边缘产生，IDC 预测中国 2025 年每年产生的数据将达 48.6ZB，集中式云计算在带宽负载、网络延时、数据管理成本等方面愈发显得捉襟见肘，难以适应数据频繁交互需求，边缘计算价值凸显。

openEuler 发布面向边缘计算的版本 openEuler 23.03 Edge，集成 KubeEdge+边云协同框架，具备边云应用统一管理和发放等基础能力，并将通过增强智能协同提升 AI 易用性和场景适应性，增强服务协同实现跨边云服务发现和流量转发，以及增强数据协同提升南向服务能力。

功能描述



版本功能如下：

1) 提供统一的跨边云的协同框架（KubeEdge+），实现边云之间的应用管理与部署，跨边云的通信，以及跨边云的南向外设管理等基础能力。

未来还将提供：

1) 边云服务协同：边侧部署EdgeMesh Agent，云侧部署EdgeMesh Server实现跨边云服务发现和服务路由。
2) 完善边缘南向服务：南向接入Mapper，提供外设Pofile及解析机制，以及实现对不同南向外设的管理、控制、业务流的接入，可兼容EdgeX Foundry开源生态。
3) 边缘数据服务：通过边缘数据服务实现消息、数据、媒体流的按需持久化，并具备数据分析和数据导出的能力。
4) 边云智能协同架构（Sedna）：基于开源sedna框架，提供基础的边云协同推理、联邦学习、增量学习等能力，并实现了基础的模型管理、数据集管理等，使能开发者快速开发边云AI协同特性，以及提升用户边云AI特性的训练与部署效率。

应用场景

可应用智能制造、城市交通、高速收费稽查、智慧加油站、医疗影像识别、智慧园区等

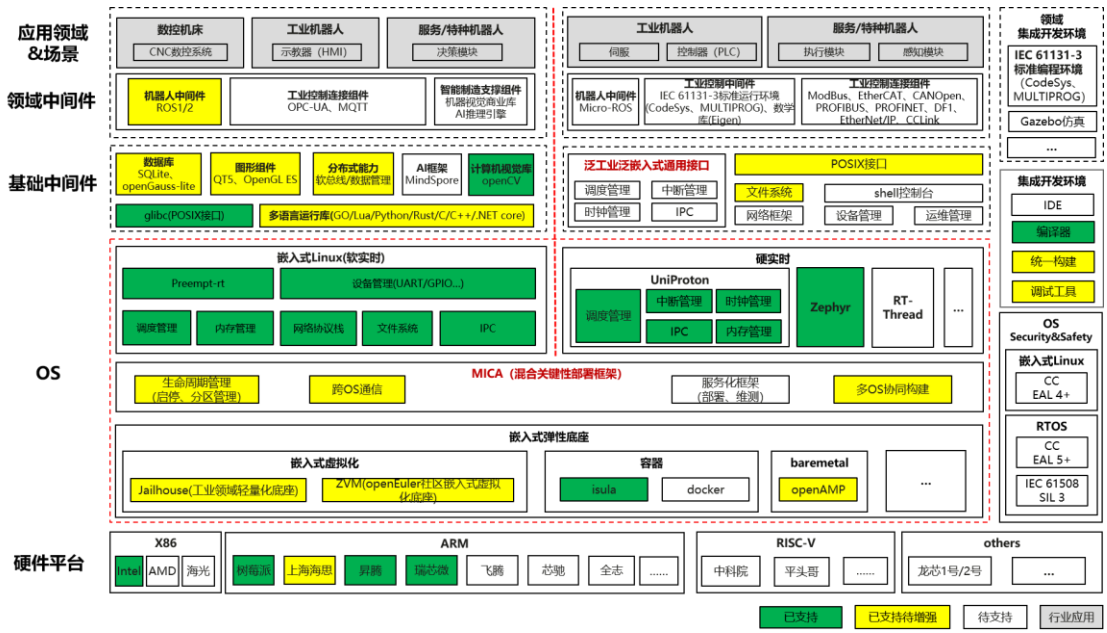
广泛的边云协同场景。

嵌入式

在中国制造 2025 及工业化和信息化融合进程加快的大背景下，我国工业软件以及信息化服务的需求持续增加，嵌入式软件作为工业软件行业最大的细分产品，其市场份额占比达到 57.4%，发展日渐壮大。

openEuler Embedded 围绕工业和机器人领域持续深耕，通过行业项目垂直打通，不断完善和丰富嵌入式技术栈和生态。openEuler 23.03 Embedded（基于 5.10 内核）首次支持嵌入式虚拟化弹性底座，提供 Jailhouse 和 ZVM 虚拟化方案、openAMP 轻量化混合部署三种方案，用户可以根据自己的使用场景选择最优的部署方案。同时首次支持 ROS2，集成 ros-core、ros-base、SLAM 等核心软件包，满足 ROS2 运行时要求。未来 openEuler Embedded 将协同 openEuler 社区生态伙伴、用户、开发者，逐步扩展支持 RISC-V、龙芯等芯片架构，丰富工业中间件、ROS 中间件、仿真系统等能力，打造嵌入式领域操作系统解决方案。

