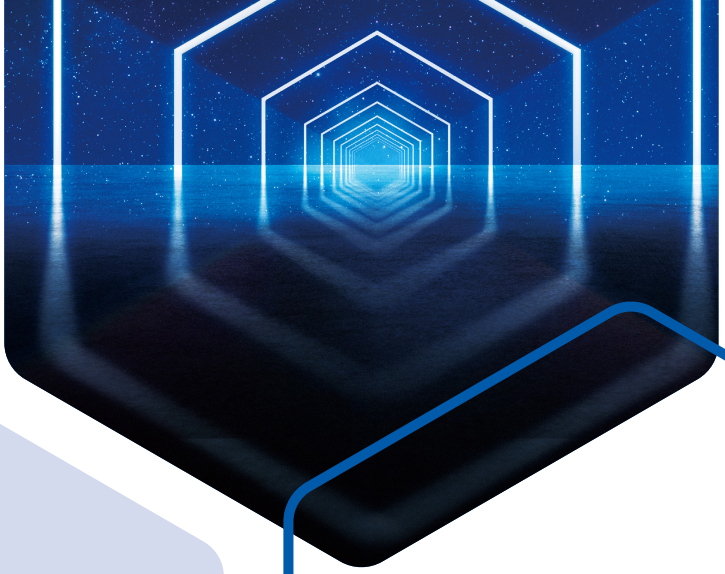
openEuler 25.03

技术白皮书



CONTENTS

01 概述 01

02 平台架构 04

03 运行环境 07

04 场景创新 10

05 内核创新 25

06 云化基座 27

07 特性增强 29

08 著作权说明 47

09 商标 48

10 附录 49

01 概述

OpenAtom openEuler（简称“openEuler”）社区是一个面向数字基础设施操作系统的开源社区，由开放原子开源基金会（以下简称“基金会”）孵化及运营。

openEuler 是一个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。

openEuler 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，面向多样性计算的操作系统开源社区 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, 并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 openEuler 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。

2022 年 9 月 30 日，发布 openEuler 22.09 创新版本，持续补齐全场景的支持。

2022 年 12 月 30 日，发布 openEuler 22.03 LTS SP1 版本，打造最佳迁移工具实现业务无感迁移，性能持续领先。

2023 年 3 月 30 日，发布 openEuler 23.03 内核创新版本，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、基础技术创新及上层应用创新。

2023 年 6 月 30 日，发布 openEuler 22.03 LTS SP2 版本，场景化竞争力特性增强，性能持续提升。

2023 年 9 月 30 日，发布 openEuler 23.09 创新版本，是基于 6.4 内核的创新版本（参见版本生命周期），提供更多新特性和功能，给开发者和用户带来全新的体验，服务更多的领域和更多的用户。

2023 年 11 月 30 日，发布 openEuler 20.03 LTS SP4 版本，其作为 20.03 LTS 版本的增强扩展版本，面向服务器、云原生、边缘计算场景，提供更多新特性和功能增强。

2023 年 12 月 30 日，发布 openEuler 22.03 LTS SP3 版本，是 22.03 LTS 版本增强扩展版本，面向服务器、云原生、边缘计算和嵌入式场景，持续提供更多新特性和功能扩展，给开发者和用户带来全新的体验，服务更多的领域和更多的用户。

2024 年 5 月 30 日，发布 openEuler 24.03 LTS，基于 6.6 内核的长周期 LTS 版本（参见版本生命周期），面向服务器、云、边缘计算、AI 和嵌入式场景，提供更多新特性和功能，给开发者和用户带来全新的体验，服务更多的领域和更多的用户。

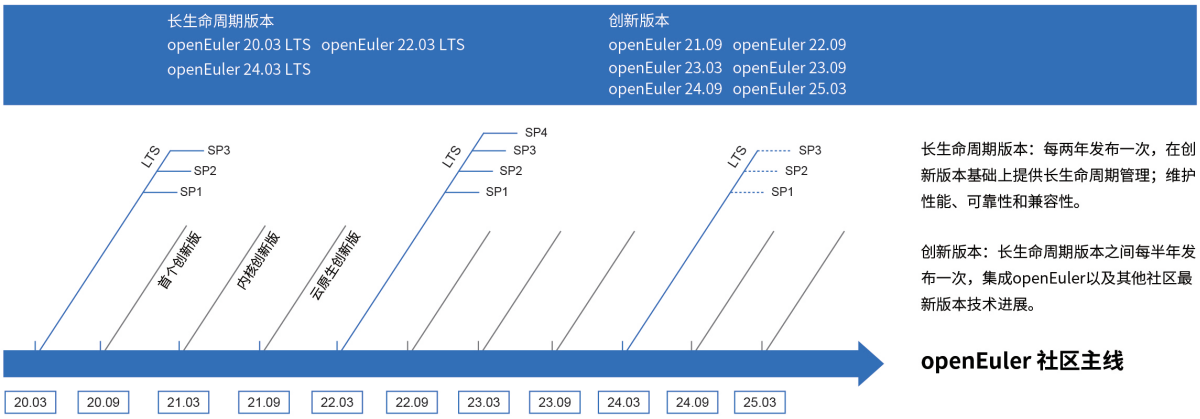
2024 年 6 月 30 日，发布 openEuler 22.03 LTS SP4，是 22.03 LTS 版本增强扩展版本，面向服务器、云原生、边缘计算和嵌入式场景，持续提供更多新特性和功能扩展，给开发者和用户带来全新的体验，服务更多的领域和更多的用户。

2024 年 9 月 30 日，发布 openEuler 24.09，基于 6.6 内核的创新版本，提供更多新特性和功能。

2024 年 12 月 30 日，发布 openEuler 24.03 LTS SP1，基于 6.6 内核的 24.03-LTS 版本增强扩展版本（参见版本生命周期），面向服务器、云、边缘计算和嵌入式场景，持续提供更多新特性和功能扩展，给开发者和用户带来全新的体验，服务更多的领域和更多的用户。

2025 年 3 月 30 日，发布 openEuler 25.03 是基于 6.6 内核的创新版本，面向服务器、云、边缘计算和嵌入式场景，提供更多新特性和功能，给开发者和用户带来全新的体验，服务更多的领域和更多的用户。

openEuler 版本管理

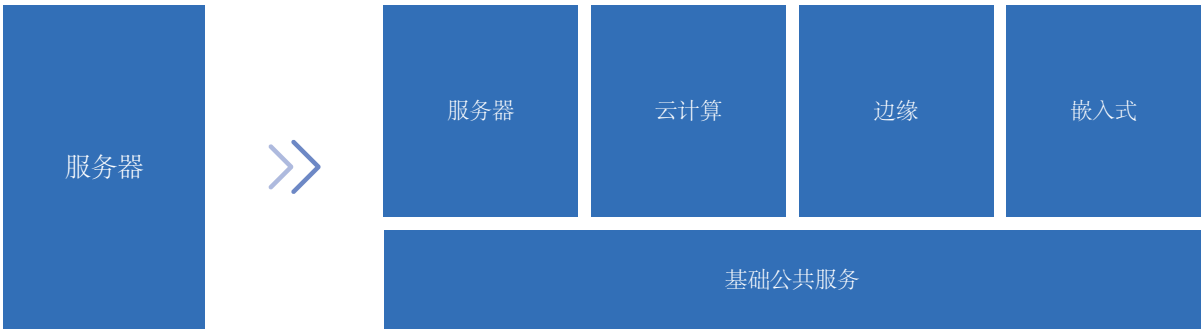


openEuler 作为一个操作系统发行版平台，每两年推出一个 LTS 版本。该版本为企业级用户提供一个安全稳定可靠的操作系统。

openEuler 也是一个技术孵化器。通过每半年发布一个创新版，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社区发行版，社区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 已支持 X86、ARM、SW64、RISC-V、LoongArch 多处理器架构及 PowerPC 芯片架构，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 Edge、面向嵌入式的版本 openEuler 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 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、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。

02 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。

openEuler 25.03 发布面向服务器、云原生、边缘和嵌入式场景的全场景操作系统版本，统一基于 Linux Kernel 6.6 构建，对外接口遵循 POSIX 标准，具备天然协同基础。同时 openEuler 25.03 版本集成分布式软总线、KubeEdge+ 边云协同框架等能力，进一步提升数字基础设施协同能力，构建万物互联的基础。

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。

夯实云化基座

- 容器操作系统 KubeOS：云原生场景，实现 OS 容器化部署、运维，提供与业务容器一致的基于 K8S 的管理体验。
- 安全容器方案：iSulad+shimv2+StratoVirt 安全容器方案，相比传统 Docker+QEMU 方案，底噪和启动时间优化 40%。
- 双平面部署工具 eggo：ARM/X86 双平面混合集群 OS 高效一键式安装，百节点部署时间 <15min。

新场景

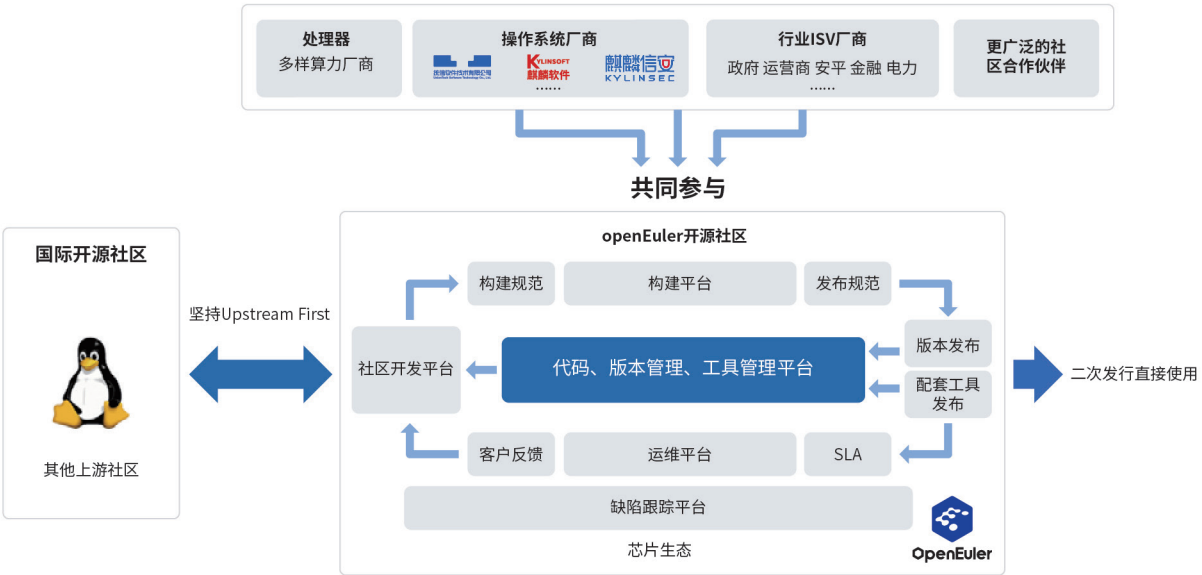
- 边缘计算：发布面向边缘计算场景的版本，支持 KubeEdge+ 边云协同框架，具备边云应用统一管理和发放等基础能力。
- 嵌入式：发布面向嵌入式领域的版本，镜像大小 < 5M，启动时间 < 5s。
- AI OS：OS 使能 AI 软件栈，开箱即用；异构融合内存，调度，训推场景降本增效；智能化交互平台，赋能开发者及管理员。

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。
- openEuler 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

全版本支持的硬件型号可在兼容性网站查询：https://www.openeuler.org/zh/compatibility/。

03 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、RISC-V
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 20GB

虚拟机

openEuler 安装时，应注意虚拟机的兼容性问题，当前已测试可以兼容的虚拟机及组件如下所示。

1. 以 openEuler 25.03 为 HostOS，组件版本如下：

- libvirt-9.10.0-12.oe2409
- libvirt-client-9.10.0-12.oe2409
- libvirt-daemon-9.10.0-12.oe2409
- qemu-8.2.0-17.oe2409
- qemu-img-8.2.0-17.oe2409

2. 兼容的虚拟机列表如下：

Host OS	GuestOS(虚拟机)	架构
openEuler 25.03	CentOS 6	x86_64
openEuler 25.03	CentOS 7	aarch64
openEuler 25.03	CentOS 7	x86_64
openEuler 25.03	CentOS 8	aarch64
openEuler 25.03	CentOS 8	x86_64
openEuler 25.03	Windows Server 2016	x86_64
openEuler 25.03	Windows Server 2019	x86_64
openEuler RISC-V 25.03	Ubuntu 24.10	RISC-V
openEuler RISC-V 25.03	Fedora 41	RISC-V

部件名称	最小虚拟化空间要求
架构	ARM64、x86_64、RISC-V
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 20GB

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、RISC-V
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 20GB

嵌入式

若需要在嵌入式环境上安装 openEuler Embedded 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、x86_64
内存	为了获得更好的体验，建议不小于 512MB
硬盘	为了获得更好的体验，建议不小于 256MB

04 场景创新

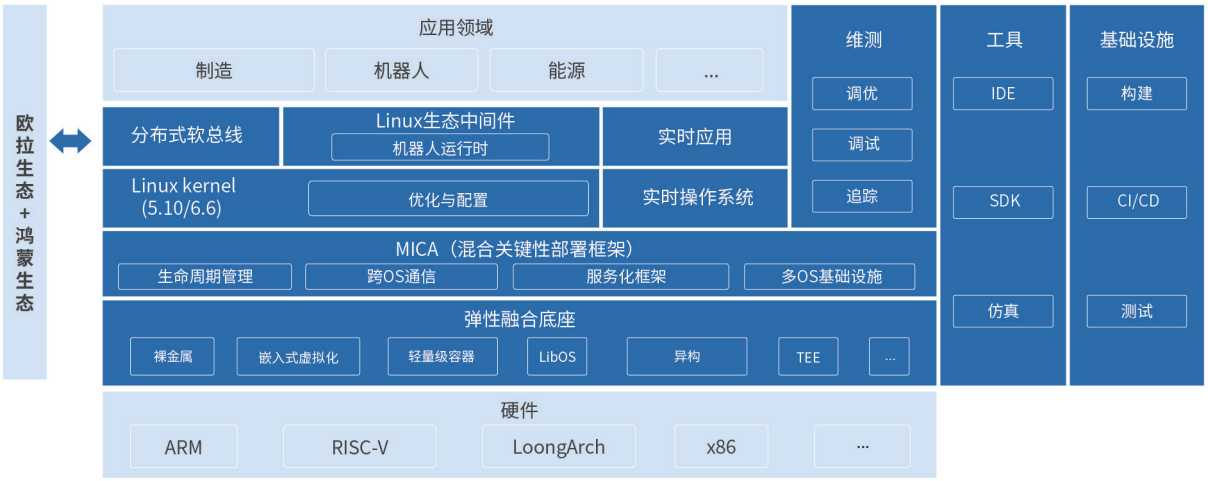
openEuler Embedded

openEuler 发布面向嵌入式领域的版本 openEuler 25.03，提供更丰富的南北向生态以及更完善的基础设施。

openEuler Embedded 围绕以制造、机器人为代表的 OT 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系统软件栈和生态。在南向生态方面，开展了“openEuler Embedded 的生态扩展计划”，持续完善了鲲鹏 920、泰山派等硬件的支持，并携手意法半导体和米尔科技，成功完成了面向工业应用的高性能微处理器 STM32MP257 的适配，推动嵌入式生态扩展。在北向，持续丰富了工业中间件、图形中间件等能力，在工业制造、机器人等领域落地应用。

