openEuler 24.03 LTS SP1 技术白皮书

1. 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。

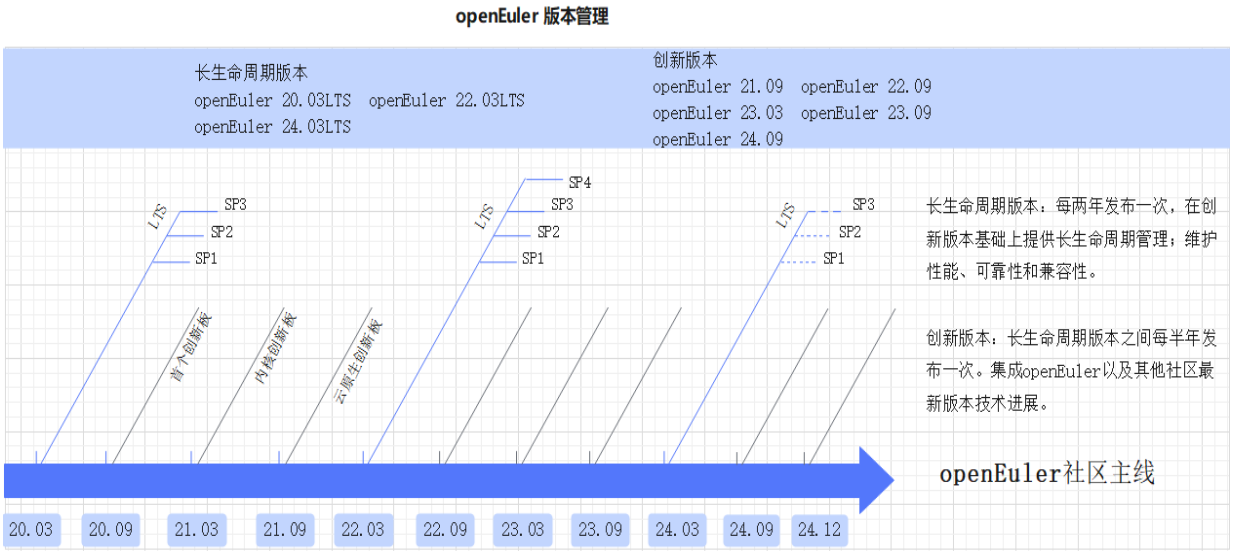
2023 年 12 月 30 日，发布 openEuler 22.03 LTS SP3 版本，是 22.03 LTS 版本增强扩展版本，面向服务器、云原生、边缘计算和嵌入式场景，持续提供更多新特性和功能扩展，给开发者和用户带来全新的体验，服务更多的领域和更多的用户。

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。

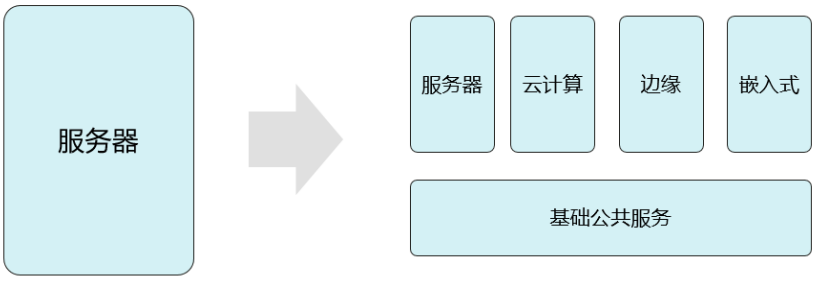


openEuler 作为一个操作系统发行版平台，每两年推出一个 LTS 版本。该版本为企业级用户提供一个安全稳定可靠的操作系统。

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。

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。

openEuler 24.03 LTS SP1 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 24.03 LTS SP1 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 原生 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。

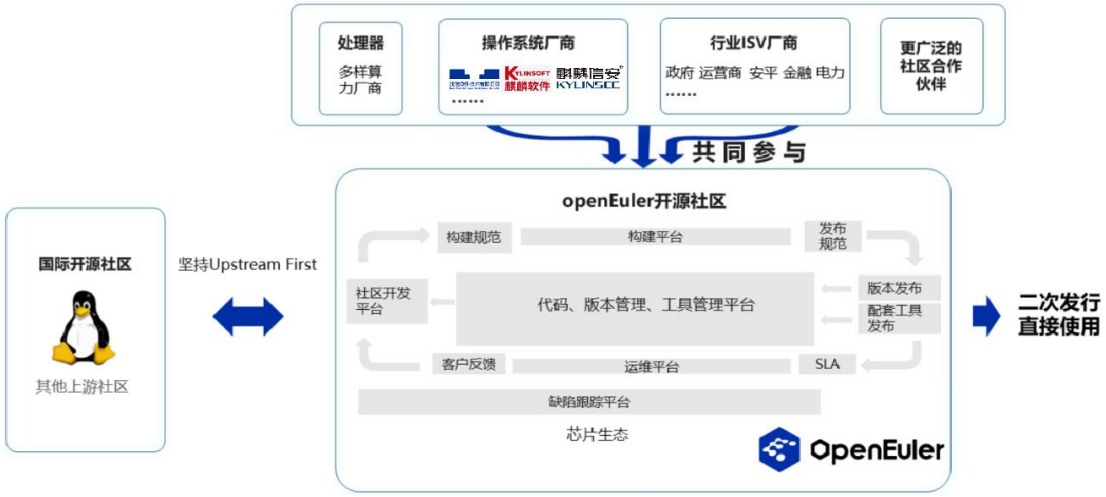
繁荣社区生态

• 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/。

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、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 24.03 LTS SP1 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：**

HostOS	GuestOS(虚拟机)	架构
openEuler 24.03 LTS SP1	Centos 6	x86_64
openEuler 24.03 LTS SP1	Centos 7	aarch64
openEuler 24.03 LTS SP1	Centos 7	x86_64
openEuler 24.03 LTS SP1	Centos 8	aarch64
openEuler 24.03 LTS SP1	Centos 8	x86_64
openEuler 24.03 LTS SP1	Windows Server 2016	aarch64
openEuler 24.03 LTS SP1	Windows Server 2016	x86_64
openEuler 24.03 LTS SP1	Windows Server 2019	aarch64
openEuler 24.03 LTS SP1	Windows Server 2019	x86_64

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 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
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
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

4. 场景创新

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。

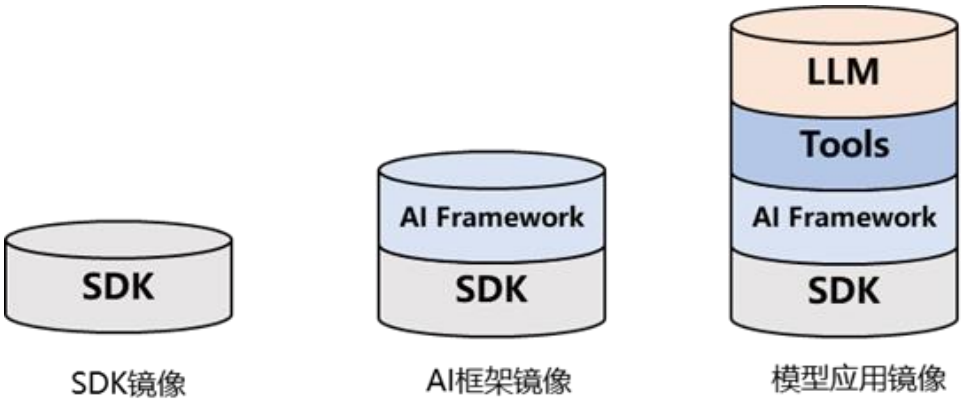
OS for AI

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。

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。

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 Copilot System，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。

智能问答

功能描述

openEuler Copilot System 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 Copilot System 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 Copilot System 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。

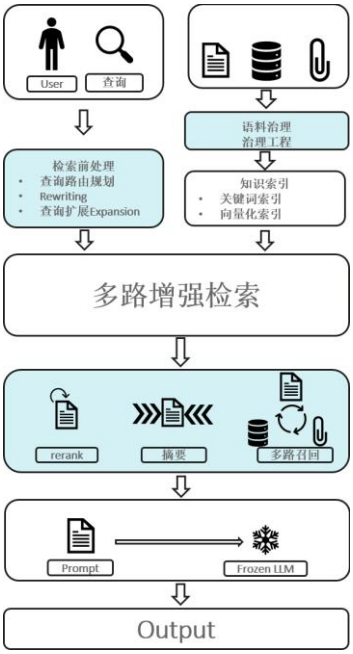
智能交互：在面对模糊或复杂的用户指令时，openEuler Copilot System 能主动询问用户，以澄清或获取更多信息。用户在补充信息后，openEuler Copilot System 能够继续执行工作流，并确保所采取的行动符合用户的期望。

任务推荐

智能响应：openEuler Copilot System 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 Copilot System 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 Copilot System 还能够学习和适应用户的习惯和偏好。

智能指引：openEuler Copilot System 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 Copilot System 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 Copilot System 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 Copilot System 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。