功能描述



版本功能如下：

1） 轻量化能力：开放 yocto 小型化构建裁剪框架，支撑 OS 镜像轻量化定制，提供 OS 镜像<5M，以及<5s 快速启动等能力。

2） 多硬件支持：支持树莓派、x86、Hi3093、RK3568 作为嵌入式场景通用硬件。

3） 软实时内核：基于 linux 5.10 内核提供软实时能力，软实时中断响应时延微秒级。

4） 嵌入式弹性虚拟化底座：提供多种虚拟化方案，满足用户不同硬件和业务场景需要：

a) baremetal：基于 openAMP 实现裸金属混合部署方案，支持外设分区管理，性能最好，但隔离性和灵活性较差。目前支持 UniProton/Zephyr/RT-Thread）和 openEuler 嵌入式 linux 混合部署。

b) 分区虚拟化：基于 Jailhouse 实现工业级硬件分区虚拟化方案，性能和隔离性较好，但灵活性较差。目前支持 FreeRTOS 和 openEuler 嵌入式 linux 混合部署。

c) 实时虚拟化：openEuler 社区自研虚拟化方案，兼顾性能、隔离性和灵活性，综合最优。目前支持 Zephyr 和 openEuler 嵌入式 linux 混合部署。

5） 嵌入式软件包支持：140+嵌入式领域常用软件包的构建。

6） 硬实时内核（UniProton）：支持 POSIX 接口（75 个），上下文切换时延 3us、中断延迟 2us。

未来还将提供：

a) 南向生态：RISC-V、龙芯支持。

b) 混合关键性部署框架：围绕生命周期管理、跨 OS 通信、服务化框架、多 OS 协同构建 4 个方面持续打造标准混部框架，支持更多的软实时和硬实时 OS 接入。

c) 嵌入式弹性底座：持续完善 Jailhouse 和 ZVM 虚拟化能力，支持更广泛的南向生态，提供更好的时延优化。

d) 硬实时（UniProton）中间件：提供丰富的 POSIX 接口支持和常用中间件，方便用户应用开发和迁移。

e) 泛工业泛嵌入式通用接口：围绕 RTOS 领域极致性能场景，定义高性能 API，为北向应用提供统一的接口。

f) 行业安全认证：联合伙伴逐步支持面向行业安全认证，如面向 IEC61508、CC EAL 等。

应用场景

嵌入式系统可广泛应用于工业控制、机器人控制、电力控制、航空航天、汽车及医疗等领域。

openGemini 时序数据库

openGemini 是专为物联网和运维监控等场景设计的一款云原生分布式时序数据库，致力于提供云原生、高性能、高扩展、低存储成本、存储分析一体化的用户体验，帮助用户降低系统维护成本，提高生产效率。

openGemini 通过 MPP 大规模并行处理分层架构，采用了多项创新优化技术，比如向量化、流计算、多级数据降采样、LSM 优化、AI4DB、数据压缩等，以应对可观测性、AIOps、IoT 等领域超大规模时间线和海量时序数据给数据库带来性能和存储成本的巨大挑战，打造面向物联网、运维监控和数据存储分析的一体化平台。

功能描述

版本功能如下：

1）分布式：采用 MPP 大规模并行处理分层架构，支持水平扩展。

2）安全：支持数据传输加密和用户密码鉴权，支持用户弱密码校验和审计日志。此外，openGemini 集群的各组件之间通信可配置 HTTPS 双向认证（Mutual TLS），确保每一个链接都是可信的。

3）多平台：支持 openEuler、Ubuntu 等多种 Linux 发型版本，CPU 架构支持鲲鹏（Arm64）和 x86。

4） 多开发语言：支持常见的主流开发语言，如 C/C++，C#，Java，Python，GO，Rust，JavaScript 等。

5）多形态部署：支持容器、虚拟机、物理服务器部署，可云、可线下、可边缘等多种部署形态。

6）丰富的分析算子：提供了丰富的聚合算子（如 COUNT、SUM、MAX 等）、统计分析算子（如 PERCENTILE 分位数、DIFFERENCE 等）、算术分析算子（如 ABS、LN 等）、Full Join、近似统计算子以及字符串算子（如 SUBSTR、STR 等）共 60+个。此外，还支持秒、毫秒、纳秒等多种时间精度以及 TAG（分组查询）和嵌套查询。