未来 openEuler Embedded 将协同 openEuler 社区生态伙伴、用户、开发者，逐步扩展支持龙芯等新的芯片架构和更多的南向硬件，完善嵌入式 AI、嵌入式边缘、仿真系统等能力，打造综合嵌入式系统软件平台解决方案。

系统架构图



南向生态

openEuler Embedded Linux 当前主要支持 ARM64、x86-64、ARM32、RISC-V 等多种芯片架构，未来计划支持龙芯等架构，从 24.03 版本开始，南向支持大幅改善，已经支持树莓派、海思、鲲鹏、瑞芯微、瑞萨、德州仪器、飞腾、赛昉、全志、意法半导体等厂商的芯片。

嵌入式弹性虚拟化底座

openEuler Embedded 的弹性虚拟化底座是为了在多核片上系统（SoC, System On Chip）上实现多个操作系统共同运行的一系列技术的集合，包含了裸金属、嵌入式虚拟化、轻量级容器、LibOS、可信执行环境（TEE）、异构部署等多种实现形态。不同的形态有各自的特点：

1. 裸金属：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 Embedded Linux 混合部署。
2. 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 UniProton/Zephyr/FreeRTOS 和 openEuler Embedded Linux 混合部署，也支持 openHarmony 和 openEuler Embedded Linux 的混合部署。
3. 实时虚拟化：openEuler 社区孵化了嵌入实时虚拟机监控器 ZVM 和基于 rust 语言的 Type-I 型嵌入式虚拟机监控器 Rust-Shyper，可以满足不同场景的需求。

混合关键性部署框架

openEuler Embedded 打造了构建在融合弹性底座之上混合关键性部署框架，并命名为 MICA（MIxed CriticAlity），旨在通过一套统一的框架屏蔽下层弹性底座形态的不同，从而实现 Linux 和其他 OS 运行时便捷地混合部署。依托硬件上的多核能力使得通用的 Linux 和专用的实时操作系统有效互补，从而达到全系统兼具两者的特点，并能够灵活开发、灵活部署。

MICA 的组成主要有四大部分：生命周期管理、跨 OS 通信、服务化框架和多 OS 基础设施。生命周期管理主要负责从 OS（Client OS）的加载、启动、暂停、结束等工作；跨 OS 通信为不同 OS 之间提供一套基于共享内存的高效通信机制；服务化框架是在跨 OS 通信基础之上便于不同 OS 提供各自擅长服务的框架，例如 Linux 提供通用的文件系统、网络服务，实时操作系统提供实时控制、实时计算等服务；多 OS 基础设施是从工程角度为把不同 OS 从工程上有机融合在一起的一系列机制，包括资源表达与分配，统一构建等功能。

混合关键性部署框架当前能力：

1. 支持裸金属模式下 openEuler Embedded Linux 和 RTOS（Zephyr/UniProton）的生命周期管理、跨 OS 通信。
2. 支持分区虚拟化模式下 openEuler Embedded Linux 和 RTOS（FreeRTOS/Zephyr）的生命周期管理、跨 OS 通信。

北向生态

1. 北向软件包支持：600+ 嵌入式领域常用软件包的构建。
2. 软实时内核：提供软实时能力，软实时中断响应时延微秒级。
3. 分布式软总线基础能力：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。
4. 嵌入式容器与边缘：支持 iSula 容器，可以实现在嵌入式上部署 openEuler 或其他操作系统容器，简化应用移植和部署。支持生成嵌入式容器镜像，最小大小可到 5MB，可以部署在其他支持容器的操作系统之上。

UniProton 硬实时系统

UniProton 是一款实时操作系统，具备极致的低时延和灵活的混合关键性部署特性，可以适用于工业控制场景，既支持微控制器 MCU，也支持算力强的多核 CPU。目前关键能力如下：

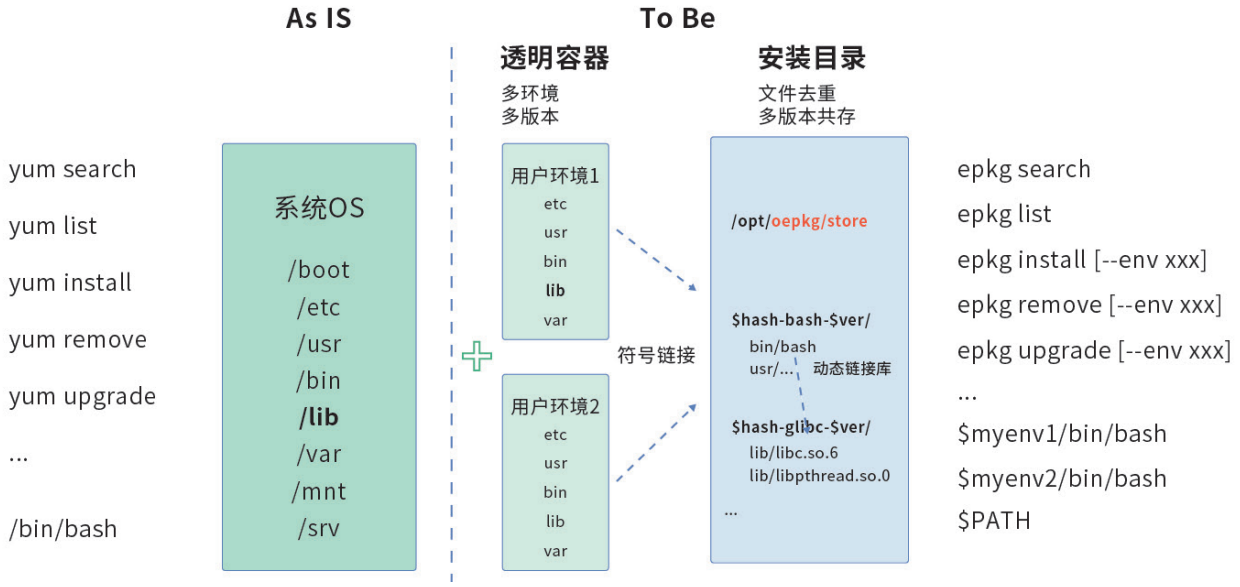
- 支持 Cortex-M、ARM64、X86_64、riscv64 架构，支持 M4、RK3568、RK3588、X86_64、Hi3093、树莓派 4B、鲲鹏 920、昇腾 310、全志 D1s。
- 支持树莓派 4B、Hi3093、RK3588、X86_64 设备上通过裸金属模式和 openEuler Embedded Linux 混合部署。
- 支持通过 gdb 在 openEuler Embedded Linux 侧远程调试

应用场景

openEuler Embedded 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。

epkg 新型软件包

epkg 是一款新型软件包，支持普通用户在操作系统中安装及使用。新的软件包格式相比现有软件包，主要解决多版本兼容性问题，用户可以在一个操作系统上通过简单地命令行安装不同版本的软件包。同时支持环境管理实现环境的创建 / 切换 / 使能等操作，来使用不同版本的软件包。目前 epkg 主要支持非服务类的软件包的安装和使用。



功能描述

多版本兼容：支持安装不同版本的软件包，实现多版本软件包的共存。

多安装模式：支持特权用户安装及普通用户安装，实现全局可用、用户级可用，最小化安装程序，绿色安装。

环境管理：支持环境创建 / 删除 / 激活 / 注册 / 查看等操作，支持多环境使用不同软件仓，实现多环境多版本能力。支持运行态注册多个环境，开发态激活单环境调试。

环境回退：支持查看环境历史操作，环境回退操作，实现用户在误操作或安装软件后出现问题时，恢复环境。

包管理：支持包安装 / 卸载 / 查询能力，对标 rpm/dnf 基本功能，满足典型用户与场景日常使用。

应用场景

适用于用户希望安装软件包的多个版本的场景，用户能够通过切换环境使用不同的版本的软件包，解决兼容性难题。

使用文档参考 : epkg 使用文档。

GCC 编译链接加速

为了提升 openEuler 软件包的编译效率，从而进一步提升门禁和开发者开发效率，通过编译器、链接器优化技术缩短应用中 C/C++ 组件总体编译时间。

功能描述

使用打开 PGO（Profile Guided Optimization）、LTO（Link Time Optimization）编译的 GCC 结合 mold（A Modern Linker）链接器缩短软件包中 C/C++ 库的编译时间，最终使得编译时间 TOP90+ 软件包的编译总时间缩短 9.5% 左右。具体支持以下两点功能：

1. 支持能够编译出带 PGO 和 LTO 优化的 gcc12.3 版本来提升编译过程效率；
2. 支持能够根据应用白名单切换到 mold 链接器来提升链接过程效率。

应用场景

适用于存在大量 C/C++ 组件的应用编译过程，可通过编译和链接优化提升 C/C++ 编译效率，从而提升应用开发效率。

AI

智能时代，操作系统需要面向 AI 不断演进。一方面，在操作系统开发、部署、运维全流程以 AI 加持，让操作系统更智能；另一方面，openEuler 已支持 ARM，x86，RISC-V 等全部主流通用计算架构，在智能时代，openEuler 也率先支持 NVIDIA、昇腾等主流 AI 处理器，成为使能多样性算力的首选。

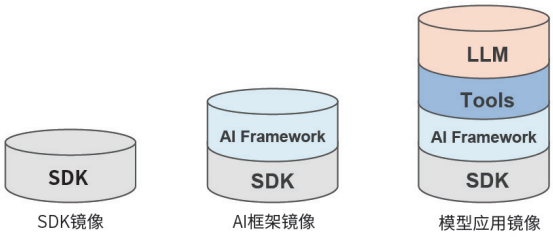
OS for AI

openEuler 兼容 NVIDIA、Ascend 等主流算力平台的软件栈，为用户提供高效的开发运行环境。通过将不同 AI 算力平台的软件栈进行容器化封装，即可简化用户部署过程，提供开箱即用的体验。同时，openEuler 也提供丰富的 AI 框架，方便大家快速在 openEuler 上使用 AI 能力。

开箱易用

功能描述

1. openEuler 已兼容 CANN、CUDA 等硬件 SDK，以及 TensorFlow、PyTorch、 MindSpore 等相应的 AI 框架软件，支持 AI 应用在 openEuler 上高效开发与运行。
2. openEuler AI 软件栈容器化封装优化环境部署过程，并面向不同场景提供以下三类容器镜像。



- SDK 镜像：以 openEuler 为基础镜像，安装相应硬件平台的 SDK，如 Ascend 平台的 CANN 或 NVIDIA 的 CUDA 软件。
- AI 框架镜像：以 SDK 镜像为基础，安装 AI 框架软件，如 PyTorch 或 TensorFlow。此外，通过此部分镜像也可快速搭建 AI 分布式场景，如 Ray 等 AI 分布式框架。
- 模型应用镜像：在 AI 框架镜像的基础上，包含完整的工具链和模型应用。

相关使用方式请参考 openEuler AI 容器镜像用户指南。

应用场景

openEuler 使能 AI，向用户提供更多 OS 选择。基于 openEuler 的 AI 容器镜像可以解决开发运行环境部署门槛高的问题，用户根据自身需求选择对应的容器镜像即可一键部署，三类容器镜像的应用场景如下：

- SDK 镜像: 提供对应硬件的计算加速工具包和开发环境，用户可进行 Ascend CANN 或 NVIDIA CUDA 等应用的开发和调试。同时，可在该类容器中运行高性能计算任务，例如大规模数据处理、并行计算等。
- AI 框架镜像：用户可直接在该类容器中进行 AI 模型开发、训练及推理等任务。
- 模型应用镜像：已预置完整的 AI 软件栈和特定的模型，用户可根据自身需求选择相应的模型应用镜像来开展模型推理或微调任务。

sysHAX 大语言模型异构协同加速运行时

功能描述

sysHAX 大语言模型异构协同加速运行时专注于单机多卡环境下大模型推理任务的性能提升，针对鲲鹏 +xPU（GPU、NPU 等）的异构算力协同，显著提升大模型的吞吐量和并发量：

- 算子下发加速：适配 vllm v0.6.6 版本，进行调度引擎优化，降低 CPU 侧时延。
- CPU 推理加速：通过 NUMA 亲和调度、矩阵运算并行加速、SVE 指令集推理算子适配等方式，提升 CPU 的吞吐量。

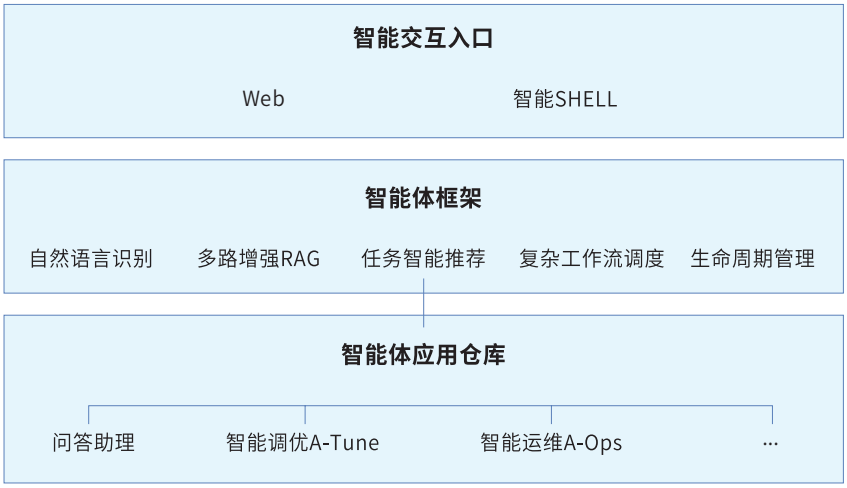
应用场景

sysHAX 大语言模型推理优化方案当前支持 DeepSeek、Qwen、baichuan、Llama 等 transformer 架构的模型。其中，CPU 推理加速能力已完成了对 DeepSeek 7B、14B、32B 以及 Qwen 系列模型的适配。主要适用于以下典型场景：