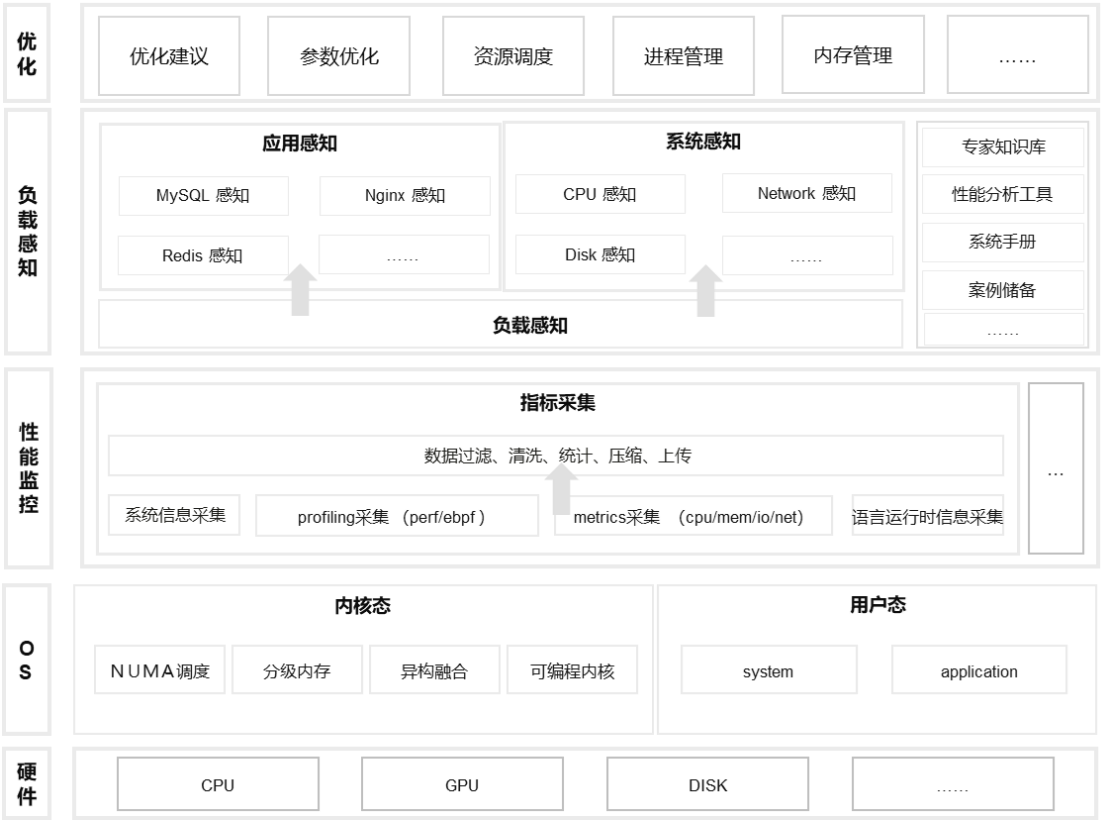
相关使用方式请参考 openEuler Copilot System 智能问答用户指南。

智能调优

功能描述

openEuler Copilot System 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 Copilot System 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。

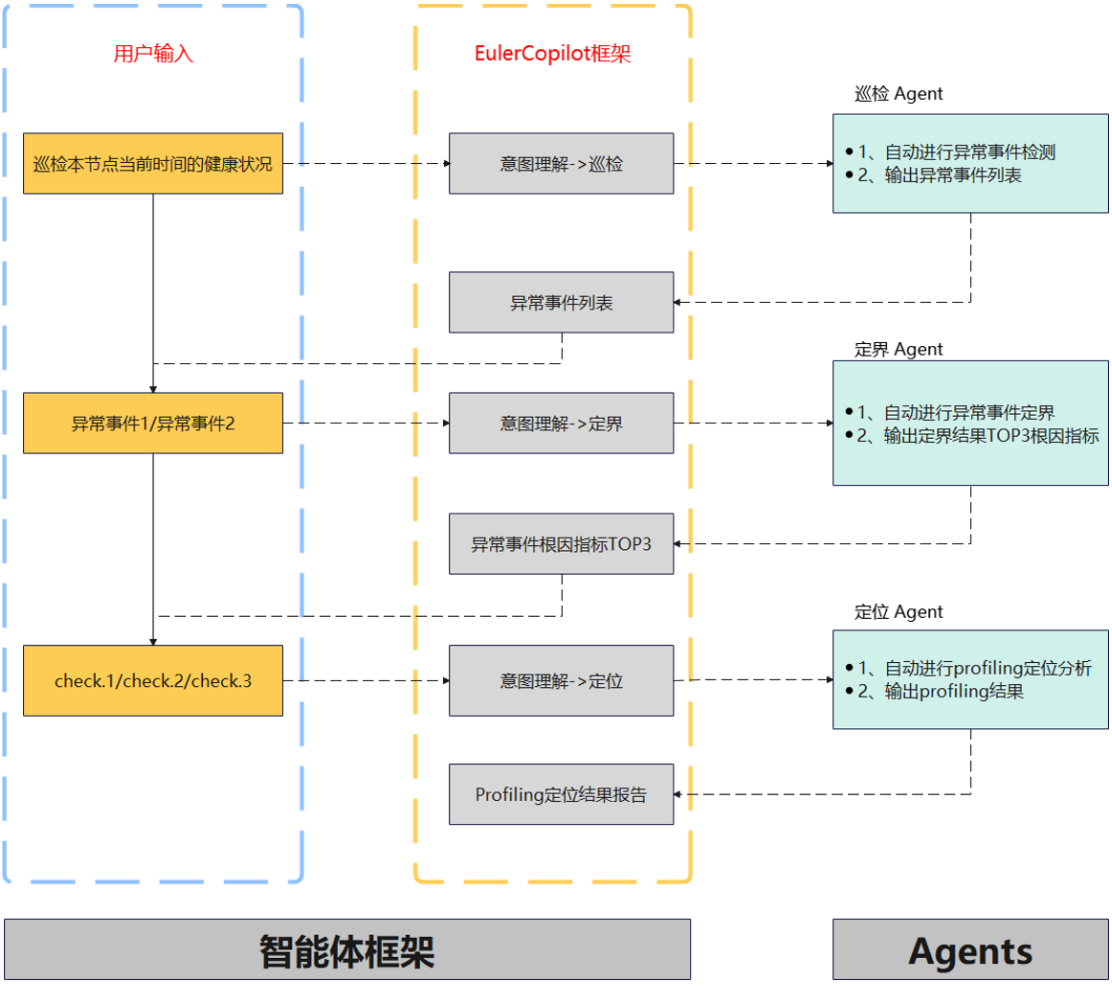


应用场景

- 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 TOP3
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

智能诊断接口在本次 openEuler 24.03 LTS SP1 版本的功能范围是：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。

智能容器镜像

功能描述

openEuler Copilot System 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
			2.13.1-oe2203sp3
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。

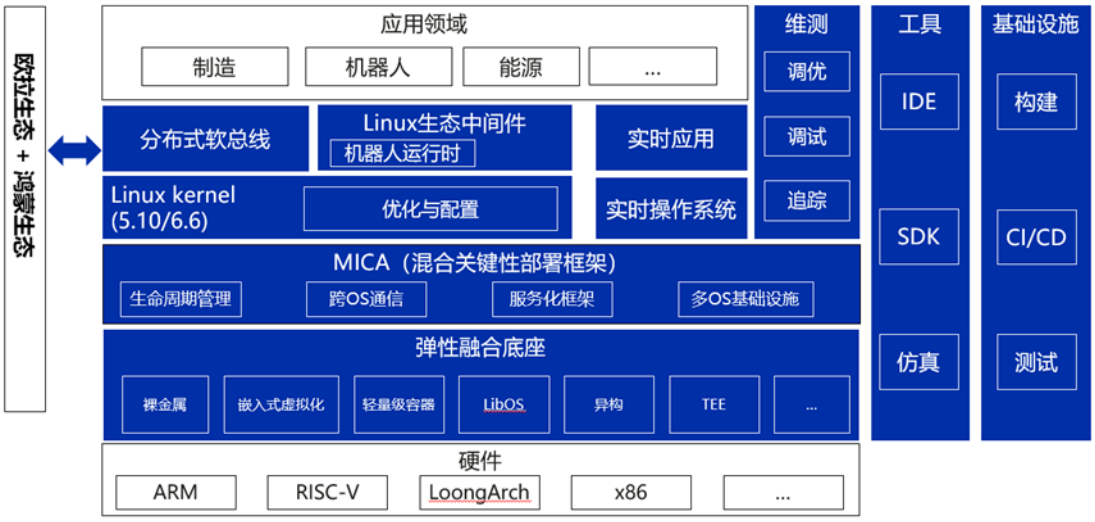
openEuler Embedded

openEuler 发布面向嵌入式领域的版本 openEuler 24.03 LTS SP1，构建了一个相对完整的综合嵌入系统软件平台，在南北向生态、关键技术特性、基础设施、落地场景等方面都有显著的进步。

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。在内核方面，结合嵌入式场景的实际需求，继续保留双内核版本支持，支持 5.10 和 6.6 双内核，用户可以结合 BSP、应用场景等实际需求在两个版本的内核中选择其一，同时开发了自定义内核的能力。嵌入式弹性底座支持多种解决方案，包括 Jailhouse 分区虚拟化方案、openAMP 裸金属混合部署方案、基于 ZVM 和 Rust-Shyper 的实时虚拟化部署方案，用户可以根据自己的使用场景选择最优的部署方案。在嵌入式弹性底座之上打造了混合关键性部署框架 MICA，对下屏蔽不同底座的差异，对上为不同运行时提供统一的接口。在北向，目前已经支持 600+软件包，包括支持 ROS humble 版本，集成 ros-core、ros-base、SLAM 等核心软件包，满足 ROS2 运行时要求，针对嵌入式上层用户开发 SDK，加入了 ROS2 的嵌入式特色能力，SDK 支持 ROS2 colcon 交叉编译；支持 BMC 生态，已初步支持 openBMC。在南向，新增飞腾、海思、瑞萨、德州仪器、全志等硬件的支持，特别是提出了面向开发者的硬件开发板概念“欧拉派/Euler Pi”，并具体推出了第一款 openEuler Embedded 原生开发板“海鸥派/HiEuler Pi”。在基础设施，正式发布 openEuler Embedded 元工具 oebuild，构建系统升级到 Yocto 4.0 LTS，工具链新增支持 LLVM 工具链，可以采用 LLVM 工具链来构建镜像，并生成对应的 SDK，可以获得在性能、体积、安全性等诸多方面的改进。在落地场景方面，openEuler Embedded 已经有了多个商业发行版/自用版，在 BMC，工业控制器，机器人控制器等领域开始应用。

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，完善工业中间件、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、瑞芯微、瑞萨、德州仪器、飞腾、赛昉、全志等厂商的芯片。openEuler Embedded 24.03 LTS SP1 新增鲲鹏 920 支持。

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。

DevStation 开发者工作站支持

Devstation 是 openEuler 首个面向开发者的镜像，预装 VSCODE，Devstation 将打通部署，编码，编译，构建，发布全流程，并集成 oeDeploy 工具，开发者可以方便的使用 oeDeploy 完成 AI 软件栈，云原生软件栈部署，使用 oeDevPlugin 插件进行一键拉取代码仓，一键使用 AI4C 编译器编译，轻松使用 Devstation 版本进行软件开发；同时 Devstation 将会集成 openEuler 新型包管理体系 EPKG，可以进行多环境，多版本安装，方便开发者在不同版本之间进行切换。