7）数据压缩：采用列式数据存储，不同数据类型对应不同的数据压缩算法，可支持 PB 级指标数据的长期数据存储。存储成本是传统关系型数据库 1/20，是 NoSQL 的 1/10。

8）数据保留策略：自定义库粒度数据保留策略，数据过期后自动删除。

9）写前日志（WAL）：设备掉电后，缓存数据不丢失。

10）流计算：当数据量较大时，传统降采样工作方式对磁盘 I/O 消耗过大，I/O 放大也很严重。可通过流式计算，写入数据的同时实现数据降采样，该方法具有高性能、网络开销小的优点。

11）数据分级存储：结合时序数据特点，数据按热、温、冷多级存储，进一步提升查询性能。

12）内核运行状态可观测：提供了开源监控工具 ts-monitor，可采集 260 余种内核和服务器关键监控指标，更好的观察系统的运行状况，快速排查、定位和解决问题。

13）多级降采样：可针对不同时间范围的历史数据进行不同方式的降采样，保留历史数据的特征，原地删除其余历史数据，节约存储空间 50%，计算资源节约 90%。

14）向量化：充分利用架构的并行处理优势，每次迭代批量返回数据，大数据量下查询性能更好。

15）异常检测和预测：内置基于 AI 的时序异常检测和预测分析框架，具备流批一体、严重程度分级等多项能力，支持每秒万级实时指标数据并发检测，内含 13 种异常检测器，可覆盖常见的离群点、数值变化、阈值、持续上升下降等常见时序异常场景。

未来还将提供：

a）高可靠性：支持数据多副本。

b）可观测性：支持 openTelemetry 的 OLTP 标准协议。

c）高时间序列基数：不再受限于时间序列规模。

d）标准 SQL：支持标准 SQL，降低用户学习成本。

e）软硬件结合：将 DPU 技术与 openGemini 相结合，依托 openEuler，打造极致性能的时序数据库。

应用场景

openGemini 时序数据库可广泛应用于电力、交通、电信、医疗健康、航空航天、互联网、智能家居、智慧园区、能源、运维监控等领域。

5. 内核创新

openEuler 内核中的新特性

openEuler 23.03 基于 Linux Kernel 6.1 构建，集成了 Linux Kernel 6.1 的众多新特性，比如：

a) CFS burstable 带宽控制器：通过 cgroups 引入 burstable CFS 控制器，允许 burst CPU 绑定的工作负载借用未来的配额，以改善整体延迟和批处理性能。
b) SCHED_IDLE 的 cgroup 支持：支持将整个 cgroup 中的所有任务放到 task scheduler 的 SCHED_IDLE 类下，使这些任务只在没有其他任务等待运行的情况下才运行。
c) 调度优化：改善唤醒延时；改善 NUMA 不平衡行为；优化/调整 select_idle_cpu()，以减少在过载系统上搜索空闲 CPU 的时间。
d) Memory tiering 优化：改进内存管理子系统，首先引入一种新算法，有助于识别 NUMA 节点中哪些页面是热的，以便系统可以将热/冷页面提升/降级到适当的 NUMA 节点。另外使分层信息可用于用户空间，并允许用户配置它。
e) BPF CO-RE 支持：支持 BPF 程序一次编译，到处运行，解决 BPF 程序移植难的问题。

f) XFS 可扩展性提升：通过删除自旋锁和全局同步点，改进了 XFS 日志可扩展性；另外，提升了缓冲区缓存的无锁查找性能。

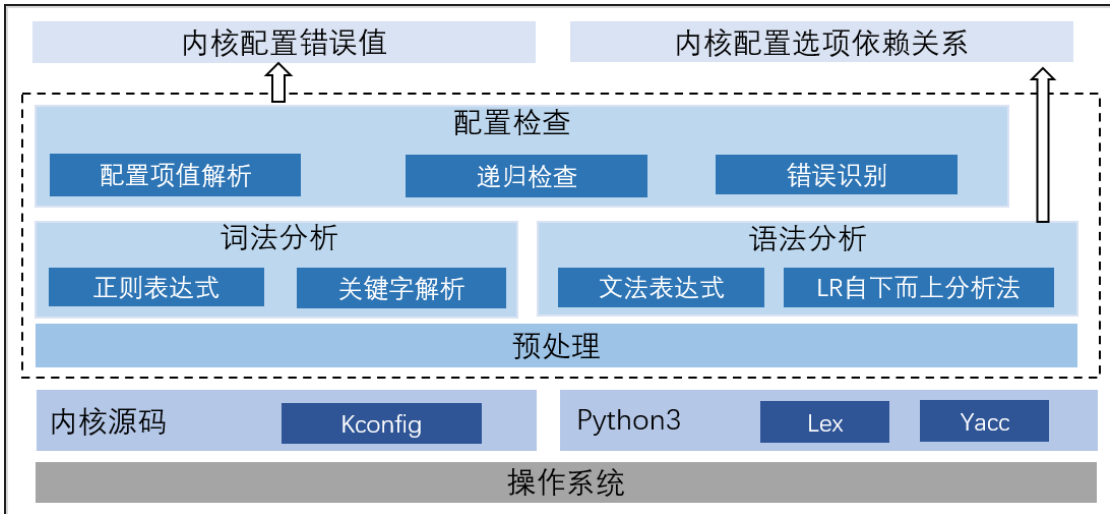
g) IO_URING 优化：异步缓冲写入特性，使用 XFS 有 3 倍性能提升。