- 数据中心场景：sysHAX 通过上述技术，利用 CPU 填充推理任务，充分利用 CPU 资源，增加大模型并发量与吞吐量。

AI for OS

当前，openEuler 和 AI 深度结合，一方面使用基础大模型，基于大量 openEuler 操作系统的代码和数据，训练出 openEuler 大模型智能交互平台，初步实现代码辅助生成、智能问题智能分析、系统辅助运维等功能，让 openEuler 更智能。



AI 应用开发框架

大模型应用是实现企业大模型落地的重要途径。RAG（检索增强生成技术）+ 大模型可以很好的弥补基础模型对行业数据缺失的不足。因此，结合 RAG 能力的 AI 应用开发平台是当前企业和开发者的一大助力。基于此，openEuler 社区孵化出了基于 openEuler 操作系统的 AI 应用开发框架。

功能描述

该框架面向个人及企业开发者，提供智能辅助工具，帮助开发者通过简单高效的交互方式快速完成 AI 应用开发任务。可有效降低AI应用开发的技术门槛，提升开发效率和数据处理质量，并能够满足多种复杂应用场景下的数据处理、模型训练及内容管理需求。框架主要包含以下核心功能：

- 多路增强 RAG 提升智能问答准确率：为解决传统 Native RAG 精确性低、指导性不强等问题，多路增强 RAG 通过语料治理、prompt 重写、多路召回比对等技术，提升 RAG 检索准确率；
- 文档治理与优化：支持语料增量更新，支持文本去重、敏感信息脱敏、文档标准化与内容格式化（如段落总结、代码注释、案例整理），提高语料质量。
- Embedding 模型微调：支持多种 Embedding 模型（如 bge 系列模型）的快速微调与评估，提升模型在特定领域的表现。

应用场景

AI 应用开发框架适用于广泛的智能开发与数据处理场景。对企业及个人开发者，可显著提升 AI 开发效率与应用质量。具体包括但不限于以下应用场景：

- 数据预处理与标注：为 AI 模型训练快速生成高质量的问答数据，节省人工标注时间。
- 文档质量提升与管理：高效完成文档的脱敏、去重与标准化处理，提升文档信息安全性与一致性。
- 行业专属模型构建：通过微调 Embedding 模型，为特定行业或业务场景打造专属的高性能模型，提升应用场景的模型匹配度与使用效果。

智能问答

功能描述

智能问答平台目前支持 Web 和智能 Shell 两个入口。

- Web 入口：简单易用。用户可以以问答方式咨询有关 openEuler 操作系统的各类基础知识、获取 openEuler 社区的最新动态、寻找运维问题的解决方案，以及使用 openEuler 大模型智能交互平台提供的各类智能体等。
- 智能 Shell 入口：允许用户以自然语言与 openEuler 进行交互，提供启发式的系统调优、故障诊断运维体验，使管理和维护系统变得更加直观高效。

工作流调度

原子化智能体操作流程：通过采用“流”的组织形式，openEuler 大模型智能交互平台允许用户将智能体的多个操作过程组合成一个内部有序、相互关联的多步骤“工作流”。每个工作流作为一个不可分割的原子操作执行，并支持用户自定义错误恢复机制。这种设计确保了在执行复杂任务时，智能体的操作具备一致性、可追溯性和稳定性。

即时数据处理：智能体在工作流的每个步骤中生成的数据和结果能够立即得到处理，并无缝传递到下一个步骤。用户可以在各步骤间轻松传递信息和引用所需数据。最终步骤产生的结果可以通过多种方式在前端展示，例如报表、图片或代码块等。

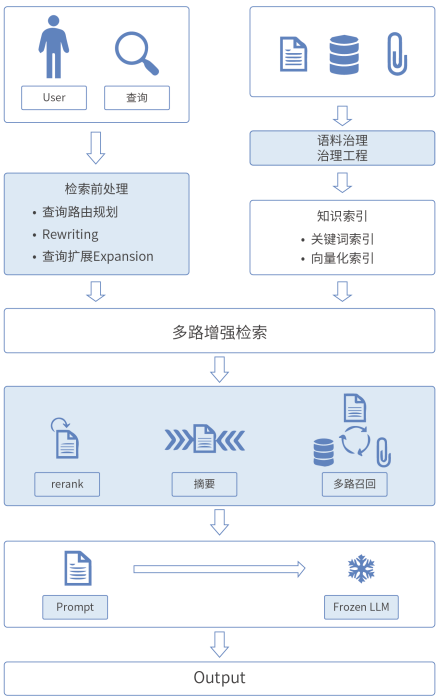
智能交互：在面对模糊或复杂的用户指令时，openEuler 大模型智能交互平台能主动询问用户，以澄清或获取更多信息。用户在补充信息后，openEuler 大模型智能交互平台能够继续执行工作流，并确保所采取的行动符合用户的期望。

任务推荐

智能响应：openEuler 大模型智能交互平台能够分析用户输入的语义信息。当用户以文字形式输入指令后，openEuler 大模型智能交互平台能够理解用户意图，并根据上下文环境选择最匹配的工作流进行执行。openEuler 大模型智能交互平台还能够学习和适应用户的习惯和偏好。

智能指引：openEuler 大模型智能交互平台综合分析当前工作流的执行状况、功能需求以及关联任务等多维度数据，为用户量身定制最适宜的下一步操作建议。这些建议不仅考虑了用户的个人偏好和历史行为模式，还通过直观的前端展示简化了决策过程，助力用户高效推进任务。此智能化推荐机制显著提升了工作效率，减少了用户在选择行动时的犹豫和不确定性。

RAG



RAG(检索增强技术) 是为了增强大模型长期记忆能力和降低大模型训练成本诞生的技术，相较传统 RAG，openEuler 大模型智能交互平台中的 RAG 技术在检索前处理、知识索引和检索增强方面做了改进：

- 检索前处理：对复杂的用户查询场景，提供用户查改写和路由能力，具有查询规划特性；
- 知识索引：对文档内容多样化场景，提供关键字提取和文本向量化能力，具有片段索引构建特性；
- 多路召回：对知识来源多样化，提供向量、关键字检索和 chat2DB 的能力，具有查询结果 rerank、过滤和融合特性。

通过以上能力，相较传统的 RAG 技术，openEuler 大模型智能交互平台中的 RAG 技术能更强的适应多种文档格式和内容场景，在不为系统增加较大负担的情况下，增强问答服务体验。

语料治理

语料治理是 openEuler 大模型智能交互平台中的 RAG 技术的基础能力之一，其通过片段相对关系提取、片段衍生物构建和 OCR 等方式将语料以合适形态入库，以增强用户查询命中期望文档的概率：

- 片段相对关系提取：对保持文档内容连续性场景，提供保留片段相对关系能力，具有上下文关联的特性；
- 片段衍生物构建：对复杂片段，提供片段摘要能力，具有通过摘要间接命中片段的特性；
- OCR：对图文混合场景，提供 OCR+ 上下文内容联合摘要的能力，具有图片文本提取和总结的特性。

通过以上语料治理手段，可以增强问答服务在多轮对话、内容完整性和图文展示上的体验。

应用场景

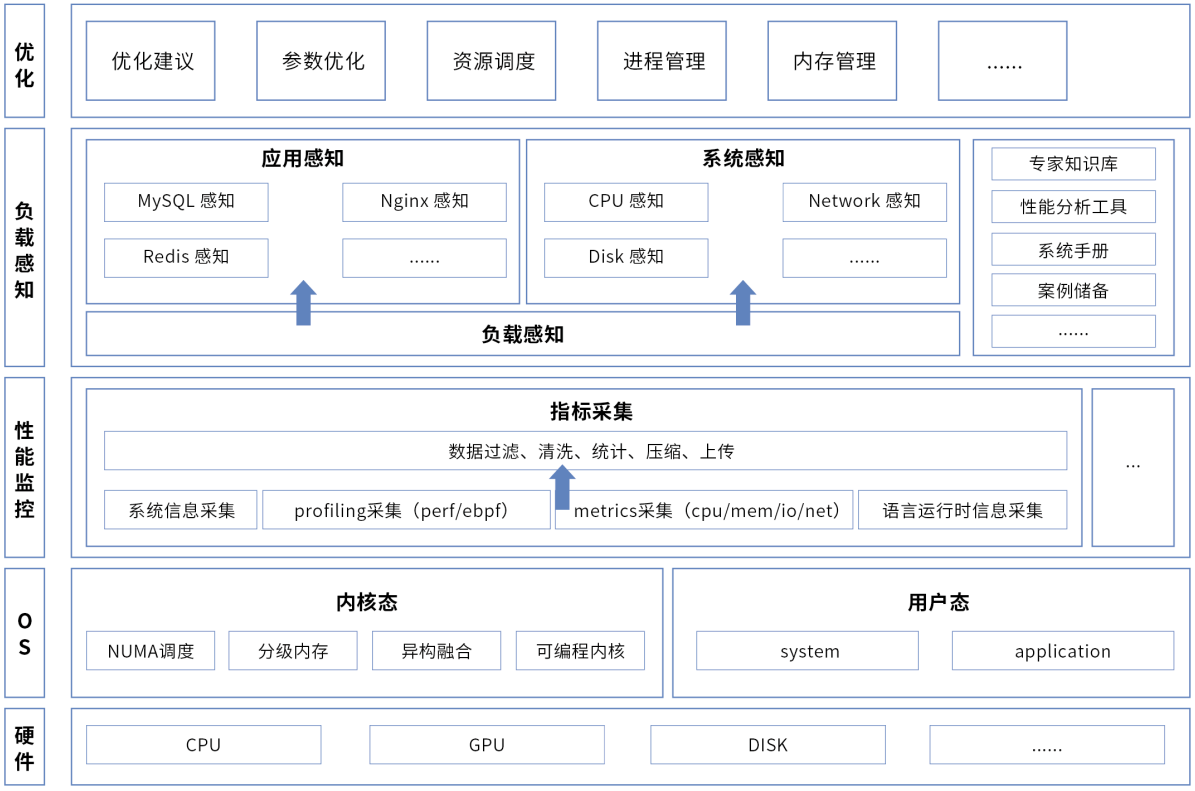
- 面向 openEuler 普通用户：深入了解 openEuler 相关知识和动态数据，比如咨询如何迁移到 openEuler。
- 面向 openEuler 开发者：熟悉 openEuler 开发贡献流程、关键特性、相关项目的开发等知识。
- 面向 openEuler 运维人员：熟悉 openEuler 常见或疑难问题的解决思路和方案、openEuler 系统管理知识和相关命令。

智能调优

功能描述

openEuler 大模型智能交互平台智能调优功能目前支持智能 shell 入口。

在上述功能入口，用户可通过与 openEuler 大模型智能交互平台进行自然语言交互，完成性能数据采集、系统性能分析、系统性能优化等作业，实现启发式调优。

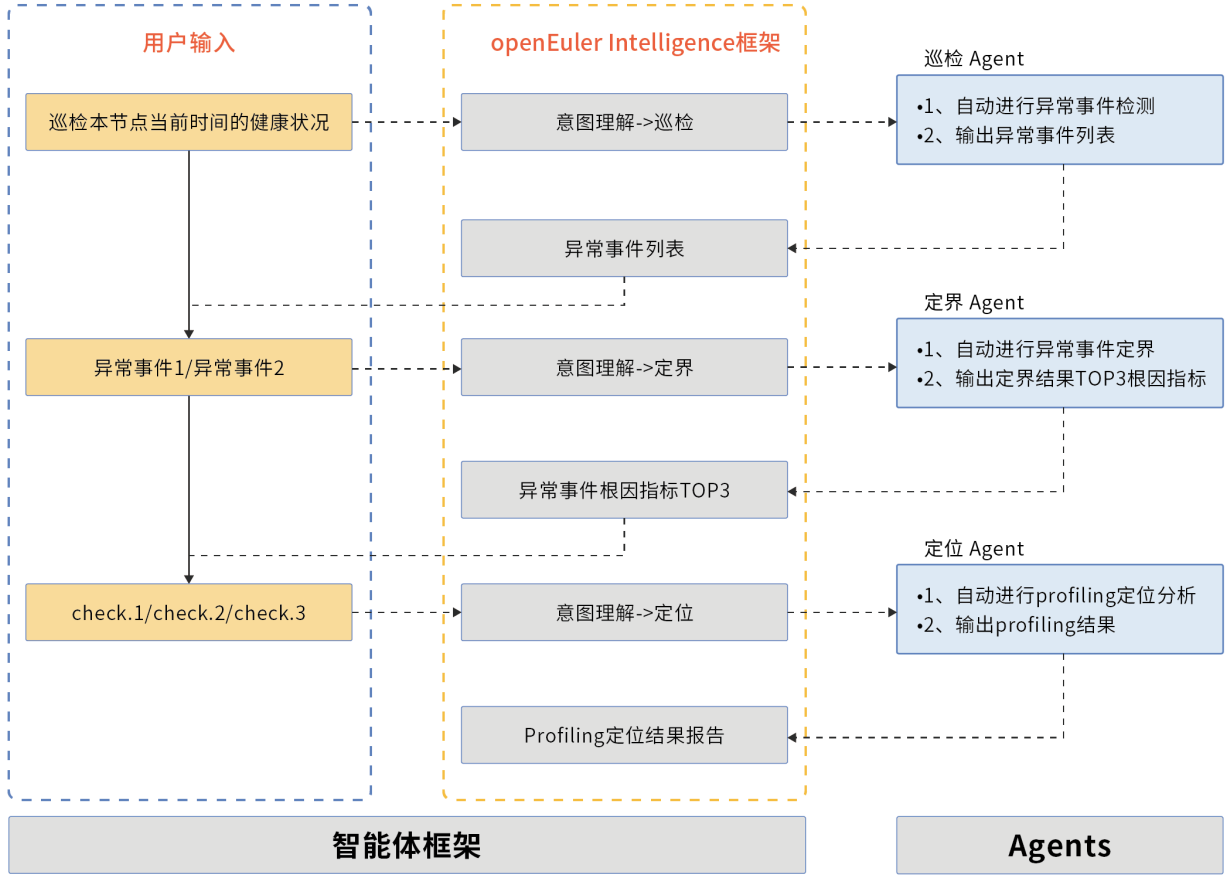


应用场景

- 快速获取系统重要性能指标数据：可快速获取当前系统中 CPU/IO/DISK/NETWORK 等多个重要维度的性能指标以及指定应用的性能指标，帮助用户快速了解系统性能数据。
- 分析系统性能状况：可生成性能分析报告，报告从 CPU/IO/DISK/NETWORK 等多个重要维度分析系统性能状况以及分析指定应用的性能状况，并提示当前系统可能存在的性能瓶颈。
- 推荐系统性能优化建议：可生成一键式执行的性能优化脚本，用户在审核脚本内容后，可执行该脚本，对系统及指定应用的配置进行优化。