功能描述

开发者友好的集成环境：发行版预装了广泛的开发工具和 IDE，如 VS Code 系列等。支持多种编程语言，满足从前端、后端到全栈开发的需求。

图形化编程环境：集成了图形化编程工具，降低了新手的编程门槛，同时也为高级开发者提供了可视化编程的强大功能。

AI 开发支持：针对 AI 领域的开发者，可以使用 oeDeploy 进行 Kubeflow 安装，大幅降低使用成本

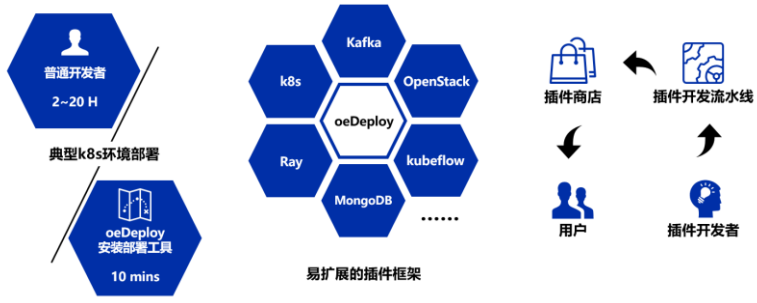
调试与测试工具：内置 GDB、CUnit、gtest、perf 等调试工具和测试、调优工具，帮助开发者快速调试和自动化测试，提升开发效率。

版本控制和协作：集成 Git、版本控制工具，并支持多种远程协作工具，如 Slack、Mattermost 和 GitLab，使得团队开发和远程协作更加顺畅。

应用场景

多语言开发环境：适用于需要同时开发多语言（如 Python、JavaScript、Java、C++）项目的开发者，无需手动配置环境，系统预装各种编译器、解释器和构建工具。

快速安装部署平台：Devstation 集成了 oeDeploy，可以对 kubeflow、k8s 等分布式软件实现分钟级部署，大幅减少开发者部署时间。oeDeploy 同时提供了统一的插件框架与原子化的部署能力，开发者只需遵循简单的开发规范，就可以快速发布自定义的安装部署插件，帮助更多用户解决安装部署问题。



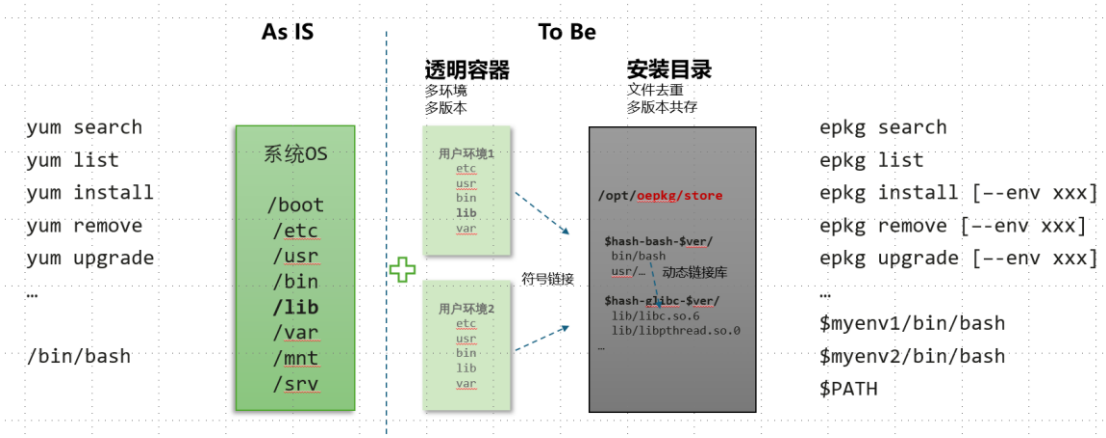
AI 开发和数据科学：Devstation 之后会为从事人工智能、机器学习、深度学习的开发者准备了完备的开发环境，预装常用数据处理库和模型训练工具，支持异构计算加速，适合进行模型训练和推理。

游戏与多媒体开发：为游戏开发者提供了图形化开发工具，同时也预装了多媒体编辑软件，支持图像、音频、视频处理的全栈开发。

学生和编程初学者：通过图形化编程工具和内置的开发教程，系统也适合计算机科学入门者使用，降低学习编程的门槛，未来 Devstation 将自带 openEuler Copilot System，提供 Linux 操作系统智能问答服务，解决初学者，开发者在操作系统上遇到的问题。

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。同时支持环境环境实现环境的创建/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：支持普通用户安装，支持安装不同版本的软件包，不同版本的同一软件包安装不冲突。使能用户在同一个节点上，快速安装同一软件包的不同版本，实现多版本软件包的共存。

环境管理：支持环境环境实现环境的创建/切换/使能等操作，用户通过环境的切换，在环境中使用不同的 channel，实现在不同的环境中使用不同版本的软件包。用户可以基于环境，快速实现软件包版本的切换。

普通用户安装：epkg 支持普通用户安装软件包，普通用户能够自行创建环境，对个人用户下的环境镜像管理，无需特权版本。降低软件包安装引起的安全问题。

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 14 多版本编译工具链支持

openEuler GCC Toolset 14 编译工具链，为用户提供了更加灵活且高效的编译环境选择。用户可以轻松地在不同版本的 GCC 之间进行切换，以便充分利用新硬件特性，同时享受到 GCC 最新优化所带来的性能提升。

功能描述

为了使能多样算例新特性，满足不同用户对不同硬件特性支持的需求，在 openEuler 24.03 LTS SP1 版本推出 openEuler GCC Toolset 14 编译工具链，该工具链提供一个高于系统主 GCC 版本的副版本 GCC 14 编译工具链，为用户提供了更加灵活且高效的编译环境选择。通过使用 openEuler GCC Toolset 14 副版本编译工具链，用户可以轻松地在不同版本的 GCC 之间进行切换，以便充分利用新硬件特性，同时享受到 GCC 最新优化所带来的性能提升。

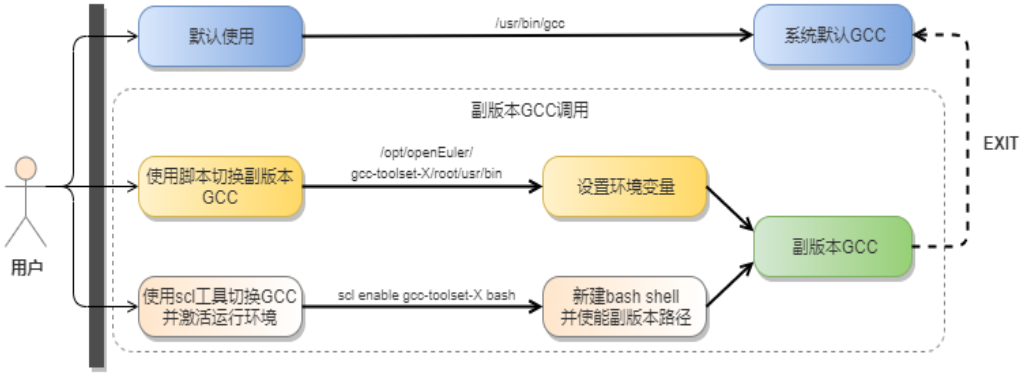
为了与系统默认主版本 GCC 解耦，防止副版本 GCC 安装与主版本 GCC 安装的依赖库产生冲突，openEuler GCC Toolset 14 工具链的软件包名均以前缀“gcc-toolset-14-”开头，后接原有 GCC 软件包名。

此外，为便于版本切换与管理，本方案引入 SCL 版本切换工具。SCL 工具的核心就是会在/opt/openEuler/gcc-toolset-14 路径下提供一个 enable 脚本，通过注册将 gcc-toolset-14 的环境变量注册到 SCL 工具中，从而可以使用 SCL 工具启动一个新的 bash shell，此 bash shell 中的环境变量即为 enable 脚本中设置的副版本环境变量，从而实现主副版本 GCC 工具链的便捷切换。

应用场景

主版本场景：正常编译使用系统默认的 gcc-12.3.1。

副版本场景：需要使用 GCC 14 高版本特性构建相关应用，使用 SCL 工具将构建环境切换为 gcc-toolset-14 编译工具链的编译环境。



GCC 主副版本切换示意图

5. 内核创新

openEuler 内核中的新特性

openEuler 24.03 LTS SP1 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。

- 内存管理 folio 特性：Linux 内存管理基于 page（页）转换到由 folio（拉丁语 foliō，对开本）进行管理，相比 page，folio 可以由一个或多个 page 组成，采用 struct folio 参数的函数声明它将对整个（1 个或者多个）页面进行操作，而不仅仅是 PAGE_SIZE 字节，从而移除不必要复合页转换，降低误用 tail page 问题；从内存管理效率上采用 folio 减少 LRU 链表数量，提升内存回收效率，另一方，一次分配更多连续内存减少 page fault 次数，一定程度降低内存碎片化；而在 IO 方面，可以加速大 IO 的读写效率，提升吞吐。全量支持匿名页、文件页的 large folio，提供系统级别的开关控制，业务可以按需使用。对于 ARM64 架构，基于硬件 contiguous bit 技术（16 个连续 PTE 只占一个 TLB entry），可以进一步降低系统 TLB miss，从而提升整体系统性能。24.03 LTS SP1 版本新增支持 anonymous shmem 分配 mTHP 与支持 mTHP 的 lazyfree，进一步增加内存

子系统对于 large folio 的支持；新增 page cache 分配 mTHP 的 sysfs 控制接口，提供系统级别的开关控制，业务可以按需使用。

- MPTCP 特性支持：MPTCP 协议诞生旨在突破传统 TCP 协议的单一路径传输瓶颈，允许应用程序使用多个网络路径进行并行数据传输。这一设计优化了网络硬件资源的利用效率，通过智能地将流量分配至不同传输路径，显著缓解了网络拥塞问题，从而提高数据传输的可靠性和吞吐量。

  目前，MPTCP 在下述网络场景中已经展现出了其优秀的性能：

  1) 网络通路的选择：在现有的网络通路中，根据延迟、带宽等指标评估，选择最优的通路。
  2) 无缝切网：在不同类型网络之间切换时，数据传输不中断。
  3) 数据分流：同时使用多个通道传输，对数据包进行分发实现并发传输，增加网络带宽。

在实验环境中，采用 MPTCP v1 技术的 RSYNC 文件传输工具展现出了令人满意的效率提升。具体而言，传输 1.3GB 大小的文件时，传输时间由原来的 114.83 s 缩短至仅 14.35s，平均传输速度由原来的 11.08 MB/s 提升至 88.25 MB/s，可以极大程度的缩减文件传输时间。同时，实验模拟了传输过程中一条或多条路径突发故障而断开的场景，MPTCP 在此种场景下可以将数据无缝切换至其他可用的数据通道，确保数据传输的连续性与完整性。