h) PSI 优化：压力停止信息（PSI）提供了系统中当前资源使用情况的详细视图。新增 per-cgroup PSI、IRQ/SoftIRQ PSI 及一些优化。

内核配置项错误值检查工具 KconfigDetector

内核源码中 Kconfig 建模语言描述配置选项的定义以及模块间的相互作用关系，约束了内核配置过程中大量配置选项之间极其严格的上下游依赖关系以及相互之间的值影响限制，这极大增加了正确配置 Linux 内核的难度。若出现内核配置文件不满足 Kconfig 定义配置项取值约束的问题，可能导致无法正确编译内核，或在系统运行时发生错误，给应用生产安全带来威胁。

KconfigDetector 提供了一种可靠、快速的内核配置项错误值自动检测工具，通过一种新的形式化语义和针对 Kconfig 文件的解析框架，检测出内核配置文件中不满足依赖关系和取值限制的错误值，并提供父类和子类配置项查询，辅助用户正确配置内核。对于常见的内核配置项类型错误、取值错误、依赖错误和匹配错误等提供校验检查能力，提供辅助定位能力，进一步简化内核配置项的校验、问题定位能力。



功能描述

KconfigDetector 功能包含两部分：

1. 内核配置文件错误值检查

针对.config 文件能够发现以下错误类型:

- 类型错误: 配置项取值与类型不符
- 依赖不满足: 配置项未通过 select 关键字启动且依赖不满足
- 依赖风险: 配置项通过 select 关键字强制启动，但依赖未满足
- 未找到配置项: 未在内核 Kconfig 文件中找到指定配置项
- range 未满足: 不满足 range 关键字规定的取值范围
- 取值告警: 配置项取值未满足 default 或 imply 关键字要求

2. 内核配置项依赖关系查询

实现内核配置空间建模，将内核配置项的依赖关系以统一的 json 格式生成文件，能够快速查询所有直接和间接依赖的父类、子类配置项，可用于其它需求二次开发。

应用场景

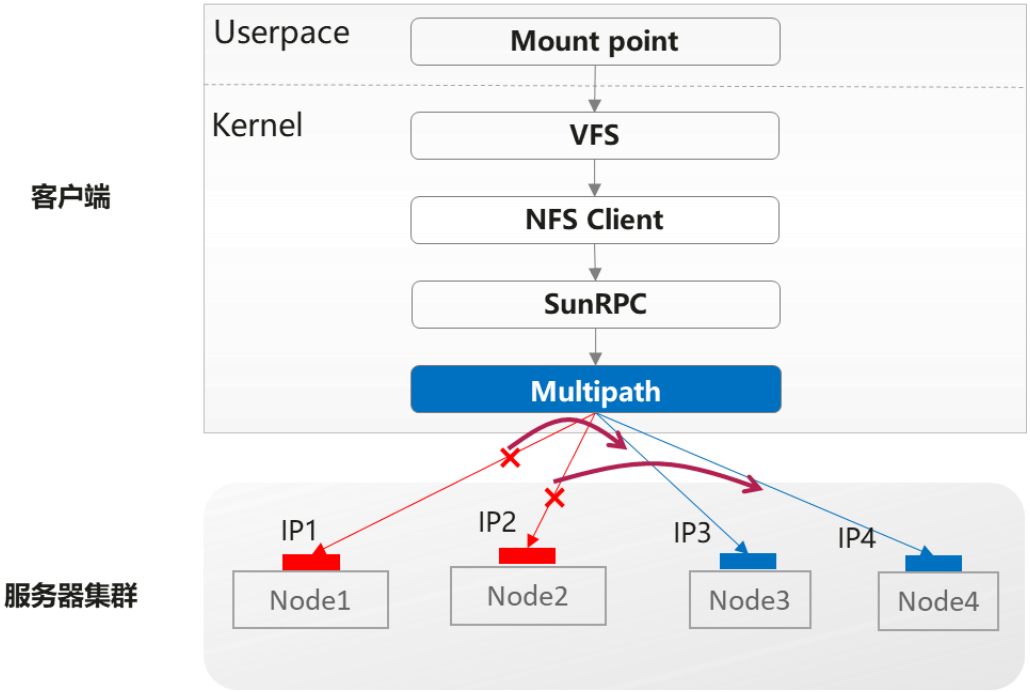
适用于内核测试、裁剪、移植，通过直接修改.config 文件而不是其它内核配置工具的方式来修改配置的情形，利用工具对修改后的.config 文件进行检查，使内核配置文件满足依赖关系。