智能诊断

功能描述



1. 巡检：调用 Inspection Agent，对指定 IP 进行异常事件检测，为用户提供包含异常容器 ID 以及异常指标（cpu、memory 等）的异常事件列表
2. 定界：调用 Demarcation Agent，对巡检结果中指定异常事件进行定界分析，输出导致该异常事件的根因指标 TOP 3
3. 定位：调用 Detection Agent，对定界结果中指定根因指标进行 Profiling 定位分析，为用户提供该根因指标异常的热点堆栈、热点系统时间、热点性能指标等信息

应用场景

智能诊断接口的功能范围是：具备单机异常事件检测 + 定界 + profiling 定位的能力。

- 其中检测能力指的是：进行单机性能指标采集、性能分析、异常事件检测。
- 其中定界能力指的是：结合异常检测结果进行根因定位，输出 top3 根因指标及其描述。
- 其中 profiling 定位能力指的是：结合 profiling 工具对根因指标进行具体问题模块（代码）定位。

智能化漏洞修补

功能描述

openEuler 智能化漏洞修补工具针对 openEuler 内核仓库，提供自动化智能漏洞管理与修复能力。该功能针对 openEuler 的 kernel 仓库，通过 /analyze 命令，来实现漏洞对 openEuler 操作系统版本的影响范围的分析；通过 /create_pr 命令，来实现补丁 pr 的分钟级自动创建。

应用场景

openEuler 智能化漏洞修补工具可广泛应用于系统安全维护和内核日常管理场景，具体应用如下：

- 漏洞影响范围分析：自动分析漏洞（CVE）对 openEuler 内核仓库各版本的影响情况，明确漏洞引入与修复状态。
- 漏洞补丁链接提供：智能匹配并提供漏洞引入与修复的具体补丁链接，帮助维护人员快速定位漏洞相关代码。
- 自动补丁 PR 创建：支持自动生成漏洞修复补丁，并创建相应的 PR，以便快速合入 openEuler 内核仓库的指定版本。
- 补丁冲突智能检测：自动识别和预警修复补丁与现有代码的兼容性冲突，确保修补过程顺畅、有效。

智能容器镜像

功能描述

openEuler 大模型智能交互平台目前支持通过自然语言调用环境资源，在本地协助用户基于实际物理资源拉取容器镜像，并且建立适合算力设备调试的开发环境。

当前版本支持三类容器，并且镜像源已同步在 dockerhub 发布，用户可手动拉取运行：

1. SDK 层：仅封装使能 AI 硬件资源的组件库，例如：cuda、cann 等。
2. SDK + 训练 / 推理框架：在 SDK 层的基础上加装 tensorflow、pytorch 等框架，例如：tensorflow2.15.0-cuda12.2.0、pytorch2.1.0.a1-cann7.0.RC1 等。
3. SDK + 训练 / 推理框架 + 大模型：在第 2 类容器上选配几个模型进行封装，例如 llama2-7b、chatglm2-13b 等语言模型。

当前支持的容器镜像汇总：

registry	repository	image_name	tag
docker.io	openeuler	cann	8.0.RC1-oe2203sp4
			cann7.0.RC1.alpha002-oe2203sp2
docker.io	openeuler	oneapi-runtime	2024.2.0-oe2403lts
docker.io	openeuler	oneapi-basekit	2024.2.0-oe2403lts
docker.io	openeuler	llm-server	1.0.0-oe2203sp3
docker.io	openeuler	mlflow	2.11.1-oe2203sp3
			2.16.2-oe2203sp3
			2.17.0rc0-oe2203sp3
			2.17.1-oe2203sp1
			2.17.1-oe2203sp3
			2.17.2-oe2203sp1
			2.17.2-oe2203sp3
			2.17.2-oe2203sp4
			2.18.0-oe2203sp1
			2.18.0-oe2403lts
			2.19.0-oe2203sp1
			2.19.0-oe2203sp3
			2.19.0-oe2203sp4
			2.19.0-oe2403lts
			2.17.1-oe2403lts
			2.17.2-oe2403lts
docker.io	openeuler	llm	chatglm2_6b-pytorch2.1.0.a1-cann7.0.RC1.alpha002-oe2203sp2
			llama2-7b-q8_0-oe2203sp2
			chatglm2-6b-q8_0-oe2203sp2
			fastchat-pytorch2.1.0.a1-cann7.0.RC1.alpha002-oe2203sp2
docker.io	openeuler	tensorflow	tensorflow2.15.0-oe2203sp2
			tensorflow2.15.0-cuda12.2.0-devel-cudnn8.9.5.30-oe2203sp2

docker.io	openeuler	pytorch	pytorch2.1.0-oe2203sp2
			pytorch2.1.0-cuda12.2.0-devel-cudnn8.9.5.30-oe2203sp2
			pytorch2.1.0.a1-cann7.0.RC1.alpha002-oe2203sp2
docker.io	openeuler	cuda	cuda12.2.0-devel-cudnn8.9.5.30-oe2203sp2

应用场景

- 面向 openEuler 普通用户：简化深度学习开发环境构建流程，节省物理资源的调用前提，比如实现在 openEuler 系统上搭建昇腾计算的开发环境。
- 面向 openEuler 开发者：熟悉 openEulerAI 软件栈，减少组件配套试错成本。

05 内核创新

openEuler 内核中的新特性

openEuler 25.03 基于 Linux Kernel 6.6 内核构建，在此基础上，同时吸收了社区高版本的有益特性及社区创新特性。

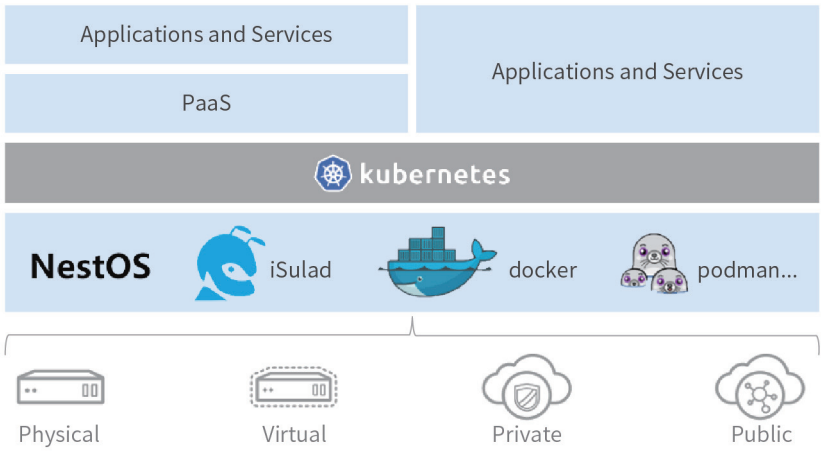
- 内核多副本（Kernel Replication）特性：Kernel Replication 旨在优化 Linux 内核在 NUMA（非一致性内存访问）架构下的内核性能瓶颈。相关研究表明，Apache、MySQL、Redis 等数据中心关键应用中，Linux 内核态的执行对整体性能影响显著，例如：内核执行占整个应用程序执行 CPU cycles 数目的 61%、总指令执行数目的 57%，占 I-cache 失效率的 61%、I-TLB 失效率的 46%。在传统 Linux 内核中，代码段、只读数据段、内核页表（swapper_pg_dir）仅驻留在主 NUMA 节点中并无法被迁移，这就导致当进程或多线程应用跨多个 NUMA 节点部署时，涉及系统调用存在频繁的跨 NUMA，导致整机访存延时增加，从而影响整机性能。Kernel Replication 特性扩展了 mm_struct 中的 pgd 全局页目录表，在内核启动阶段会自动创建内核代码段、数据段以及对应页表的各 NUMA 节点副本，此机制允许相同的内核虚拟地址在不同 NUMA 节点上映射到所在各自节点的物理地址，从而提升访存的本地局部性、减少跨 NUMA 性能开销。功能完备性上，Kernel Replication 特性同时支持 vmalloc，module 动态加载，Kprobe/KGDB/BPF 等动态指令注入机制，KPTI/KASLR/KASAN 等安全机制，64K 大页等。易用性上，新增内核启动阶段 cmdline 配置选项（默认关闭），可以在 boot 阶段进行动态开关，确保可控性和兼容性。Kernel Replication 适用于高并发、多线程的服务器负载场景。
- HAOC 3.0 安全特性：HAOC (Hardware-assisted OS compartmentalization) 含义是基于硬件辅助的操作系统隔离技术。 基于 x86、ARM 处理器提供的硬件特性，设计复式架构内核，在内核中提供隔离执行环境，为 Linux 内核提供进一步的隔离能力，从而阻止攻击者实施横向移动和权限提升。当前版本提供了隔离执行环境 IEE， 允许后续选项在隔离执行环境中新增需要隔离的敏感资源，并保证这些敏感资源不能被普通的内核代码随意的访问，而只能通过对 IEE 提供的接口进行调用来实现访问。

06 云化基座

NestOS 容器操作系统

NestOS 是在 openEuler 社区孵化的云底座操作系统，集成了 rpm-ostree 支持、ignition 配置等技术。采用双根文件系统、原子化更新的设计思路，使用 nestos-assembler 快速集成构建，并针对 K8S、OpenStack 等平台进行适配，优化容器运行底噪，使系统具备十分便捷的集群组建能力，可以更安全的运行大规模的容器化工作负载。

功能描述



1. 开箱即用的容器平台：NestOS 集成适配了 iSulad、Docker、Podman 等主流容器引擎，为用户提供轻量级、定制化的云场景 OS。
2. 简单易用的配置过程：NestOS 通过 ignition 技术，可以以相同的配置方便地完成大批量集群节点的安装配置工作。
3. 安全可靠的包管理：NestOS 使用 rpm-ostree 进行软件包管理，搭配 openEuler 软件包源，确保原子化更新的安全稳定状态。
4. 友好可控的更新机制：NestOS 使用 zincati 提供自动更新服务，可实现节点自动更新与重新引导，实现集群节点有序升级而服务不中断。
5. 紧密配合的双根文件系统：NestOS 采用双根文件系统的设计实现主备切换，确保 NestOS 运行期间的完整性与安全性。

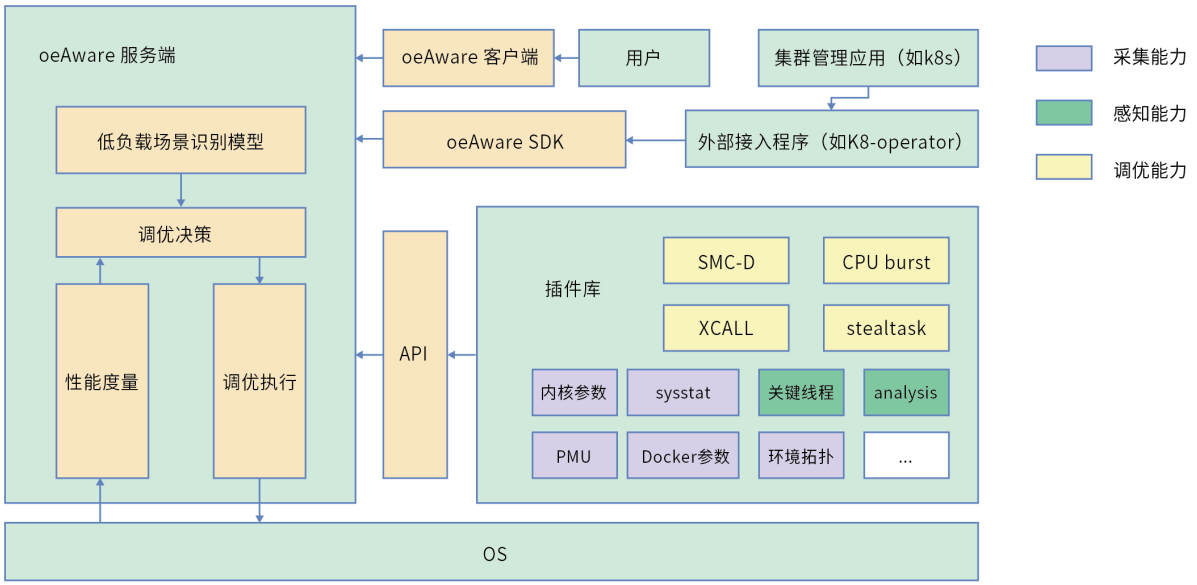
应用场景

NestOS 适合作为以容器化应用为主的云场景基础运行环境，引入社区孵化项目 NestOS-Kubernetes-Deployer，辅助 NestOS 解决在使用容器技术与容器编排技术实现业务发布、运维时与底层环境高度解耦而带来的运维技术栈不统一，运维平台重复建设等问题，保证了业务与底座操作系统运维的一致性。

07 特性增强

oeAware 支持瓶颈评估一键推荐调优等特性增强

oeAware 是在 openEuler 上实现低负载采集感知调优的框架，目标是动态感知系统行为后智能使能系统的调优特性。传统调优特性都以独立运行且静态打开关闭为主，oeAware 将调优拆分为采集、感知和调优三层，每层通过订阅方式关联，各层采用插件式开发尽可能复用。