在 openEuler 24.03 LTS SP1 中，已经完成了对 linux 主线内核 6.9 中 MPTCP 相关特性的全面移植与功能优化。

- ext4 文件系统支持 Large folio：iozone 性能总分可以提升 80%，iomap 框架回写流程支持批量映射 block。支持 ext4 默认模式下批量申请 block，大幅优化各类 benchmark 下 ext4 性能表现（华为贡献）。ext4 buffer io 读写流程以及 pagecache 回写流程弃用老旧的 buffer_head 框架，切换至 iomap 框架，并通过 iomap 框架实现 ext4 支持 large folio。24.03 LTS SP1 版本新增对于 block size < folio size 场景的小 buffered IO（<=4KB）的性能优化，性能提升 20%。

- xcall/xint 特性：随着 Linux 内核的发展，系统调用成为性能瓶颈，尤其是在功能简单的调用中。AARCH64 平台上的 SYSCALL 共享异常入口，包含安全检查等冗余流程。降低 SYSCALL 开销的方法包括业务前移和批量处理，但需业务适配。XCALL 提供了一种无需业务代码感知的方案，通过优化 SYSCALL 处理，牺牲部分维测和安全功能

来降低系统底噪，降低系统调用处理开销。

内核为了使中断处理整体架构统一，将所有中断处理全部归一到内核通用中断处理框架中，同时随着内核版本演进，通用中断处理框架附加了很多与中断处理自身的功能关系不大的安全加固和维测特性，这导致中断处理的时延不确定性增大。xint 通过提供一套精简的中断处理流程，来降低中断处理的时延和系统底噪。

- 按需加载支持 failover 特性 ： cachefiles 在按需模式下，如果守护进程崩溃或被关闭，按需加载相关的读取和挂载将返回 -EIO。所有挂载点必须要在重新拉起 daemon 后重新挂载后方可继续使用。这在公共云服务生产环境中发生时是无法接受的，这样的 I/O 错误将传播给云服务用户，可能会影响他们作业的执行，并危及系统的整体稳定性。cachefiles failover 特性避免了守护进程崩溃后重新挂载所有挂载点，只需快速重新拉起守护进程即可，用户和服务是不感知守护进程崩溃的。

- 可编程调度特性：支持可编程调度功能，为用户态提供可编程的调度接口，是 CFS 算法的扩展。用户可以根据实际的业务场景定制化调度策略，bypass 原有的 CFS 调度算法。提供的功能包括支持标签机制、支持任务选核可编程、支持负载均衡可编程、支持任务选择可编程、支持任务抢占可编程以及 kfunc 接口函数。

- 支持 SMC-D with loopback-ism 特性： SMC-D (Shared Memory Communication over DMA) 是一种兼容 socket 接口，基于共享内存，透明加速 TCP 通信的内核网络协议栈。SMC-D 早期只能用于 IBM z S390 架构机器，SMC-D with loopback-ism 技术通过创建虚拟设备 loopback-ism 模拟 ISM 功能，使得 SMC-D 可用于非 S390 架构机器，成为内核通用机制。SMC-D with loopback-ism 适用于采用 TCP 协议进行 OS 内进程间或容器间通信的场景，通过旁路内核 TCP/IP 协议栈等方法，实现通信加速。结合使用 smc-tools 工具，可以通过 LD_PRELOAD 预加载动态库的方法实现 TCP 协议栈透明替换，无需更改原有应用程序。根据社区反馈结果，与原生 TCP 相比，SMC-D with loopback-ism 能够提升网络吞吐量提升 40%以上。

- IMA RoT 特性：当前，Linux IMA（Integrity Measurement Architecture）子系统主要使用 TPM 芯片作为可信根（Root of Trust，RoT）设备，针对度量列表提供完整性证明，其在编码上也与 TPM 的操作紧耦合。而机密计算等新场景要求 IMA 可使用新型 RoT 设备，例如 openEuler 已支持的 VirtCCA。本特性为一套 IMA RoT 设备框架，在 IMA 子系统和 RoT 设备之间实现一个抽象层，既简化各类 RoT 设备对 IMA 子系统的适配，也方便用户和 IMA 子系统对各类 RoT 设备实施配置和操作。

- 支持脚本类病毒程序防护：目前勒索病毒主要是脚本类文件（如 JSP 文件），而当前内核防御非法入侵的 IMA 完整性保护技术，主要针对 ELF 类病毒文件。脚本类病毒文件通过解释器运行，能够绕开内核中的安全技术实施攻击。为了支持内核中的 IMA 完整性保护技术可以检查系统中间接执行的脚本类文件，通过系统调用 execveat() 新增执行检查的 flags，查验其执行权限，并在检查中调用 IMA 完整性度量接口以实现对脚本类文件的完整性保护。经测试验证，目前已经支持脚本解释器主动调用 execveat() 系统调用函数并传入 AT_CHECK 参数对脚本文件进行可执行权限检查（包括 IMA 检查），只有当权限检查成功后，才可继续运行脚本文件。
- haltpoll 特性：haltpoll 特性通过虚拟机 guest vcpu 在空闲时进行轮询的机制，避免 vcpu 唤醒时发送 IPI 中断，降低了中断发送和处理的开销，并且由于轮询时虚拟机不需要陷出，减少了陷入陷出的开销，该特性能够降低进程间通信时延，显著提升上下文切换效率，提升虚机性能。

内核TCP/IP协议栈支持CAQM拥塞控制算法

内核 TCP/IP 协议栈支持 CAQM 拥塞控制算法：CAQM 是一种主动队列管理算法，一种网络拥塞控制机制，主要运行于数据中心使用 TCP 传输数据的计算端侧节点和传输路径上的网侧交换机节点。

通过网侧交换节点主动计算网络空闲带宽和最优带宽分配，端侧协议栈与网侧交换机协同工作，在高并发场景下获得网络交换机“零队列“”拥塞控制效果和极低传输时延。

CAQM 算法通过在以太链路层增加拥塞控制标记字段，实现动态调整队列长度，减少延迟和丢包，提高网络资源的利用率。对于数据中心低延时通算场景，可极大减少延迟和丢包的发生，增强用户体验。

在数据中心典型场景下，对比经典 Cubic 算法，关键指标提升：1）传输时延：CAQM vs Cubic 时延降低 92.4%。2）带宽利用率：在交换机队列缓存占用降低 90%的情况下，仍保持 TCP 传输带宽利用率逼近 100%（99.97%）。

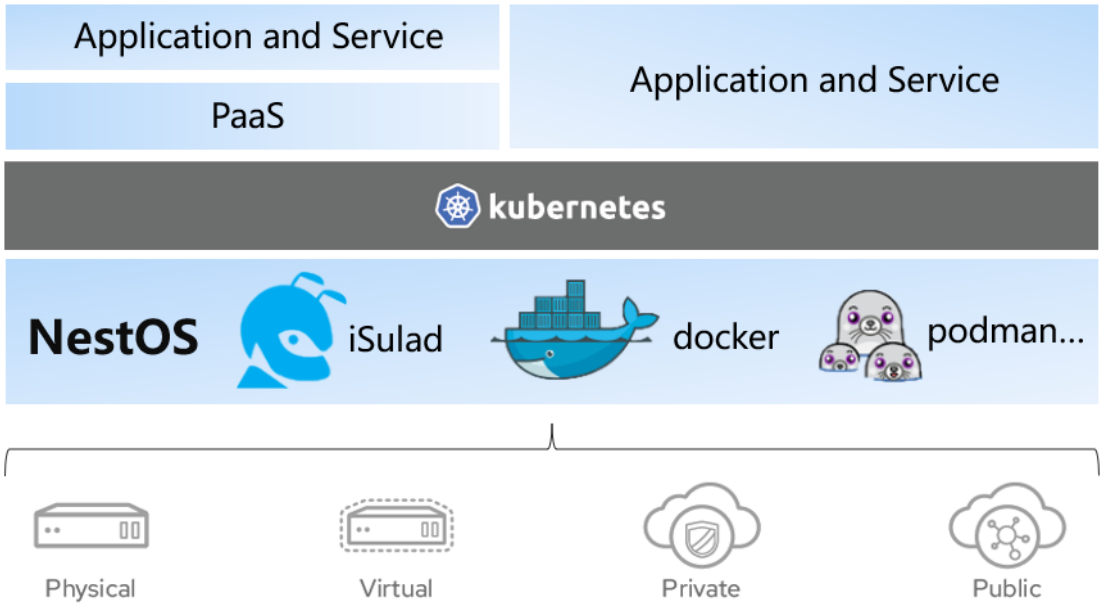
CAQM 算法使用说明：1）本算法需要端侧服务器和网侧交换机协同配合，故中间节点交换机，需要支持 CAQM 协议（协议头识别，拥塞控制字段调整等）；2）本算法通过内核编译宏（CONFIG_ETH_CAQM）控制，默认不使能，用户需通过打开编译宏，重新编译替换内核后，使能算法功能。

6. 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。

7. 特性增强

syscare 特性增强

SysCare 是一个系统级热修复软件，为操作系统提供安全补丁和系统错误热修复能力，主机无需重新启动即可修复该系统问题。 SysCare 将内核态热补丁技术与用户态热补丁技术进行融合统一，用户仅需聚焦在自己核心业务中，系统修复问题交予 SysCare 进行处理。后期计划根据修复组件的不同，提供系统热升级技术，进一步解放运维用户提升运维效率。