NFS 多路径

网络文件系统 （NFS） 是一种分布式文件系统协议，最初由 Sun Microsystems （Sun） 于 1984 年开发，允许 NFS 客户端上的用户通过计算机网络访问 NFS 服务端上文件。随着 NFS 服务广泛应用于金融、EDA、AI、容器等行业应用，对 NFS 的性能和可靠性提出了更高的诉求。传统 NFS 存在以下缺点

- 单个主机上的单个挂载点只能通过一个客户端 IP 和一个服务端 IP 进行访问，在客户端和服务端之间存在多条物理链路时，无法发挥多条链路的性能。
- 由于单个挂载点单条链路的缺陷，单挂载点下的单条链路故障后，无法进行链路切换，导致主机业务中断。

功能描述



NFS 多路径的诞生主要解决上述传统 NFS 在使用过程中遇到的缺陷，提出单个挂载点下客户端和服务端之间建立多条链路，支持 IO 在多条链路中进行传输，进而提升单个挂载点性能，同时周期性检测链路状态信息，支持链路故障 IO 快速切换。

NFS 多路径具有以下功能：

- NFSv3 支持 Round Robin 链路选路算法，实现多条链路性能均衡。
- NFSv3 和 NFSv4 支持链路故障快速切换，提升 NFS 可靠性。
- 提供链路选路算法注册接口，支持 NFS 服务端开发者自定义选路算法。
- 支持周期性链路可用性检测。
- 支持查看链路状态。

应用场景

NFS 在 UNIX/Linux 环境里面应用比较广泛，应用场景涵盖企业办公场景、互联网应用场景、虚拟机应用场景、高性能计算场景等。

6. 特性增强

分布式软总线增强

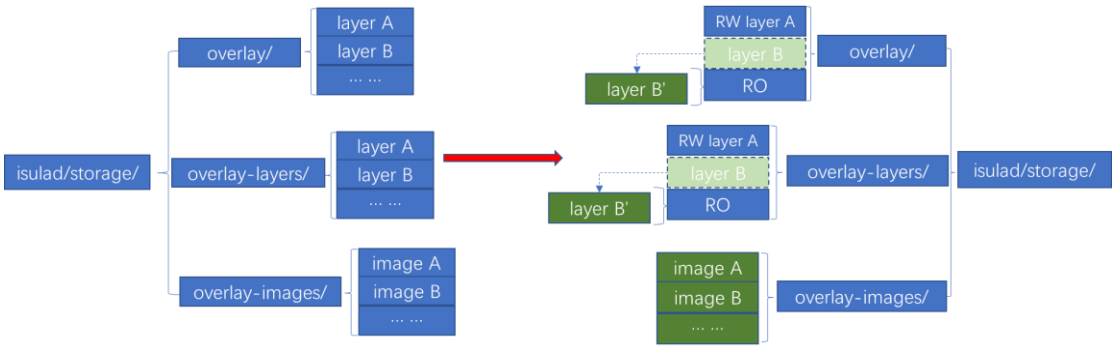
分布式软总线已集成 OpenHarmony 的分布式软总线和 hichain 点对点认证模块，实现欧拉嵌入式设备之间互联互通、欧拉嵌入式设备和 OpenHarmony 设备之间互联互通。本次版本新增 BLE 蓝牙发现能力，新增支持文件和流传输接口，引入 nstack 和 fillp 协议，相对历史版本在不稳定网络环境下传输效率明显提升。

iSulad 支持镜像 RO 数据管理目录拆分

iSulad支持镜像RO数据管理目录拆分：基于镜像的数据只读的特性，对镜像数据进行分类管理，实现镜像数据和容器数据的拆分；解耦了容器数据和镜像数据的存储，有利于用户对容器和镜像数据的灵活管理，为镜像数据分部署共享提供技术可能性。