功能描述

oeAware 的每个插件都是按 oeAware 标准接口开发的动态库，包含若干个实例，每个实例可以是一个独立的采集、感知或调优功能集，每个实例包含若干个 topic，其中 topic 主要用于提供采集或者感知的数据结果，这些数据结果可供其他插件或者外部应用进行调优或分析。新增 transparent_hugepage_tune 和 preload_tune 插件实例。

- SDK 提供的接口可以实现订阅插件的 topic，回调函数接收 oeAware 的数据，外部应用可以通过 SDK 开发定制化功能，例如完成集群各节点信息采集，分析本节点业务特征。
- PMU 信息采集插件：采集系统 PMU 性能记录。
- Docker 信息采集插件：采集当前环境 Docker 的一些参数信息。
- 系统信息采集插件：采集当前环境的内核参数、线程信息和一些资源信息（CPU、内存、IO、网络）等。
- 线程感知插件：感知关键线程信息。
- 评估插件：分析业务运行时系统的整体状态，给用户推荐使用的调优方式。
- 系统调优插件：（1）stealtask：优化 CPU 调优 （2）smc_tune（SMC-D）：基于内核共享内存通信特性，提高网络吞吐，降低时延 （3）xcall_tune：跳过非关键流程的代码路径，优化 SYSCALL 的处理底噪（4）transparent_hugepage_tune：使能透明大页，提高 tlb 命中率 （5）preload_tune：无感加载动态库
- Docker 调优插件：利用 cpuburst 特性在突发负载下环境 CPU 性能瓶颈。

约束限制

- SMC-D：需要在服务端客户端建链前，完成使能 smc 加速。比较适用于长链接多的场景。
- Docker 调优：暂不适用于 K8s 容器场景。
- xcall_tune : 内核配置选项 FAST_SYSCALL 打开。

应用场景

stealtask 适用于希望提高 CPU 利用率的场景，例如 Doris，该调优实例可以有效提高 CPU 利用，避免 CPU 空转。

XCALL 适用于应用程序的 SYSCALL 开销较大的场景，XCALL 提供一套跳过非关键流程的代码路径，来优化 SYSCALL 的处理底噪，这些被跳过的非关键流程会牺牲部分维测和安全功能。

SMC-D 特别适用于需要高吞吐量和低延迟的应用场景，如高性能计算（HPC）、大数据处理和云计算平台。通过直接内存访问（DMA），SMC-D 能够显著减少 CPU 负载并提升交互式工作负载的速率。

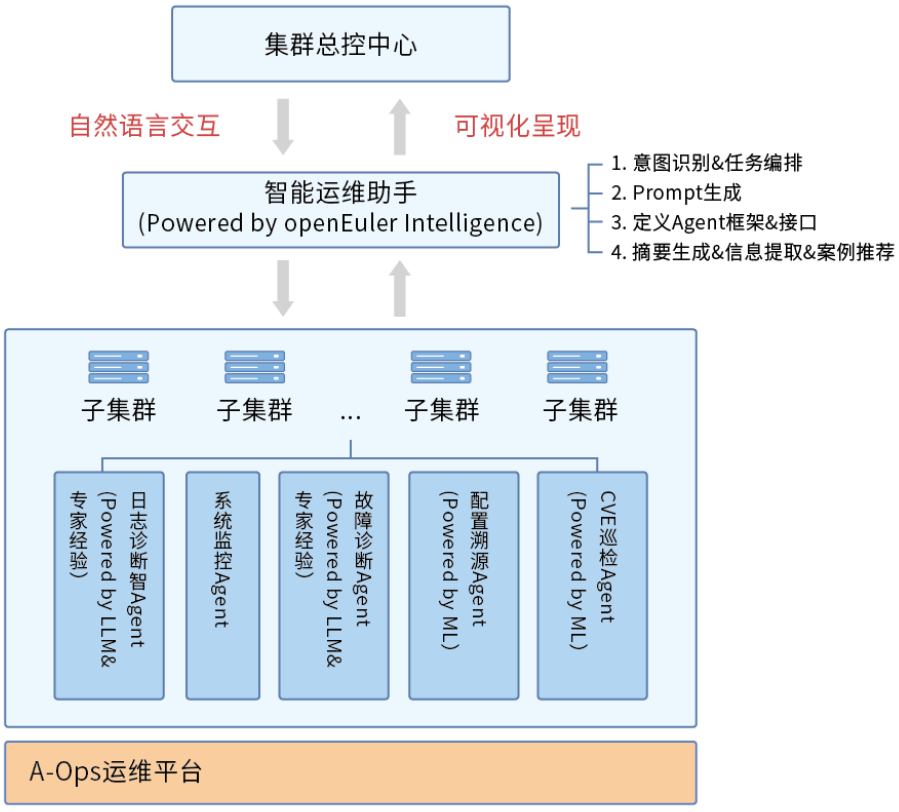
cpuburst 适用于高负载容器场景，例如 Doris，能够缓解 CPU 限制带来的性能瓶颈。

透明大页适用于 tlb miss 较高的应用场景。

Aops cve 修复 / 配置溯源实现运维智能化，支持主动通知和图文混合交互

aops 运维智能化，通过对话交互实现运维操作以替代传统运维操作方式降低运维门槛，导航式操作，简化操作。现交互场景支持 cve 提示和修复推荐，配置溯源配置异常追溯和配置基线同步，通过和助手交互，让运维助手帮忙执行日常运维操作。

功能描述

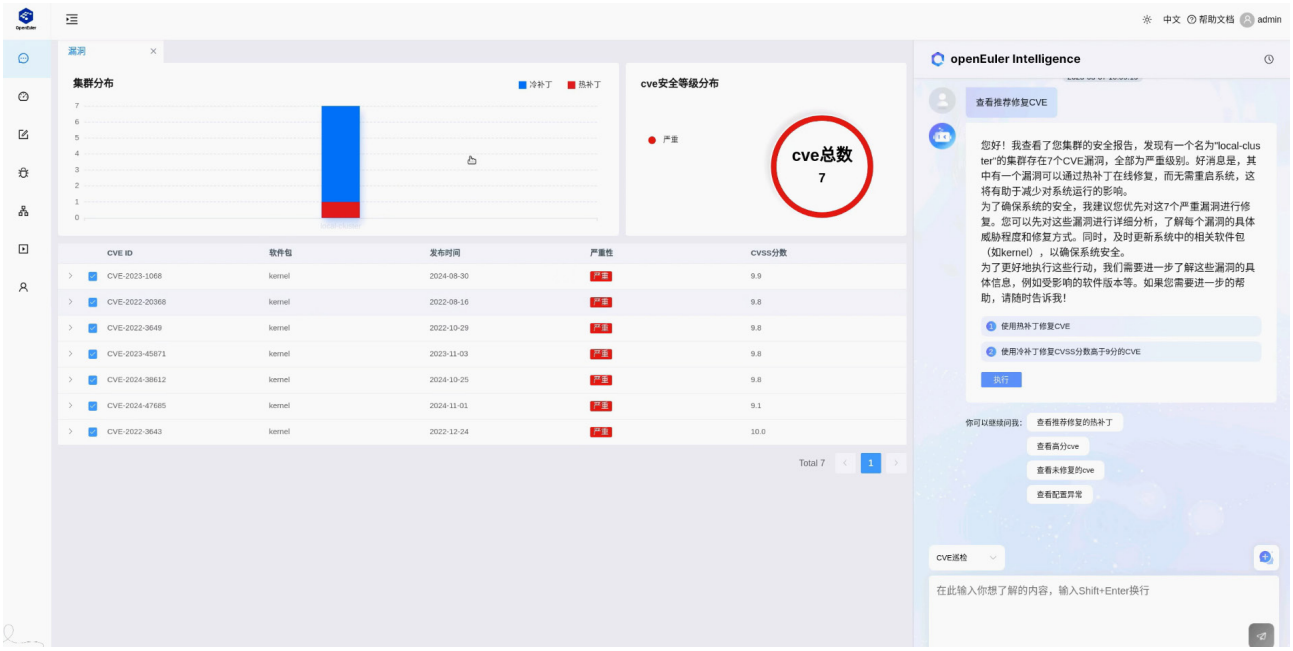


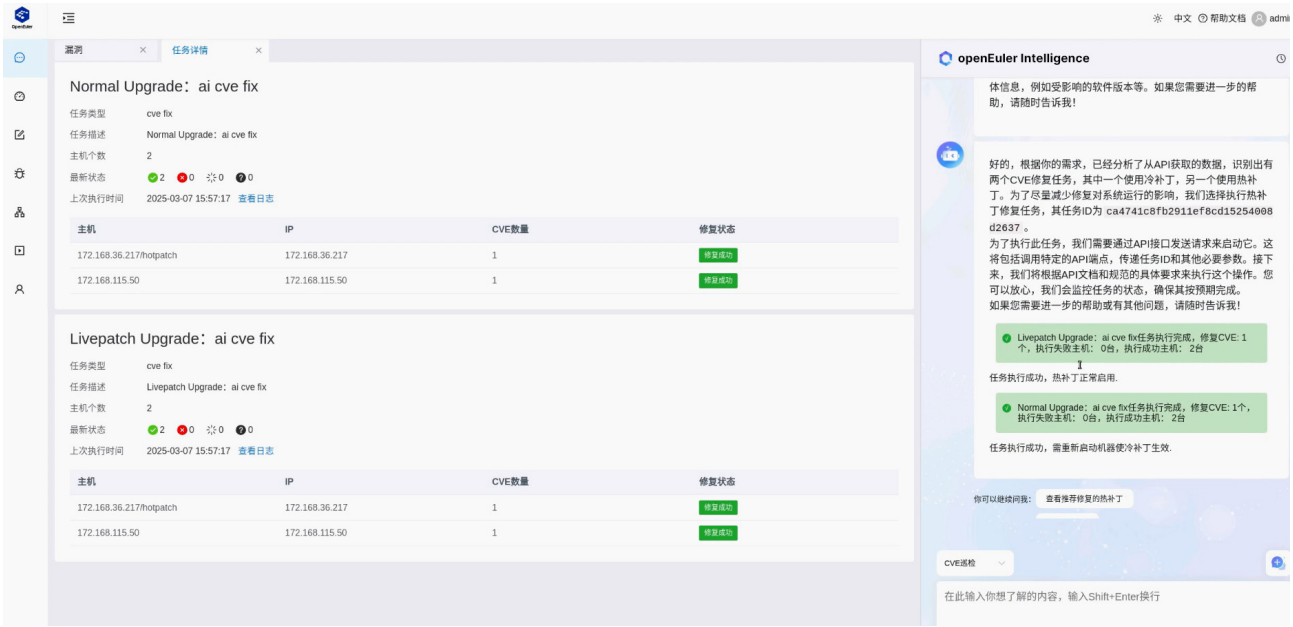
aops 基于 openEuler 大模型智能交互平台集成智能运维助手，实现 cve 修复和配置溯源操作智能化：

- cve 修复：aops 自动显示集群 cve 情况，筛选高分高严重等级 cve 进行推荐并提供修复方式，用户通过助手辅助和页面混合交互实现 cve 的修复和结果查看。
- 配置溯源：通过助手查询基线配置异常的机器，页面展示异常机器和异常配置项，助手进行智能总结并提供推荐修复方式，可通过页面混合交互进行修复。

应用场景

cve 修复





配置异常检测

Aops cve 修复和配置溯源实现运维智能化，通过对话交互实现运维操作以替代传统运维操作方式，协助运维人员日常运维，日常运维提供专家级指导，关联当前操作提供推荐操作，简化运维上手难度，提高使用体验。

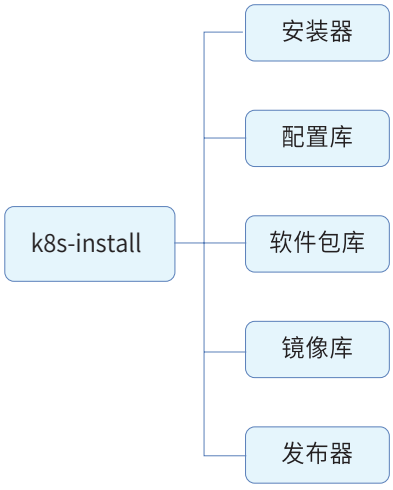
云原生基础设施部署升级工具 k8s-isntall 加入版本

K8s-install 是一款在多个发行版、多个架构上生成云原生基础设施的在线安装工具，及离线安装包生成工具。支持多维度地一键安装、部署、安全更新云原生基础设施套件，极大地节省部署和适配的时间，并使得该流程标准化，可追踪。

k8s-install 项目产生背景与意义：

- 欧拉本身云原生工具链版本过低，同一发行版内没有维护多版本基线（如 k8s 1.20 1.25 1.29 等），已发布分支不能更新大版本，业务需要更高版本时就需要自主适配维护；
- 业务方经常采用 Ansible 等工具部署云底座，使用非本发行版软件包甚至是静态二进制文件及 tar 包部署，无法支持 CVE 安全更新；
- 离线安装与在线安装版本同步问题突出，离线包升级及改动困难；
- 安装布署流程不标准，各组件版本不统一，不同组件版本的不适配、配置存在差异情况时常发生，且解决非常耗时，难以定位问题源头。

k8s-install 项目架构



- 安装器：用来解决 runc、containerd、docker、k8s 这几个组件及其依赖的系统库的检测、安装、更新流程；
- 配置库：用来储存 docker k8s 软件的配置文件模板；
- 软件包库：用来储存 runc、containerd、docker、k8s 及其依赖的系统库的各个版本的各个架构的 rpm 软件包；
- 镜像库：用来储存 k8s 软件启动时所需要的各版本的 kube-apiserver、 kube-scheduler、 etcd、 coredns 等镜像，也包含基础网络插件如 flannel 的镜像；
- 发布器：用来将最新的代码脚本、rpm、镜像、配置封装在一起生成在线安装包和离线安装包的工具。
- K8s-install 主体程序使用 bash 写成，不需要编译链接，在线安装包被封装为 rpm 包，它的打包采用 spec 文件方式。

模块 1：k8s-install 安装器

k8s-install 是用于云原生基础设施的安装与安全更新工具。

解决版本适配难题： 欧拉上游云原生工具链版本过低，且已发布分支不能更新大版本，业务版本需要自主适配维护。k8s-install 支持多版本基线，可满足业务刚性需求，避免因版本不兼容导致的部署失败或功能异常。