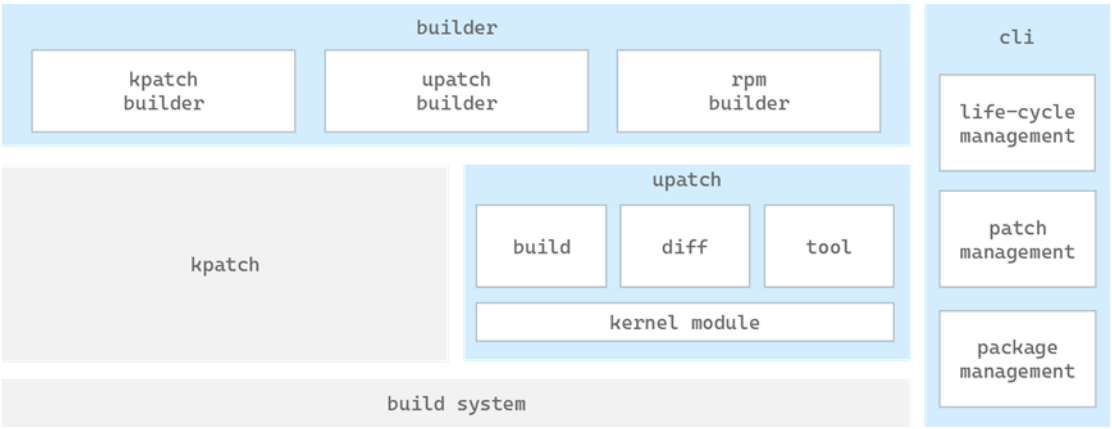
功能描述

1. 热补丁制作

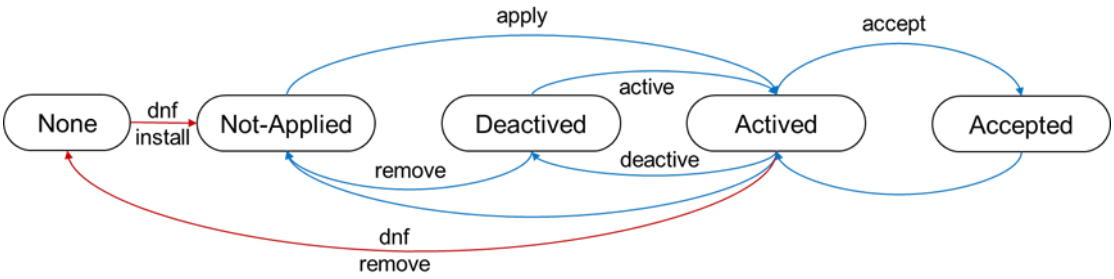
用户仅需输入目标软件的源码 RPM 包、调试信息 RPM 包与待打补丁的路径，无需对软件源码进行任何修改，即可生成对应的热补丁 RPM 包。

2. 热补丁生命周期管理

SysCare 提供一套完整的，傻瓜式补丁生命周期管理方式，旨在减少用户学习、使用成本，通过单条命令即可对热补丁进行管理。依托于 RPM 系统，SysCare 构建出的热补丁依赖关系完整，热补丁分发、安装、更新与卸载流程均无需进行特殊处理，可直接集成放入软件仓 repo。



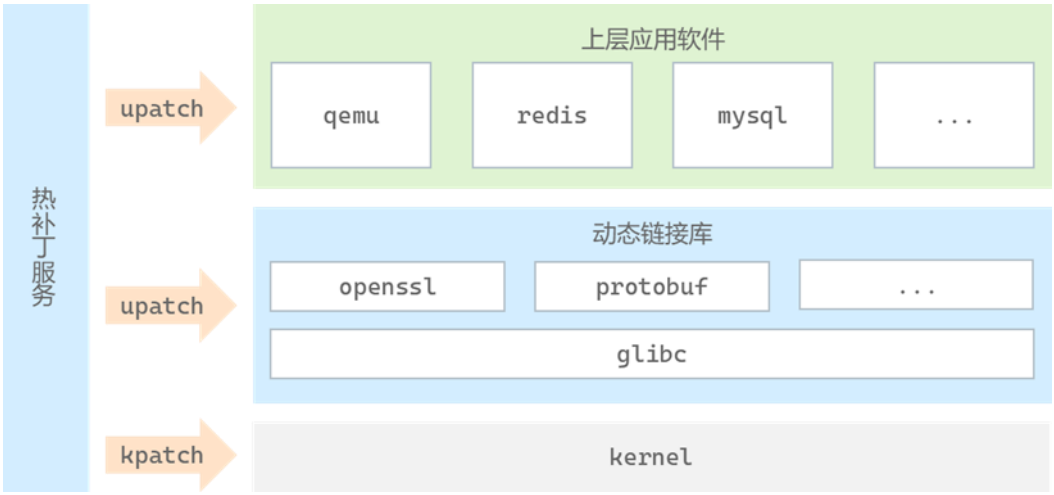
SysCare 整体架构



热补丁生命周期

3. 内核热补丁与用户态热补丁融合

SysCare 基于 upatch 和 kpatch 技术，覆盖应用、动态库、内核，自顶向下打通热补丁软件栈，提供用户无感知的全栈热修复能力。



热补丁应用范围

新增特性

- 支持重启后按照用户操作顺序恢复 ACCEPTED 状态热补丁。

约束限制

- 仅支持 64 位系统；
- 仅支持 ELF 格式的热修复，不支持解释型语言，不支持纯汇编修改；
- 仅支持 GCC / G++编译器，且不支持交叉编译；
- 暂不支持修改 TLS 变量；
- 暂不支持 LTO 优化；
- 仅 upatch-build 实现编译器配置接口，syscare-build 暂未提供。

应用场景

应用场景 1：CVE 补丁快速修复。

应用场景 2：现网问题临时定位。

iSula 支持 NRI 插件式扩展

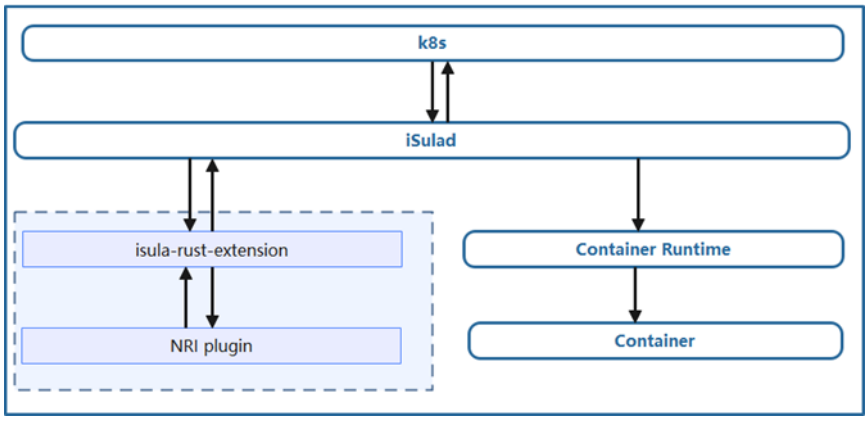
NRI (Node Resource Interface)，是用于控制节点资源的公共接口，是 CRI 兼容的容器运行时插件扩展的通用框架。它为扩展插件提供了跟踪容器状态，并对其配置进行有限修改的基本机制。允许将用户某些自定的逻辑插入到 OCI 兼容的运行时中，此逻辑可以对容器进行受控更改，或在容器生命周期的某些时间点执行 OCI 范围之外的额外操作。iSulad 新增对 NRI 插件式扩展的支持，减少 k8s 场景下对于容器资源管理维护成本，消除调度延迟，规范信息的一致性。

功能描述

NRI 插件通过请求 isula-rust-extension 组件中启动的 NRI runtime Service 服务与 iSulad 建立连接后，可订阅 Pod 与 Container 的生命周期事件：

1. 可订阅 Pod 生命周期事件，包括：creation、stopping 和 removal。
2. 可订阅 Container 生命周期事件，包括 creation、post-creation、starting、post-start、updating、post-update、stopping 和 removal。

iSulad 在接收到 k8s 下发的 CRI 请求后，对于所有订阅了对应生命周期事件的 NRI 插件都发送的请求，NRI 插件可在请求中获得 Pod 与 Container 的元数据与资源信息。之后 NRI 插件可根据需求更新 Pod 与 Container 的资源配置，并将更新的信息传递给 iSulad，iSulad 将更新后的配置传递给容器运行时，生效配置。



约束限制

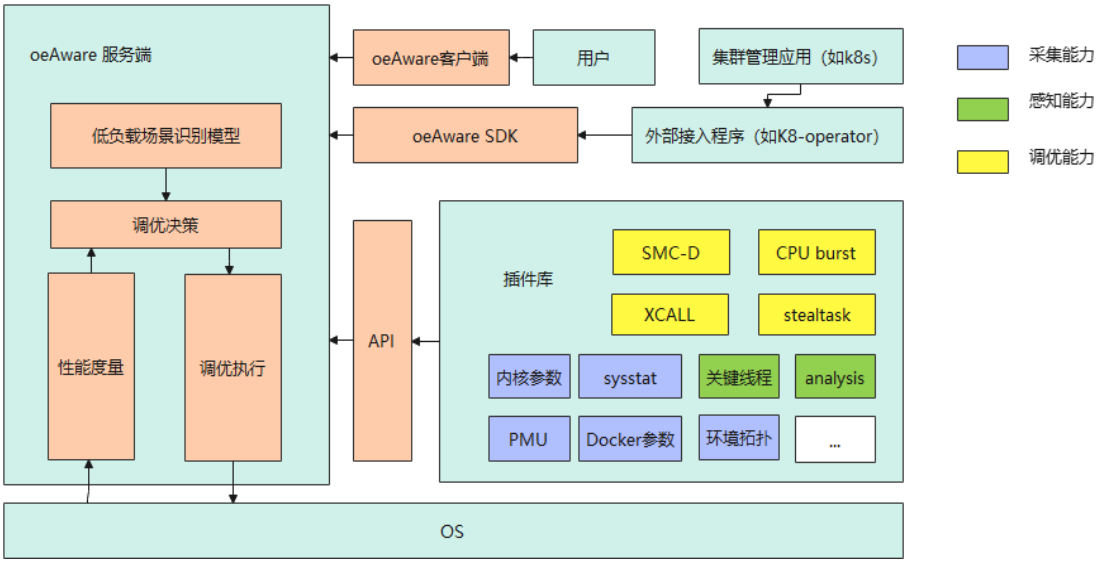
- 仅支持在 k8s CRI V1 中使用 NRI 特性
- 支持的 NRI API 版本为 0.6.1

应用场景

iSulad 的 NRI 特性可用于 k8s 场景下资源的管理。用户可通过实现 NRI 插件，订阅容器生命周期事件跟踪容器资源状态，并根据资源管理逻辑对 NRI API 范围内允许的容器配置进行修改。例如，可以利用 NRI 实现以标准化插件的形式解决在线与离线业务混部场景中资源利用率与优先级调度的问题。

oeAware 采集、调优插件等功能增强

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。

- 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 NUMA 和网络信息，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
- 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。