功能描述

拆分目录结构效果如下：

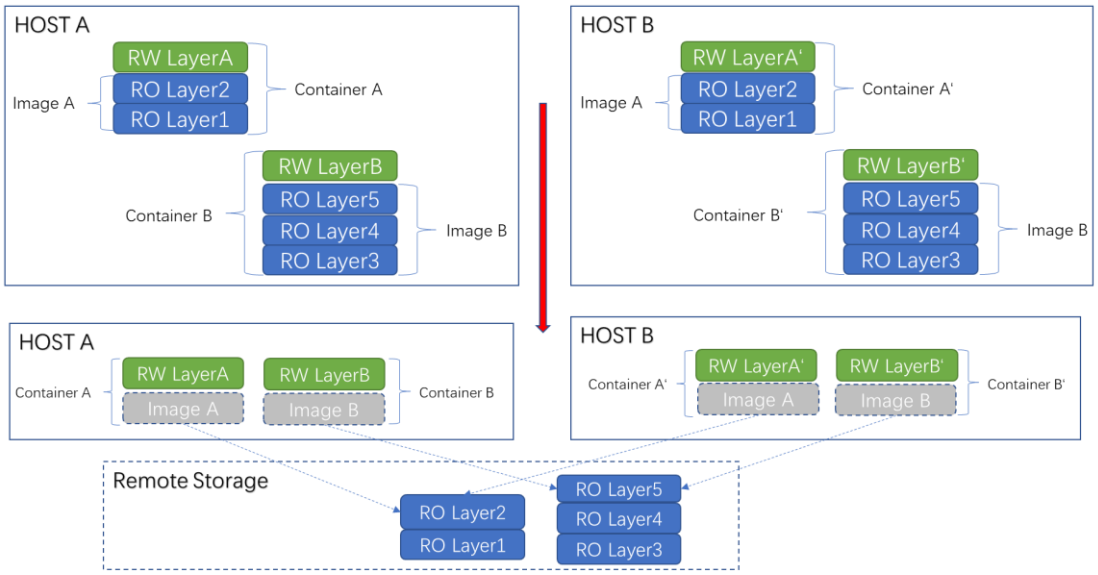


主要功能点：

- 支持用户配置，是否开启镜像 RO 数据拆分的能力；
- overlay-images 目录：支持镜像元数据的拆分能力；
- overlay-layers 目录：支持镜像层元数据的拆分能力；
- overlay 目录：支持镜像解压后数据的拆分能力；

应用场景

场景 1：基于分布式存储的镜像共享，大幅降低集群镜像的存储空间，提高容器的首次启动速度，以及减少容器镜像仓库的网络负载。



场景 2：本地独立分区或者磁盘作为镜像的存储空间，可以提高镜像数据的管理和转移便捷性。

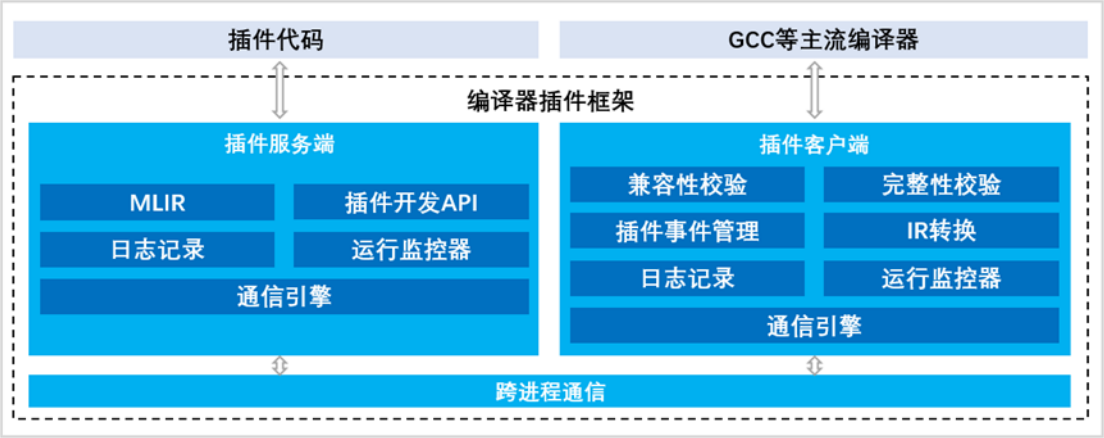
约束限制

1. 仅支持 overlayfs 镜像存储驱动。
2. 分布式存储场景，多节点删除、下载镜像操作同步由 K8S 层负责保障。

编译器插件框架特性增强

编译器插件框架提供面向 MLIR 的插件开发接口，以实现一次开发和多编译器落地为目标，避免编译工具的重复开发。编译器插件框架作为插件工具开发平台，提供对工具兼容性、完整性校验等公共能力的支持与维护，帮助用户以插件的形式提高优化特性的开发效率。