提升部署效率与标准化： 各部门或项目安装布署流程不标准，组件版本不统一，导致适配问题频发且解决耗时。使用 k8s-install 能实现标准化部署，确保各组件版本兼容，减少问题定位时间，提升整体部署效率。

增强安全性与可维护性： 业务方原采用静态二进制及 tar 包部署，无法支持 CVE 安全更新。k8s-install 可实现 CVE 漏洞更新，及时修复安全漏洞，保障系统安全稳定。同时，全组件代码完成公司内建仓及欧拉建仓，利于版本追溯与问题定位，增强系统可维护性。

促进社区开源与协作： 兼容 openeuler 发行版， 开源建仓并活跃更新，促进技术共享与社区生态发展，吸引更多开发者参与，提升项目影响力，推动云原生技术的持续进步。

功能描述

具备以下核心功能：

多发行版本适配：支持 k8s 1.20、1.25、1.29 等多基线版本，可满足不同业务场景下的版本需求，实现按需部署。

多架构支持：兼容 x86_64、aarch64、loongarch64 等多种架构，适用于 diverse 硬件环境，拓展应用范围。

多组件管理：涵盖 golang、runc、containerd、docker、k8s 及相关组件的一体化安装与配置，简化复杂组件的部署流程，提升效率。

在线离线部署：提供在线安装器 k8s-install、离线安装器 k8s-install-offline，结合 publish.sh 发布器，可灵活应对网络条件，保障部署稳定性。

应用场景

资源池集群快速部署：在线或离线资源池利用 k8s-install 进行一键安装、创建集群、结点加入集群。

资源池 CVE 漏洞更新：多个资源池利用 k8s-install 进行 CVE 漏洞更新，及时修复安全问题，保障系统安全稳定运行。

问题排查：借助该工具排查解决 docker 和 k8s 相关问题，快速定位故障点，提升运维效率。

社区开源与建仓：在 openeuler 开源、建仓并活跃更新，促进技术共享与社区生态发展，提升项目影响力。

模块 2：k8s-install publish 发布器

publish.sh 是 k8s-install 工具链中的发布器。它具体以下功能：

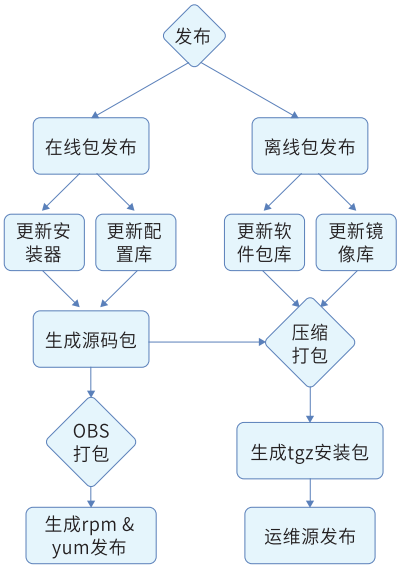
保障离线部署可行性： 在某些网络受限或无网络连接的环境中，如部分数据中心或特殊生产环境，无法直接从在线仓库获取资源。publish.sh 可生成完整的离线安装包，确保在这些场景下也能顺利部署云原生基础设施，拓展工具的适用范围。

实现高效版本迭代与发布管理： 随着 k8s-install 工具及组件的不断更新，publish.sh 能实现自动化构建、测试与发布流程，提高版本迭代效率，确保新版本及时、准确地推送给用户，促进系统的持续演进。

提升资源获取的稳定性与可靠性： 在线仓库可能因网络波动、资源更新不及时等原因导致获取的软件包或镜像存在问题。publish.sh 从官方或可信的在线仓库拉取资源，并经过整合与测试，能保障所发布的资源稳定可靠，避免因资源问题引发的部署失败。

促进多团队协作与资源同步： 在大型项目中，不同团队可能负责不同的组件或模块。publish.sh 可将各团队的更新成果整合发布，确保各团队获取的资源一致，利于协同工作，提升整体项目进度与质量。

功能描述



主要功能包括：

离线包生成与发布： 从在线 yum 和镜像仓库中拉取最新软件包、镜像，结合最新配置文件和安装器，打包成离线 tgz 安装包，满足离线环境的部署需求。

在线代码更新与发布： 将更新的代码上传至 git 仓库，选取配置库和安装器进行打包源码，经本地 build 测试后上传至 obs 服务器进行正式编译，并发布到 yum 源，实现在线资源的更新与同步。

应用场景

版本迭代与发布管理： 在 k8s-install 代码、RPM 包、或组件镜像有更新时，通过 publish.sh 实现高效、可靠的发布管理，确保新版本能够及时、准确地拉取更新的软件包与镜像、生成离线安装包、包上传、打包测试、源码 push 等工作。帮助系统开发及发布的高效、可持续迭代演进。

GCC for openEuler

GCC for openEuler 基线版本已经从 GCC 10.3 升级到 GCC 12.3 版本，支持自动反馈优化、软硬件协同、内存优化、SVE 向量化、矢量化数学库等特性。

1. GCC 版本升级到 12.3，默认语言标准从 C14/C++14 升级到 C17/C++17 标准，支持 Armv9-a 架构，X86 的 AVX512 FP16 等更多硬件架构特性。

	GCC 10.3.0	GCC 11.3.0	GCC 12.3.0
发布时间	2021/4/8	2022/4/21	2023/5/8
C 标准	默认 c17 支持 c2x	默认 c17 支持 c2x	默认 c17 支持 c2x
C++ 标准	默认 c++14 支持 c++17 实验性 C++2a 改进 支持部分 C++20	默认 c++17 实验性 C++2a 改进 支持部分 C++20	默认 c++17 实验性 C++2a 改进 支持部分 C++20
架构新特性	armv8.6-a(bfloat16 extension/Matrix Multiply extension) SVE2 Cortex-A77 Cortex-A76AE Cortex-A65 Cortex-A65AE Cortex-A34	armv8.6-a, +bf16, +i8mm armv8.6-r Cortex-A78 Cortex-A78AE Cortex-A78C Cortex-X1	armv8.7-a, +ls64 atomic load and store armv8.8-a, +mop, accelerate memory operations armv9-a Ampere-1 Cortex-A710 Cortex-X2 AVX512-FP16 SSE2-FP16

2. 支持结构体优化，指令选择优化等，充分使能 ARM 架构的硬件特性，运行效率高，在 SPEC CPU 2017 等基准测试中性能大幅优于上游社区的 GCC 12.3 版本。
3. 支持自动反馈优化特性，实现应用层 MySQL 数据库等场景性能大幅提升。

功能描述

- 支持 ARM 架构下 SVE 矢量化优化，在支持 SVE 指令的机器上启用此优化后能够提升程序运行的性能。
- 支持内存布局优化，通过重新排布结构体成员的位置，使得频繁访问的结构体成员放置于连续的内存空间上，提升 Cache 的命中率，提升程序运行的性能。
- 支持 SLP 转置优化，在循环拆分阶段增强对连续访存读的循环数据流分析能力，同时在 SLP 矢量化阶段，新增转置操作的分析处理，发掘更多的矢量化机会，提升性能。
- 支持冗余成员消除优化，消除结构体中从不读取的结构体成员，同时删除冗余的写语句，缩小结构体占用内存大小，降低内存带宽压力，提升性能。
- 支持结构体成员静态压缩优化，减小结构体整体占用的内存大小来提高 cache 命中率。
- 支持数组比较优化，实现数组元素并行比较，提高执行效率。
- 支持 ARM 架构下指令优化，增强 ccmp 指令适用场景，简化指令流水。

- 支持 if 语句块拆分优化，增强程序间常量传播能力。
- 增强 SLP 矢量优化，覆盖更多矢量化场景，提升性能。
- 支持自动反馈优化，使用 perf 收集程序运行信息并解析，完成编译阶段和二进制阶段反馈优化，提升 MySQL 数据库等主流应用场景的性能。

应用场景

通用计算领域，运行 SPECCPU 2017 测试，相比于上游社区的 GCC 10.3 版本可获得 20% 左右的性能收益。

其他场景领域，使能自动反馈优化后，MySQL 性能提升 15% 以上；使能内核反馈优化后，实现 Unixbench 性能提升 3% 以上。

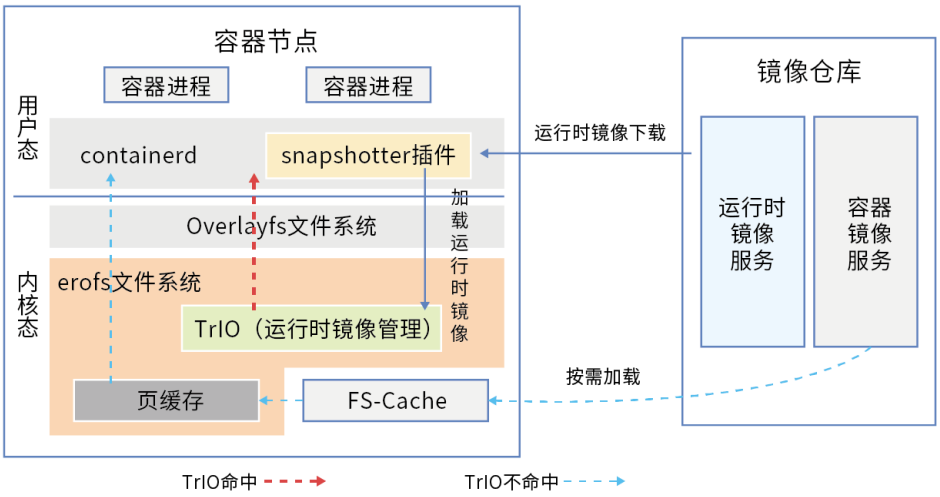
Trace IO 加速容器快速启动

Trace IO 简称 TrIO，用于优化基于 EROFS over fscache 的容器镜像按需加载场景。它通过对容器启动过程中的 IO 进行精确的追踪，并将 IO 高效编排进入容器镜像中来优化容器冷启动过程。相比于现有的容器镜像加载方案，TrIO 能使得容器任务的冷启动的时延更低、带宽利用率更高效。TrIO 涉及内核和用户态模块，内核态在 EROFS 文件系统下做了相应的适配，用户态模块提供了抓取容器运行时的 IO trace 的工具、并给出了一种基于 nydus-snapshotter 的适配修改指导，使得容器使用者无需更改 containerd 及 runc，从而做到对现有容器管理工具的兼容。

TrIO 的核心优势是实现容器按需加载过程中的 IO 聚合，通过编排容器任务运行时 IO 轨迹的方式精确的拉取容器运行过程中所需要的 IO 数据；这极大提高了容器启动过程中拉取镜像数据的效率，从而实现容器启动阶段的低时延。

功能描述

TrIO 的功能包括两个部分：容器运行时 IO 的抓取和容器启动过程运行时 IO 的使用。容器运行时 IO 的抓取通过使用 ebpf 技术，在文件系统侧进行 IO 追踪，从而获得容器任务启动过程中的 IO 读取请求，并将这些 IO 请求对应的数据进行编排，从而构建最小运行时镜像。容器启动过程，自定义的 snapshotter 插件模块会采用大 IO 的方式对最小运行时镜像镜像拉取，并导入到内核中；后续容器任务执行过程中所有的 IO 会优先从最小运行时镜像中读取。利用 TrIO 启动容器的执行流程可用下图进行表示：



相比现有容器按需加载方案，TrIO 具有如下功能优势：

- 无 IO 放大：通过对运行时 IO 精确抓取并用于任务启动，对容器任务的启动过程中的 IO 无放大。
- IO 聚合: 容器任务启动过程通过大 IO 的方式将容器任务启动过程所需要的数据一次性拉取到容器节点，提高镜像数据加载效率，降低启动时延。

应用场景

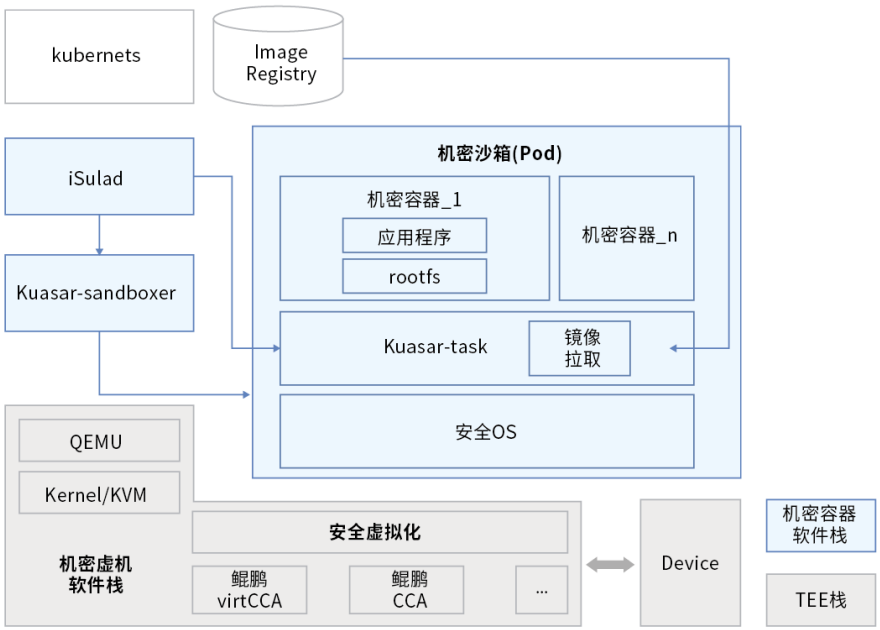
TrIO 的核心优势是实现 IO 的聚合，将容器镜像按需加载过程中的离散网络 IO 进行高效聚合；因此非常适合容器启动服务启动过程中有大量离散 IO 的场景，这种场景下可以实现容器镜像按需加载性能的进一步提升。

机密容器 Kuasar 适配 virtCCA

Kuasar 机密容器特性基于鲲鹏 920 系列 virtCCA 能力。北向可对接 iSulad 容器引擎，南向适配鲲鹏 virtCCA 硬件，实现鲲鹏机密计算无缝对接云原生技术栈。

Kuasar 机密容器充分利用 Sandboxer 架构优势，提供了高性能、低开销的机密容器运行时。Kuasar-sandboxer 集成 openEuler QEMU 的 virtCCA 能力，负责管理机密沙箱的生命周期，允许用户在机密硬件上创建机密沙箱，确保机密容器运行在可信执行环境中。