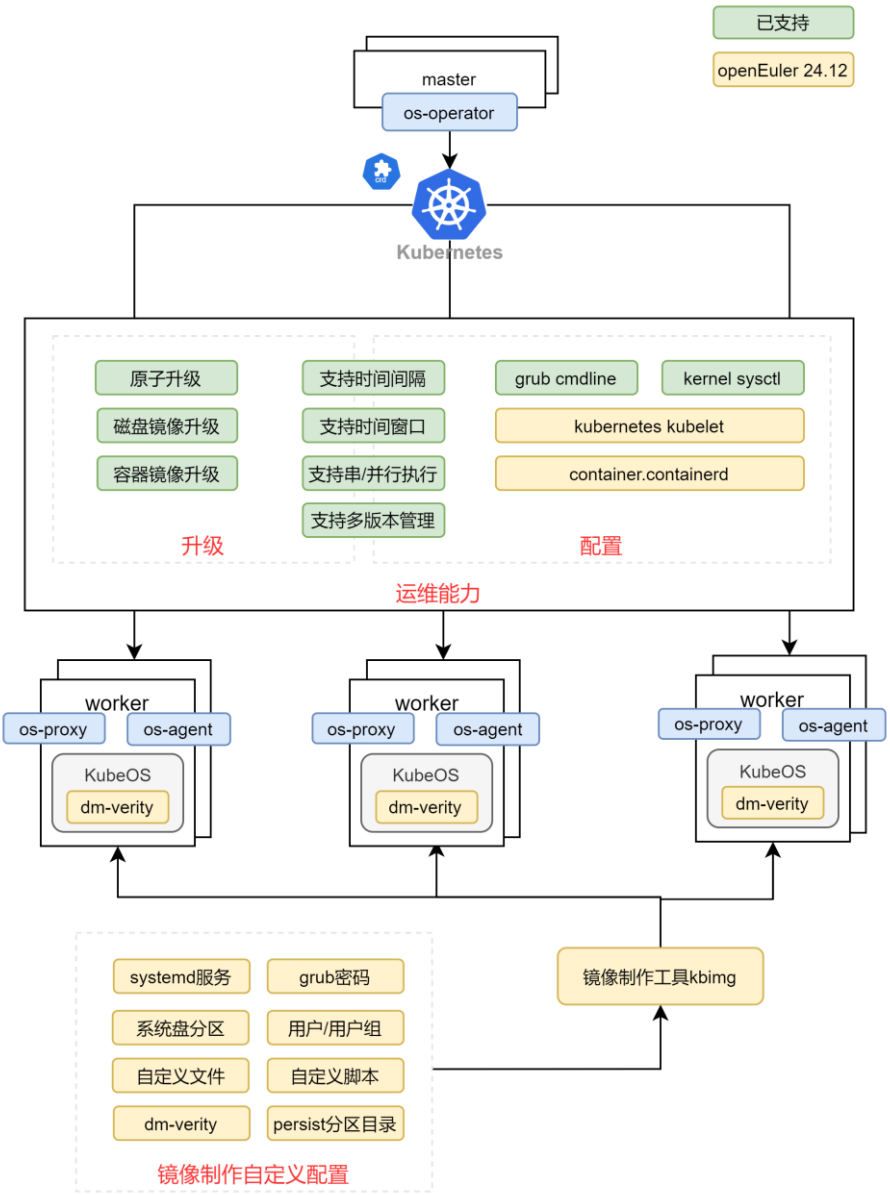
SMC-D 特别适用于需要高吞吐量和低延迟的应用场景，如高性能计算（HPC）、大数据处理和云计算平台。通过直接内存访问（DMA），SMC-D 能够显著减少 CPU 负载并提升交互式工作负载的速率。

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。

KubeOS 特性增强

KubeOS 是针对云原生场景而设计、轻量安全的云原生操作系统及运维工具，提供基于 kubernetes 的云原生操作系统统一运维能力。KubeOS 设计了专为容器运行的云原生操作系统，通过根目录只读，仅包含容器运行所需组件，dm-verity 安全加固，减少漏洞和攻击面，提升资源利用率和启动速度，提供云原化的、轻量安全的操作系统。KubeOS 支持使用 kubernetes 原生声明式 API，统一对集群 worker 节点 OS 的进行升级、配置和运维，从而降低云原生场景的运维难度、解决用户集群节点 OS 版本分裂，缺乏统一的 OS 运维管理方案的问题。

功能描述



KubeOS新增配置能力、定制化镜像制作能力和rootfs完整性保护dm-verity如图所示，具体能力如下：

1） KubeOS支持集群参数统一配置，支持通过KubeOS统一settings配置，支持如下配置：

a） KubeOS支持limits.conf文件参数统一配置；

b） KubeOS支持containerd、kubelet等集群参数统一配置。

2） KubeOS支持镜像定制化，镜像制作时支持systemd服务、grub密码、系统盘分

区、用户/用户组、文件、脚本和persist分区目录的自定义配置。

3） KubeOS支持静态完整性保护dm-verity，支持在虚拟机镜像制作时开启dm-verity，对rootfs进行完整行校验，并支持dm-verity开启时的升级和配置。

应用场景

KubeOS 支持节点 containerd、kubelet 等集群组件参数统一配置能力，解决集群配置操作复杂，节点配置不统一的问题，减少人工配置操作(天级->小时级)，降低集群维护难度。提供支持定制化功能的镜像制作工具协助用户打造定制化的云原生 OS。增加启动完整性校验，确保基础设施可信不可变，系统每次启动均符合预期，避免特权攻击篡改篡改系统设置（rootkit 攻击），构建安全的云原生操作系统。

IMA 特性增强

内核完整性度量架构（IMA, Integrity Measurement Architecture）是 Linux 开源，且在业界被广泛使用的文件完整性保护技术。在实际应用场景中，IMA 可以用来对系统运行的程序发起完整性检查，一方面检测应用程序篡改，另一方面通过白名单机制以保证只有经过认证（如签名或 HMAC）的文件才可以被运行。

目前运行在 Linux 操作系统上的应用程序可分为两种:

- 二进制可执行程序：即符合 ELF 格式标准的程序文件，可直接通过 exec/mmap 系统调用运行;
- 解释器类应用程序：即通过解释器间接运行的程序文件，如通过 Bash/Python/Lua 等解释器加载运行的脚本程序，或 JVM 运行的 Java 程序等。

对于二进制可执行程序，IMA 可以通过在 exec/mmap 系统调用的 hook 函数发起度量或校验流程，从而实现完整性保护。

但是针对解释器类应用程序，现有的 IMA 机制无法进行有效保护。其原因是此类应用程序主要通过 read 系统调用由解释器加载并解析运行，而 IMA 无法将其和其他可变的文件（如配置文件、临时文件等）进行区分，一旦针对 read 系统调用配置开启 IMA 机制，则会将其他可变文件也纳入保护范围，而可变文件无法预先生成度量基线或校验凭据，从而导致完整性检查失败。

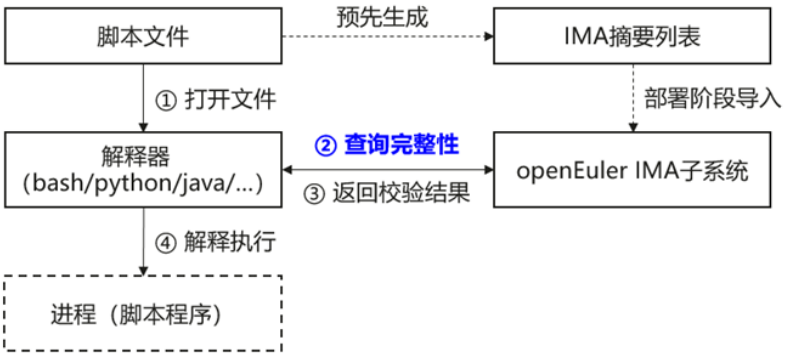
因此本特性旨在对现有 IMA 机制进行增强，有效提升对解释器类应用程序的完整性保护能力。

功能描述

本需求的核心功能是，对OS现有权限机制进行扩展，保证通过exec运行的程序（如./test.sh）和通过解释器间接运行的程序（bash ./test.sh）具备相同的权限检查流程。具体方案如下：

- openEuler 24.03 LTS SP1版本针对execveat系统调用新增AT_CHECK参数的支持，实现对文件进行可执行权限检查并返回结果，而不真正地执行该文件；
- 针对bash、jdk等组件进行功能扩展，调用execveat+CHECK对待运行的脚本文件或java文件进行可执行权限检查，只有当检查成功后，才可继续运行程序。

该特性与IMA机制相结合，即可实现如下图所示的解释器类应用程序完整性保护方案。由解释器调用execveat触发对脚本程序的可执行权限检查，并间接触发IMA的exec检查策略，在有效实现脚本程序完整性保护的同时，避免开启IMA read检查策略导致的保护范围扩大化风险。



同时其他社区开发人员或用户可基于该特性，自行扩展其他解释器或类似机制的支持。

应用场景

在高安全要求的场景，用户可基于该特性实现“应用程序白名单”功能，即只允许通过 IMA 度量或校验的二进制程序或脚本程序才可在系统中运行。

异构可信根

典型的攻击手段往往伴随着信息系统真实性、完整性的破坏，目前业界的共识是通过硬件可信根对系统关键组件进行度量/验证，一旦检测到篡改或仿冒行为，就执行告警或拦截。

当前业界主流是采用 TPM 作为信任根，结合完整性度量软件栈逐级构筑系统信任链，从而保证系统各组件的真实性和完整性。openEuler 当前支持的完整性度量特性包括：度量启动、IMA 文件度量、DIM 内存度量等。

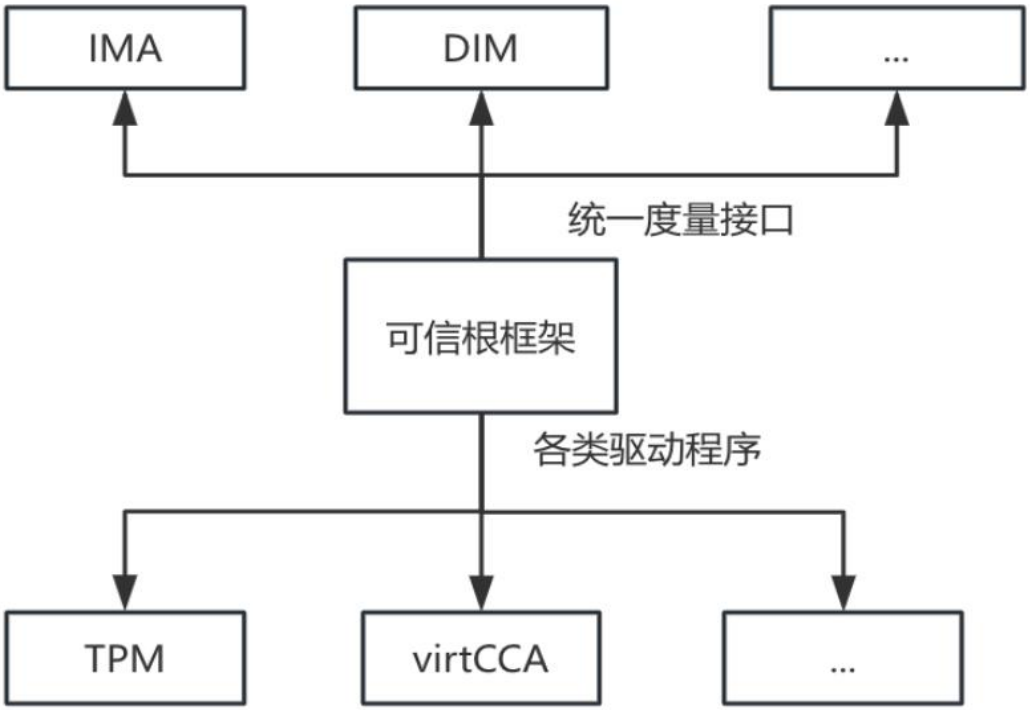
近年来，随着可信计算、机密计算等安全技术的发展，业界出现了各种形态不一的硬件可信根，及其配套的证明体系，例如：