功能描述



- 提供基于 MLIR 的插件开发能力，支持与 GIMPLE 等编译器中间表示的转换。
- 插件框架提供对 19 类 GIMPLE 的支持。
- 支持兼容性检测、二进制完整性校验等公共能力。
- 支持安全编译选项校验、操作合法性校验等插件运行监控校验功能。
- 支持插件客户端作为 GCC 插件加载，可以在完全不需要修改 GCC 编译器代码的情况下实现插件功能。
- 支持为链接时优化（LTO，Link Time Optimization）使能插件框架。·

应用场景

应用场景 1：编译工具开发者构建工具，并需要完整性校验等公共能力

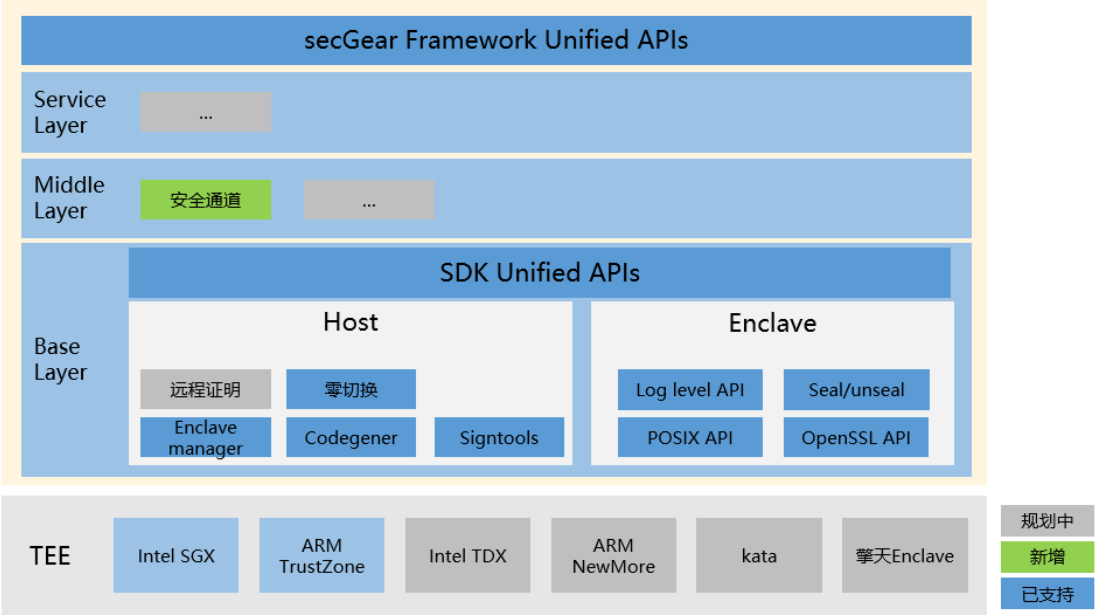
有多编译器落地需求的工具开发者，可以使用编译器插件框架作为开发平台，基于 MLIR 进行一次工具开发，即可在 GCC 等主流编译器上使能工具。编译器插件框架统一提供对兼容性检测、二进制完整性校验等公共能力的支持和维护。

应用场景 2：开发者以插件形式快速使能与验证编译相关工具

编译器插件框架提供以插件的形式运行在 GCC 等主流编译器上。无需对编译器进行源码改动，提高开发效率。

secGear 机密计算统一开发框架

secGear 是 openEuler 机密计算统一开发框架，致力于兼容业界主流 TEE，屏蔽 TEE 及 SDK 差异，对开发者提供统一、简易的开发接口，实现不同架构共源码，使开发者聚焦业务，降低机密计算应用开发维护成本，打通各 TEE 应用生态，助力机密计算生态建设。



secGear 从逻辑上分为三层，共同组成 openEuler 机密计算软件生态底座。

- Base Layer：机密计算 SDK 统一层，屏蔽 TEE 及 SDK 差异，实现不同架构共源码。
- Middleware Layer：通用组件层，机密计算软件货架，无需从头造轮子，帮助用户快速构建机密计算解决方案。
- Server Layer：机密计算服务层，提供典型场景机密计算解决方案。

本次版本新增支持安全通道通用特性。

客户痛点

数据拥有者在请求云上机密计算服务时，需要把待处理数据上传到云上 TEE 环境中处理，由于 TEE 没有网络，用户数据需要经过网络先传输到 REE，REE 接收到数据的明文后，再传入 TEE 中。用户数据的明文暴露在 REE 内存中，存在安全风险。

解决方案

安全通道是一种结合机密计算远程证明，实现数据拥有者与云上 TEE 之间安全的密钥协商技术，协商出仅数据拥有者与云上 TEE 拥有的 sessionkey，再通过 sessionkey 加密用户