Kuasar-task 提供 Task API 接口，允许 iSulad 直接管理机密沙箱内容器的生命周期。机密容器的镜像通过 Kuasar-task 的镜像拉取能力直接从镜像仓拉入机密沙箱的加密内存中，保障了容器镜像的机密性。



技术约束

1. Kuasar 机密容器尚未支持远程验证服务，将在 openEuler 2403 LTS 增强扩展版本中集成 secGear 组件，完成对远程验证服务的支持。
2. 暂不支持镜像加解密，在完成对 secGear 组件的对接后，可以支持镜像将解密。

功能描述

Kuasar 统一容器运行时在支持安全容器的基础上添加了对机密容器的支持。用户可以通过配置 iSulad 的运行时参数，完成对 Kuasar 机密容器的纳管。

支持功能特性：

- 支持 iSulad 容器引擎对接 Kuasar 机密容器运行时，兼容 Kubernetes 云原生生态。
- 支持基于 virtCCA 的机密硬件，允许用户在鲲鹏 virtCCA 可信执行环境中部署机密容器。

应用场景

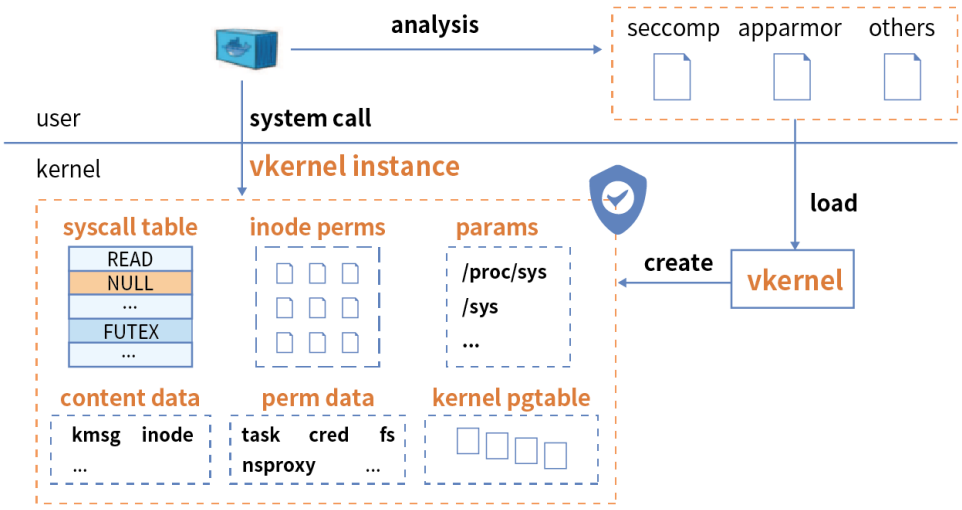
Kuasar 机密容器在满足客户数据安全的诉求下，兼容云原生生态，确保用户的机密应用具备高可用性、弹性伸缩、快速交付等云原生优势，在 AI 保护、数据可信流通、隐私保护等机密计算的业务中有广泛的应用场景。

引入 vkernel 概念增强容器隔离能力

Vkernel (Virtual Kernel) 全称虚拟内核技术，旨在打破容器场景下的内核共享的局限性，实现独立的虚拟内核以完善隔离能力，同时兼顾容器性能。

功能描述

Vkernel 构建独立的系统调用表、独立的文件权限表，增强容器的基础安全防护能力；构建独立的内核参数，允许容器定制宏观资源策略和微观资源参数；还会通过划分内核信息数据的归属、结合硬件特性保护内核权限数据、构建独立内核页表保护用户数据等，进一步强化安全能力。在未来，Vkernel 还将探索与性能干扰相关内核数据，以增强容器性能隔离能力。



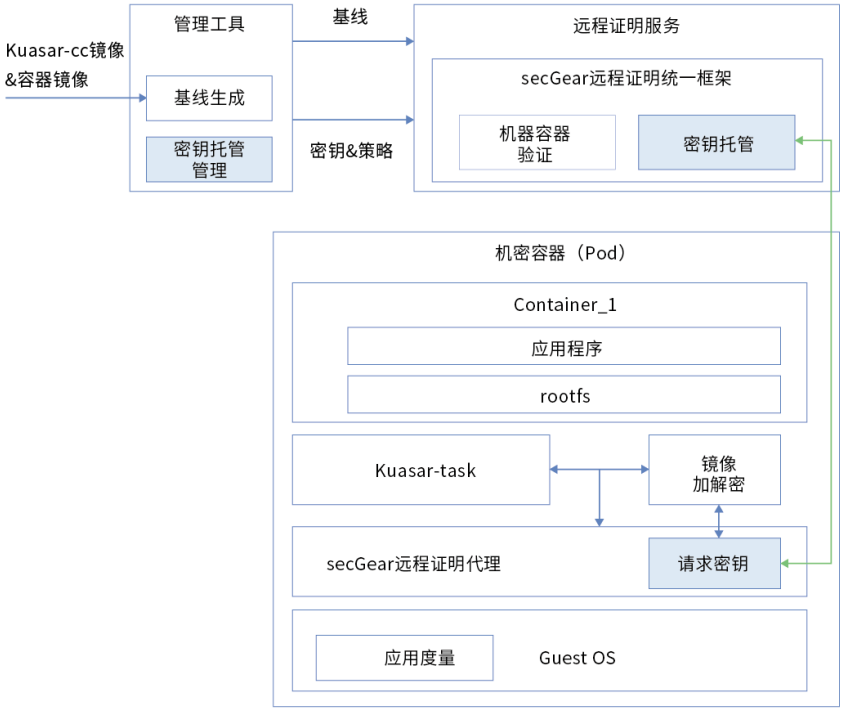
应用场景

Vkernel 特性可以提升容器场景下的安全隔离与功能定制能力。

- 安全防护：支持更灵活的系统调用过滤策略，如参数解引用 ; 基于 inode 标记快速检索文件，可自定义文件拦截处理手段
- 数据隔离：仅允许容器访问自身内核日志，避免敏感日志信息泄漏
- 资源定制：限制系统资源用量，如 fd，避免恶意容器耗尽；配置独立的内核参数 , 支持配置冲突的负载并存 ; 单独进行内核参数调优，如透明大页，提升特定负载性能

secGear 支持机密容器镜像密钥托管

secGear 远程证明服务提供机密容器镜像密钥托管能力，重点构建覆盖密钥安全存储、动态细粒度授权、跨环境协同分发的管理体系，并借助零信任策略与自动化审计能力，在确保数据机密性与操作可追溯性的同时，实现密钥治理复杂度与运维成本的最优平衡，为云原生环境提供“默认加密、按需解密”的一体化防护能力。



功能描述

secGear 整合远程证明技术构建分层密钥托管功能体系。

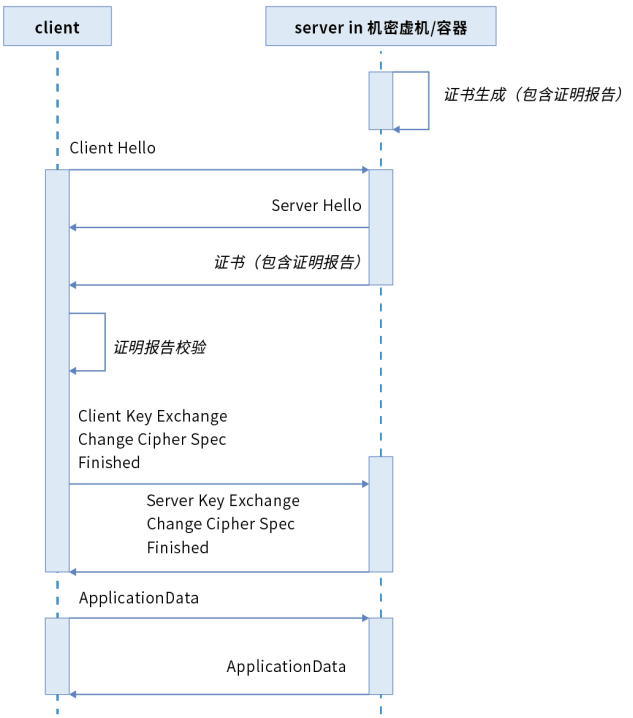
- 证明服务：通过构建集中式密钥托管服务端，基于机密执行环境（TEE）远程证明机制实现对加密镜像密钥的安全存储与生命周期管理，并面向授权用户提供精细化策略配置接口；
- 证明代理：机密计算节点内部署轻量级证明代理组件，提供本地 restAPI 接口。机密容器运行时通过调用证明代理接口完成机密执行环境的完整性验证，并与服务端建立动态会话实现密钥的安全传输。

应用场景

secGear的密钥托管功能与Kuasar容器运行时形成完整的机密容器解决方案，适用于任何需要高度数据安全和隐私保护的场景，尤其是在金融、医疗、政府、云计算、物联网等领域。通过容器镜像加密和密钥托管功能，可以有效保护容器镜像的完整性，减少容器数据泄露风险，防止供应链攻击，提升用户对云原生厂商的信任程度。

构建基于远程证明的 TLS 协议（RA-TLS）

RA-TLS 是将机密计算的远程证明嵌入到 TLS 协商过程中，保护隐私数据安全传入机密计算环境，同时简化机密计算安全通道的建立，降低机密计算的使用门槛，具体流程如下：



功能描述

单向认证：TLS 服务端部署在机密计算环境，客户端部署在普通环境，RA-TLS 协商过程中，基于远程证明对服务端机密计算环境及应用的合法性验证通过后，再做 TLS 密钥协商。

双向认证：TLS 服务端和客户端都部署在机密计算环境，RA-TLS 协商过程中，基于远程证明相互验证对端机密计算环境及应用的合法性，再做 TLS 密钥协商。

技术约束：机密计算环境能够访问网络（如 virtCCA）。

应用场景

RA-TLS 可广泛应用于使用机密计算的场景，尤其在云计算、大数据、金融、医疗等行业，用户基于 RA-TLS 验证云上机密计算环境及应用的合法性，并建立加密传输通道，再将隐私数据传入到机密环境中处理，从而保护用户隐私数据免遭非授权实体的访问。

openAMDC 提供高性能内存数据缓存及 KV 数据存储

openAMDC（open Advanced in-Memory Data Cache）是一款内存数据缓存软件，主要用于在内存中存储、缓存数据，起到数据访问加速、提升应用访问并发和降低数据访问延迟的作用，也可作为消息代理、内存数据库使用。

功能描述

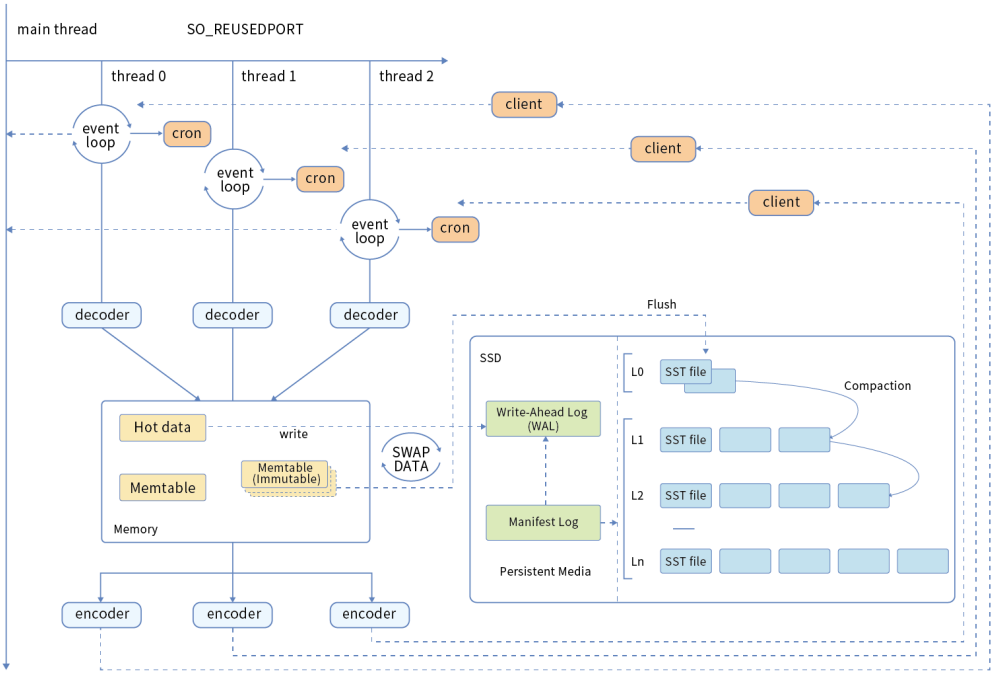
1. 核心能力

   openAMDC 兼容 RESP 协议，提供完整的数据缓存功能，支持 string、list、hash、set 等主流数据格式，提供主备、集群和哨兵等部署模式。具体功能如下表所示。

功能	功能说明
多数据类型缓存	提供 string、list、hash、set、sortedset、GEO、hyperloglog、stream 的数据缓存类型。
发布订阅	实现了发布订阅功能，增强系统功能性
ACL 权限管控	为服务提供了安全的访问机制，支持细粒度的访问权限控制。
IP 白名单	支持 IP、网段的白名单过滤，提高安全性。
内存数据淘汰策略	提供多种数据淘汰策略，满足多种数据淘汰要求，提高内存利用率。
持久化	为缓存核心提高可用性，防止在宕机时丢失数据。
Lua 脚本支持	支持使用 lua 脚本操作缓存核心。
SSL 支持	支持使用 SSL 加密通信。
主从模式	支持主从备份。
哨兵模式	为主从模式提供节点监控、自动故障转移、故障通知、配置传播功能。
集群模式	支持弹性伸缩（内存的扩 / 缩容），并且同时具备了故障转移功能。