- TCM (Trusted Cryptography Module)
- TPCM (Trusted Platform Control Module)
- TDX (Intel Trust Domain Extensions)
- CCA (Arm Confidential Compute Architecture)
- virtCCA (Virtualized Arm Confidential Compute Architecture)

因此，为支撑 openEuler 在不同硬件可信根下使能完整性度量软件栈，亟需使能一套异构可信根实例对接机制。

功能描述

openEuler 24.03 LTS SP1 版本在内核的 integrity 子模块中实现了一套可信根框架，南向支持多种可信根驱动，北向提供统一度量接口，对接上层完整性保护软件栈，将完整性度量特性的硬件可信根支持范围从单 TPM 扩展为多元异构可信根。



应用场景

当前该框架已实现对于TPM和virtCCA可信根的支持，用户可在支持TPM和virtCCA的环境中使能IMA特性，并通过可信根证明IMA度量结果的完整性。同时其他社区开发人员或硬件厂商可基于该特性，自行扩展完整性度量特性和硬件的支持。

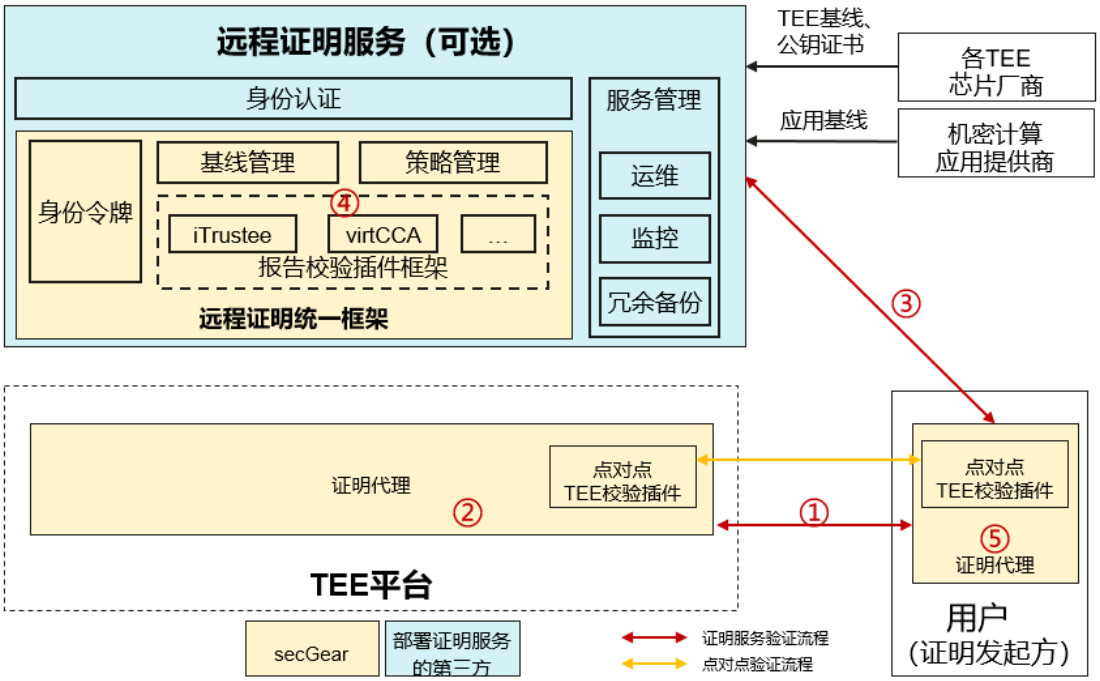
secGear 特性增强

secGear 远程证明统一框架是机密计算远程证明相关的关键组件，屏蔽不同 TEE 远程证明差异，提供 Attestation Agent 和 Attestation Service 两个组件，Agent 供用户集成获取证明报告，对接证明服务；Service 可独立部署，支持 iTrustee、virtCCA 远程证明报告的验证。

功能描述

远程证明统一框架聚焦机密计算相关功能，部署服务时需要的服务运维等相关能力由服务部署第三方提供。远程证明统一框架的关键技术如下:

- **报告校验插件框架**：支持运行时兼容 iTrustee、vritCCA、CCA 等不同 TEE 平台证明报告检验，支持扩展新的 TEE 报告检验插件。
- **证书基线管理**：支持对不同 TEE 类型的 TCB/TA 基线值管理及公钥证书管理，集中部署到服务端，对用户透明。
- **策略管理**：提供默认策略（易用）、用户定制策略（灵活）。
- **身份令牌**：支持对不同 TEE 签发身份令牌，由第三方信任背书，实现不同 TEE 类型相互认证。
- **证明代理**：支持对接证明服务/点对点互证，兼容 TEE 报告获取，身份令牌验证等，易集成，使用户聚焦业务。



根据使用场景，支持点对点验证和证明服务验证两种模式。

证明服务验证流程如下：

① 用户（普通节点或 TEE）对 TEE 平台发起挑战。

② TEE 平台通过证明代理获取 TEE 证明报告，并返回给用户。

③ 用户端证明代理将报告转发到远程证明服务。

④ 远程证明服务完成报告校验，返回由第三方信任背书的统一格式身份令牌。

⑤ 证明代理验证身份令牌，并解析得到证明报告校验结果。

点对点验证流程（无证明服务）如下：

① 用户向 TEE 平台发起挑战，TEE 平台返回证明报告给用户。

② 用户使用本地点对点 TEE 校验插件完成报告验证。

注意：点对点验证和远程证明服务验证时的证明代理不同，在编译时可通过编译选项，决定编译有证明服务和点对点模式的证明代理。

应用场景

在金融、AI 等场景下，基于机密计算保护运行中的隐私数据安全时，远程证明是校验机密计算环境及应用合法性的技术手段，远程证明统一框架提供了易集成、易部署的组件，帮助用户快速使能机密计算远程证明能力。

容器干扰检测，分钟级完成业务干扰源（CPU/IO）识别与干扰源发现

gala-anteater 是一款基于 AI 的操作系统灰度故障的异常检测平台，集成了多种异常检测算法，通过自动化模型预训练、线上模型的增量学习和模型更新，可以实现系统级故障发现和故障点上报。

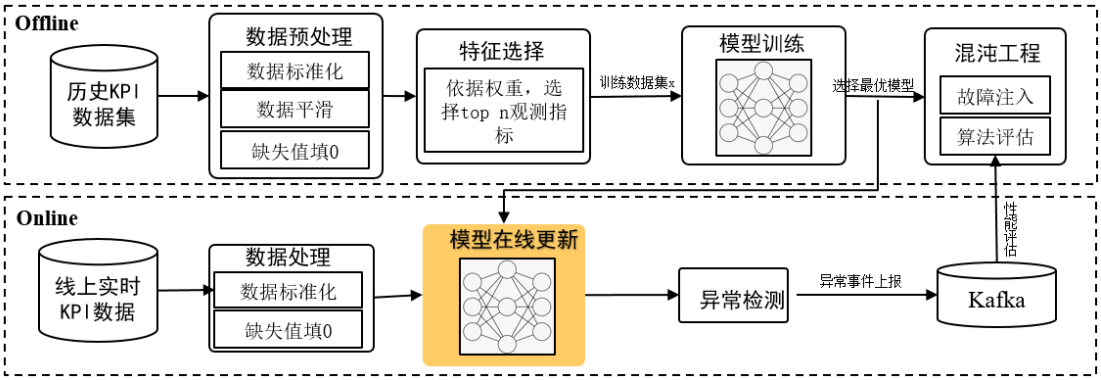
在线容器高密部署场景下，存在资源无序竞争现象，导致容器实例间相互干扰，使用 gala-anteater 可以分钟级完成业务干扰源（CPU/IO）识别与干扰源发现，辅助运维人员快速跟踪并解决问题，保障业务 Qos。

功能描述

gala-anteater 通过线上线下相结合，利用在线学习技术，实现模型的线下学习，线上更新，并应用于线上异常检测。

Offline: 首先，利用线下历史 KPI 数据集，经过数据预处理、特征选择，得到训练集；然后，利用得到的训练集，对无监督神经网络模型（例如 Variational Autoencoder）进行训练调优。最后，利用人工标注的测试集，选择最优模型。

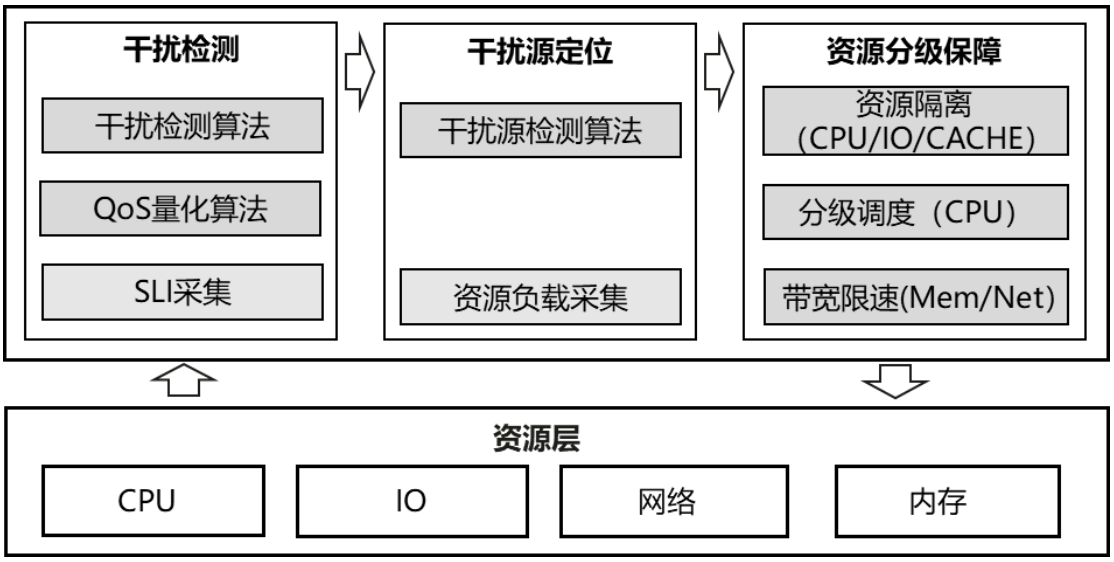
Online: 将线下训练好的模型，部署到线上，然后利用线上真实的数据集，对模型进行在线训练以及参数调优，然后利用训练好的模型，进行线上环境的实时异常检测。