数据，网络传输的是 sessionkey 加密后的数据，REE 接收到密文数据，再传入 TEE 中解密，处理。

功能描述

安全通道 SDK 分为客户端、服务端 host、服务端 enclave 三部分，分别由业务的客户端、服务端 CA、服务端 TA 集成。

- 支持客户端（REE）-服务端（TEE）之间安全通道协商：协商完成后，两端拥有相同的 key 加解密。
- 客户端支持加解密。
- 服务端 TA 支持加解密。

应用场景

应用场景 1：密态数据库

密态数据库在 TEE 中提供 SQL 查询和计算能力，当数据库客户端请求查询时会将密文传入 TEE 中执行查询，数据库密文记录的密钥在客户端，客户端通过安全通道将密钥送到 TEE 中，在 TEE 中解密密文，完成查询。

应用场景 2：Mindspore 基于可信执行环境的特征保护

纵向联邦学习中 Follower 方从原始数据得到中间结果发送给 Leader 方，Leader 方使用中间结果和标签训练得到梯度信息，回传给各 Follower 方。然而攻击者可能从内存中获取到 Follower 方上传的中间结果并反推出用户信息，存在安全隐患。将 Leader 方中间结果处理逻辑部署在 TEE 中，Follower 方通过安全通道将中间结果传入 Leader 方的 TEE 中处理，从而保护 Follower 方的中间结果仅在 Leader 方 TEE 中明文存在。

radiaTest 社区测试平台

radiaTest 是一个 openEuler 社区孵化的用以承载社区全流程测试活动的管理平台。该平台核心为 web 端数据中台，帮助社区版本测试高效运作，使能社区版本测试可跟踪可追

溯。具备支撑资源管理以及自动化测试功能的插件化服务，支持对接多元测试引擎。

本次版本更新主要是为了解决平台的高鉴权门槛问题，并对能力进行补全。

功能描述

- 重构登录鉴权逻辑，使能各组织/社区采用多元化鉴权方式。如登录 openEuler 组织基于 openEuler 社区统一鉴权，而非码云企业仓应用鉴权
- 质量看板新增版本基线用例测试进展展示，支撑以版本视角看护用例执行状况
- 测试设计 IT 化，提供组织/社区测试策略管理易用入口
- 优化问题单提交功能，支持对测试任务关联问题单数据进行统计

应用场景

应用场景 1：社区开发者于版本测试前期贡献测试策略

社区开发者可以在工作台的测试设计页面找到当前处于版本测试阶段的产品版本后，选择负责编写的特性节点。平台可提供测试策略脑图的新增、编辑、删除功能，以及支持对接码云 openEuler QA 仓库，使能归档一致性和测试策略格式互转与导入导出功能。

应用场景 2：版本质量看护，实时跟踪版本基线用例执行总体进展

社区开发者、测试经理或版本 QA 可以在版本质量看板上直观看到当前版本用例基线的测试执行情况，对版本测试的实际执行进展能够感知。

应用场景 3：测试者基于确认的问题，向社区提交问题单，问题单和测试任务存在关联

测试任务的关联用例如果出现了确认的问题，测试者可以在测试任务页面提交问题单同步至码云仓库。通过这种形式提单后，测试任务即可完成与问题单的关联，实现任务与问题的 IT 化承载。

7. 著作权说明

何人不得将 openEuler 白皮书上的任何内容以任何方式进行复制、经销、翻印、传播、以超级链路连接或传送、以镜像法载入其他服务器上、存储于信息检索系统或者其他任何商业目的的使用，但对于非商业目的的、用户使用的下载或打印（条件是不得修改，且须保留该材料中的版权说明或其他所有权的说明）除外。

8. 商标

openEuler 白皮书上使用和显示的所有商标、标志皆属 openEuler 社区所有，但注明属于其他方拥有的商标、标志、商号除外。未经 openEuler 社区或其他方书面许可，openEuler 白皮书所载的任何内容不应被视作以暗示、不反对或其他形式授予使用前述任何商标、标志的许可或权利。未经事先书面许可，任何人不得以任何方式使用 openEuler 社区的名称及 openEuler 社区的商标、标记。

9. 附录

附录 1：搭建开发环境

环境准备	地址
下载安装 openEuler	https://openeuler.org/zh/download/
开发环境准备	https://gitee.com/openeuler/community/blob/master/zh/contributors/prepare-environment.md
构建软件包	https://gitee.com/openeuler/community/blob/master/zh/contributors/package-install.md

附录 2：安全处理流程和安全批露信息

社区安全问题披露	地址
安全处理流程	https://gitee.com/openeuler/security-committee/blob/master/security-process.md
安全披露信息	https://gitee.com/openeuler/security-committee/blob/master/security-disclosure.md