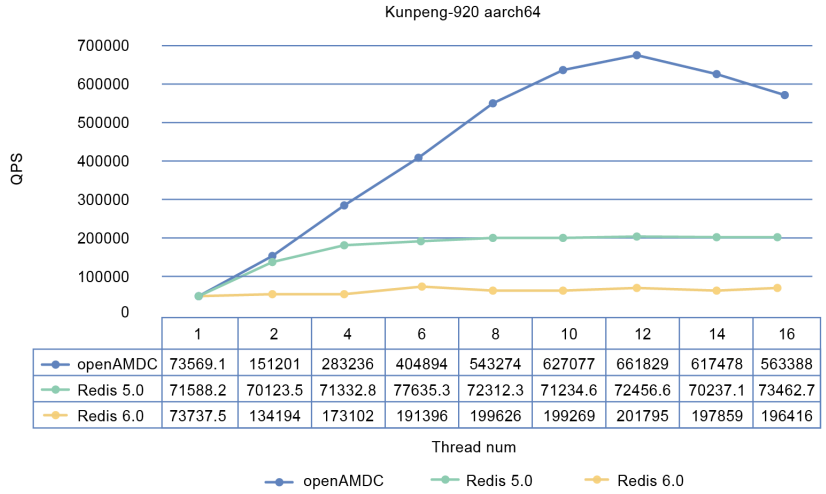
2. 架构特性

   openAMDC 采用了多线程架构实现内存数据缓存性能的大幅提升，并提供冷热数据交替模式支持内存与硬盘的混合数据存储。其架构如下图所示。



openAMDC 架构

1）多线程架构：openAMDC 在启动时会初始化多个 worker 线程，每个 worker 线程中会启动一个事件循环用于监听事件，为每个 listen 监听设置 SO_REUSEPORT 选项，启用网络连接内核负载均衡，允许多个线程绑定相同的端口号，相当于每个线程拥有队里的 listen socket 完成队列，避免了共享 listen socket 的资源争抢，提升了并发吞吐。



	1	2	4	6	8	10	12	14	16
openAMDC	73569.1	151201	283236	404894	543274	627077	661829	617478	563388
Redis 5.0	71588.2	70123.5	71332.8	77635.3	72312.3	71234.6	72456.6	70237.1	73462.7
Redis 6.0	73737.5	134194	173102	191396	199626	199269	201795	197859	196416

openAMDC 的性能提升

2）数据交换架构：openAMDC 在多线程架构的基础上扩展数据交换功能，实现数据冷热多级存储，降低缓存的综合使用成本。

应用场景

openAMDC 支持基于内存对数据访问加速的各种场景，如：热点数据缓存、分布式锁、数据共享、轻量级消息队列、位操作、限时操作等。具体场景介绍如下表所示。

应用场景	功能描述
热点数据缓存	拥有高性能的数据缓存能力，为大规模、高并发、高可用的关键应用提供安全可靠的缓存支撑能力，保障系统的正常、高效运行。
分布式锁	为分布式系统提供加锁操作，防止出现多个节点同时对数据做出操作而导致数据错误的问题。
数据共享	openAMDC 是分布式的独立服务，缓存的通用数据可以在分布式系统或多个应用、服务之间共享。
轻量级消息队列	能够通过发布订阅模式 /LIST 数据类型 /STREAM 数据类型快速实现轻量级的消息队列。
位操作（大数据处理）	用于超大数据量场景下的数据统计、去重等业务中，例如在线用户统计，留存用户统计等等。
限时操作	拥有 TTL 机制，可以实现一些限时业务，在规定时间内完成操作，否则该操作将失效。

支持 OpenStack Antelope 版本

OpenStack 是一个开源的云计算管理平台项目，旨在提供一个可扩展的、灵活的云计算服务，支持私有云和公有云环境。

功能描述

OpenStack 提供了一系列的服务和工具，用于构建和管理公共云、私有云和混合云。其主要功能包括：

- 计算服务：提供虚拟机的创建、管理和监控等功能。它允许用户快速创建、部署和销毁虚拟机和容器实例，从而实现对计算资源的灵活管理和高效利用。
- 存储服务：提供对象存储、块存储和文件存储等多种存储服务。块存储服务（如 Cinder）允许用户动态分配和管理持久性块存储设备，如虚拟机硬盘。对象存储服务（如 Swift）则提供了可扩展的、分布式的对象存储解决方案，用于存储大量非结构化数据。
- 网络服务：提供虚拟网络的创建、管理和监控等功能，包括网络拓扑规划、子网管理、安全组配置等，这使得用户能够轻松构建复杂的网络结构，并确保网络的安全性和可靠性。
- 身份认证服务：提供用户、角色和权限等身份管理功能，管理用户、角色和权限的访问控制。这使得用户能够安全地访问和管理云资源，并确保数据的机密性和完整性。
- 镜像服务：提供虚拟机镜像的创建、管理和共享等功能，包括创建、上传、下载和删除镜像。这使得用户能够轻松地创建和管理虚拟机镜像，并快速部署新的虚拟机实例。
- 编排服务：提供自动化部署和管理应用程序的功能，支持多个服务之间的协作和集成。通过编排服务（如 Heat），用户可以定义应用程序的部署模板，并自动执行相关的部署和管理任务。

应用场景

OpenStack 的应用场景主要包括以下几种情况：

- 私有云：拥有私有云环境的企业可以根据自身的需求和 IT 资源的现状，选择 OpenStack 部署在自己的数据中心或云环境中。这可以实现资源集中管理、自动化部署和弹性扩展等功能，同时 OpenStack 还可以为企业提供强大的安全保障措施，如访问控制、数据加密和审计日志等。
- 公有云：OpenStack 也可以为公有云提供资源池，解决多租户云环境中不同用户之间的性能和隔离问题。在这种场景下，OpenStack 可以为公有云提供弹性计算、容器、网络和存储等基础设施服务，帮助公有云提供商实现资源集中管理和高可用性。
- 混合云：OpenStack 还可以为混合云提供一套完整的解决方案。混合云是指将私有云和公有云相互结合，以在云计算中获得更灵活、高效和安全的服务。混合云可以在私有云和公有云之间实现数据和应用的迁移、备份和恢复等操作。
- 大规模虚拟机管理：OpenStack 可以规划并管理大量虚拟机，从而允许企业或服务提供商按需提供计算资源。

DevStation 工具链 / 图形化 / 南向兼容性等新增特性

openEuler DevStation 是专为开发者设计的 Linux 桌面发行版，聚焦简化开发流程与生态兼容性。新版本围绕社区工具链增强、图形化交互优化以及系统兼容性与生产力升级三大方向进行了全面升级，为开发者提供更加高效、安全且适应复杂场景的开发体验。

功能描述

1. 开发者友好的社区工具链

   全栈开发工具集成：默认搭载 VSCodium（尊重开源许可的无 telemetry 版本 IDE）及主流语言开发环境（Python/Java/Go/

Rust/C/C++）。

社区工具生态：新增 oeDeploy（一键式部署工具）、epkg（扩展软件包管理器）、devkit 和本地化 AI 开发助手，实现从环境配置到代码落地的全链路支持。

oedevplugin 插件支持：专为 openEuler 社区开发者设计的 VSCodium 插件，提供 Issue/PR 可视化管理面板，支持快速拉取社区代码仓、提交 PR、自动化检视代码规范（如 License 头校验、代码格式检查），并实时同步社区任务状态。

智能助手：支持自然语言生成代码片段、一键生成 API 文档及 Linux 命令解释，内置隐私保护模式（离线运行）。

2. 图形化界面与生产力工具升级

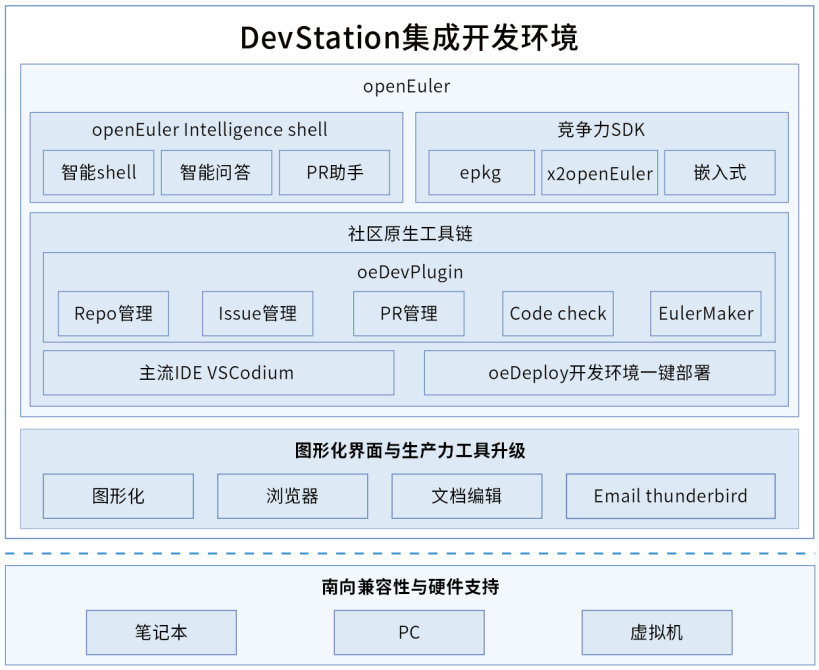
导航栏与工作区优化：新增智能导航栏，动态聚合开发工具、系统管理及常用应用的快捷入口，支持自定义工作区布局。

桌面增强工具：预装 Thunderbird 邮件客户端，提升办公效率。

3. 南向兼容性与硬件支持增强

笔记本友好适配：全面支持主流笔记本硬件（触摸板、Wi-Fi 6 芯片组及蓝牙外设管理，多架构与驱动覆盖），提升 AI 训练与渲染性能。

DevStation 树莓派镜像发布：新增 DevStation 树莓派专用镜像，默认集成轻量化桌面环境及 IoT 开发工具链（如 VScodium、oedevplugin），针对 ARM 架构优化 Python 科学计算库（NumPy/Pandas）性能。



应用场景

社区开发者与开源协作：初次接触 openEuler 的开发者使用 oedevplugin 在 VSCodium 中直接克隆社区仓库，通过引导式 PR 模板补全测试用例。

维护者高效批量审核：社区 Maintainer 利用插件内置的“冲突检测引擎”预览多个 PR 的冲突风险，结合 openEuler 大模型智能交互平台生成代码变更摘要，快速评估贡献质量，降低人工审查成本。

分布式团队协作：跨时区团队通过 epkg 共享自定义软件源，确保依赖版本一致性；配合 oedeploy 将开发环境一键部署至分布式多节点。

CI/CD 流水线：开发者在本地完成本地架构调试后，通过将编译产物直连 openEuler 社区 流水线，触发自动化测试。

oeDeploy 部署能力增强

oeDeploy 是一款轻量级的软件部署工具，旨在帮助开发者快速、高效地完成各类软件环境部署，对单节点与分布式场景均可适配。

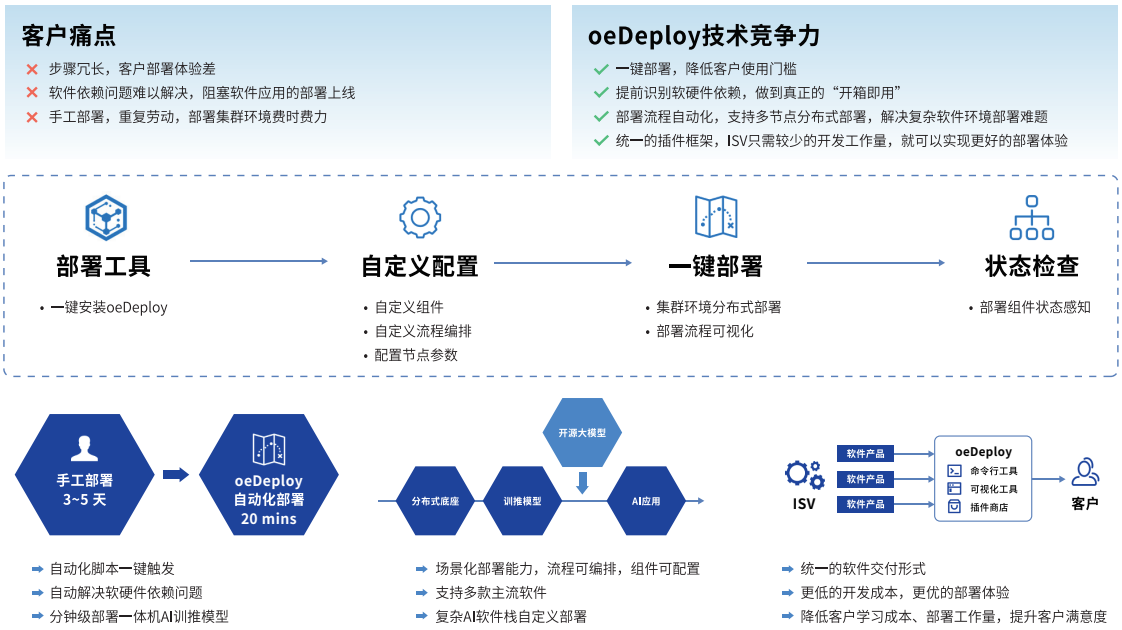
功能描述

多场景支持：支持单节点与集群环境的软件应用一键部署，用自动化脚本代替手动操作，大幅降低部署时长。

主流软件一键部署：支持主流软件的典型部署方案，开发团队将持续引入其它主流插件，不断拓展 oeDeploy 的部署能力范围。

灵活的插件化管理：oeDeploy 提供可扩展的插件架构，开发者可基于实际需求，快速开发自定义部署插件，匹配自身业务特性。

高效开发体验：oeDeploy 已支持命令行工具，并即将上线可视化工具与插件商店；开发者仅需关注步骤编排与核心功能开发，用更少的代码，实现更高效的软件部署。



应用场景

ISV 与开发团队可以将 oeDeploy 作为向客户交付软件产品的统一形式。借助 oeDeploy 提供的命令行工具与插件框架，开发团队只需较少的开发工作量，就可以实现优秀的部署效果，降低客户的额外学习成本，提高客户满意度。当前，oeDeploy 已经支持了 EulerMaker、Mindspore-DeepSeek 以及部分 AI 智能体应用的快速部署。

面向开发者与学生群体，oeDeploy 提供了复杂软件环境的快速部署能力，可以在几分钟内部署 Kuberay、TensorFlow、Pytorch 等 AI 训推框架，大幅降低学习门槛，避免重复操作。同时，开发者也可以基于 oeDeploy 发布自定义的部署能力，让更多用户受益于一键部署的便利。

08 著作权说明

09 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。

附录 10

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/docs/zh/vulnerability-management-process/security-process.md
安全披露信息	https://gitee.com/openeuler/security-committee/blob/master/docs/zh/vulnerability-management-process/security-disclosure.md