应用场景

gala-anteater 支持应用级和系统级的异常检测和上报。

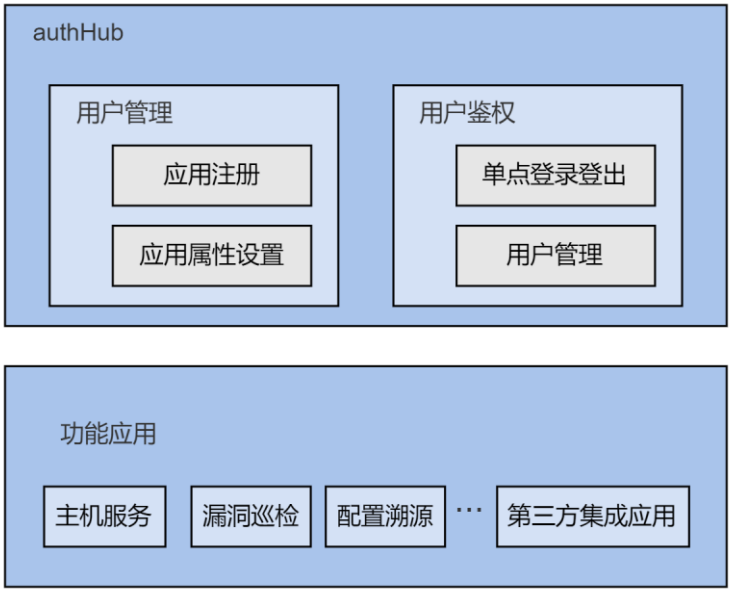
在实际应用场景中可以结合 gala-gopher 和 rubik 来实现实时干扰检测，定位以及自愈。gala-anteater 利用 gala-gopher 实时采集的资源层指标进行干扰检测和定位，上报的检测结果通过 rubik 实现资源分级保障对干扰进行快速恢复。



Aops 适配 authHub 统一用户鉴权

authhub 是一个基于 oauth2 协议开发的统一用户鉴权中心，Aops 通过 authHub 管理应用注册，实现应用间的统一用户鉴权。

功能描述



authHub 基于 oauth2 协议，实现统一用户鉴权中心，用户鉴权中心功能包括：

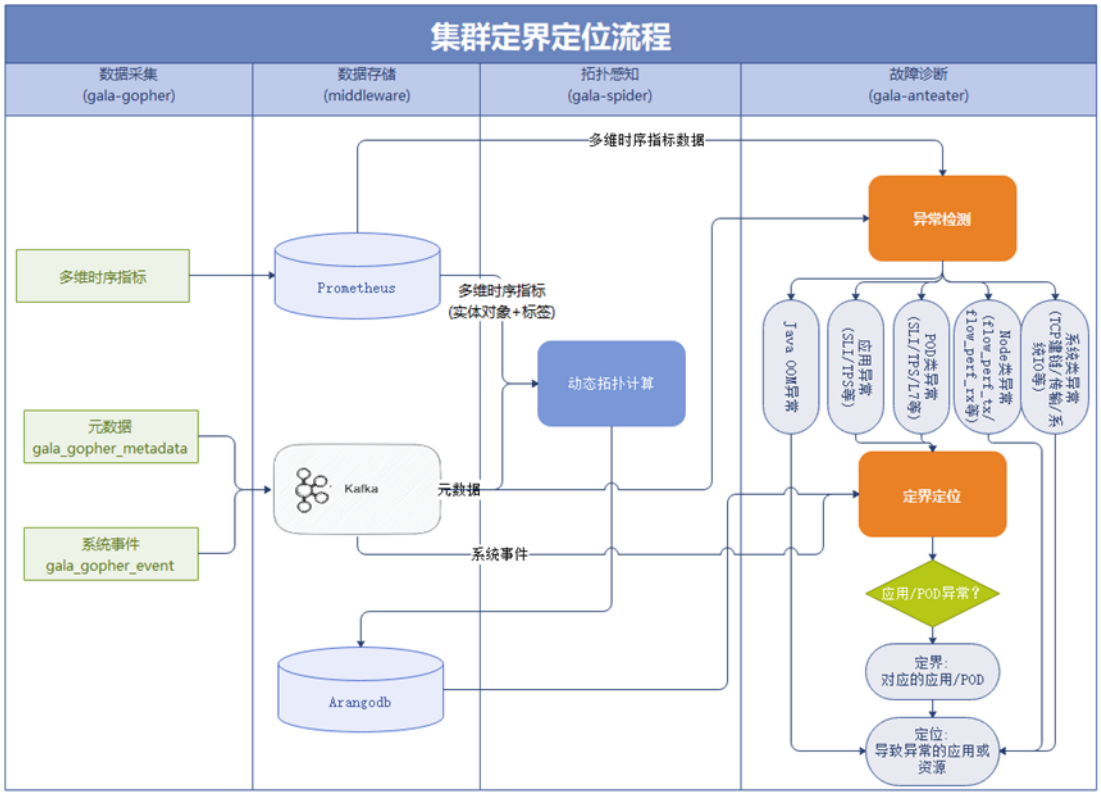
- 应用管理：应用部署后需要在 authHub 应用管理界面进行注册和配置，用户可以使用注册的应用的功能。
- 用户鉴权：在 authHub 管理的应用中，用户可以实现单点登录和单点登出。

应用场景

Aops 适配 authHub 统一用户鉴权，支持第三方应用现有的鉴权对接 authHub，提升应用的可拓展性；不同应用间的鉴权通过 authHub 统一管理，管理服务的访问权限。

微服务性能问题分钟级定界/定位（TCP，IO，调度等）

功能描述



基于拓扑的根因推导技术提供了大规模集群情况下的故障检测、根因定位服务，新增云原生场景故障定界定位能力，特别是针对网络 L7 层协议，如 HTTPS、PGSQL、MYSQL 等。这项技术通过 gala-gopher、gala-spider 和 gala-anteater 实现，新增内容提供了更细粒度的故障定界能力，通过这种能力运维团队可以快速定位问题源头，从而提高系统的稳定性和可靠性。该服务主要功能如下:

- 指标采集服务: gala-gopher 通过 ebpf 技术提供网络、IO 相关的指标采集上报功能。
- 集群拓扑构建服务: gala-spider 通过接受指标采集服务上报的数据信息，构建容器、进程级别的调用关系拓扑图构建。
- 故障检测服务:gala-anteater 通过故障检测模型对应用上报的指标采集进行分类，判断是否发生异常。
- 根因定位服务: gala-anteater 通过节点异常信息和拓扑图信息， 定位导致此次异

常的根因节点。

应用场景

分钟级别应用问题定界/定位场景主要包括以下几种情况：

- 云 K8S POD 部署场景：拥有云环境的企业可根据自身的需求和 IT 资源的现状选择模块。gala-gopher 采集 K8S 容器和进程级数据，gala-spider 基于采集数据构建拓扑图，gala-anteater 通过数据与拓扑图实现故障检测与根因定位。
- 裸金属部署场景：拥有裸金属部署场景的企业可根据自身的需求和 IT 资源的现状选择模块。gala-gopher 采集容器和进程级数据，gala-spider 基于采集数据构建拓扑图，gala-anteater 通过数据与拓扑图实现故障检测与根因定位。
- 大规模虚拟机场景：拥有大规模虚拟机场景的企业可根据自身的需求和 IT 资源的现状选择模块。gala-gopher 采集虚拟机数据，gala-spider 基于采集数据构建拓扑图，gala-anteater 通过数据与拓扑图实现故障检测与根因定位。

utsudo 项目发布

Sudo 是 Unix 和 Linux 操作系统中常用的工具之一，它允许用户在需要超级用户权限的情况下执行特定命令。然而， 传统 Sudo 在安全性和可靠性方面存在一些缺陷，为此 utsudo 项目应运而生。

utsudo 是一个采用 Rust 重构 Sudo 的项目，旨在提供一个更加高效、安全、灵活的提权工具，涉及的模块主要有通用工具、整体框架和功能插件等。

功能描述

基本功能：

- 访问控制：可以根据需求，限制用户可以执行的命令，并规定所需的验证方式。
- 审计日志：可以记录和追踪每个用户使用 utsudo 执行的命令和任务。
- 临时提权：允许普通用户通过输入自己的密码，临时提升为超级用户执行特定的命令或任务。

- 灵活配置：可以设置参数如命令别名、环境变量、执行参数等，以满足复杂的系统管理需求。

增强功能：

utsudo 在 openEuler 24.03 LTS SP1 版本中是 0.0.2 版本，当前版本主要功能有：

- 提权流程：把普通用户执行命令的进程，提权为 root 权限。
- 插件加载流程：实现了对插件配置文件的解析，以及对插件库的动态加载。

应用场景

通过部署 utsudo，管理员能够更好地管理用户权限，确保合适的授权级别，防止未经授权的特权操作，从而降低安全风险。

常见的应用场景有：系统管理维护、用户权限控制、多用户环境等。

utshell 项目发布

utshell 是一个延续了 bash 使用习惯的全新 shell，它能够与用户进行命令行交互，响应用户的操作去执行命令并给予 反馈。并且能执行自动化脚本帮助运维。

功能描述

基本功能：

- 命令执行：可以执行部署在用户机器上的命令，并将执行的返回值反馈给用户。
- 批处理：通过脚本完成自动任务执行。
- 作业控制：能够将用户命令作为后台作业，从而实现多个命令同时执行。并对并行执行的任务进行管理和控制。
- 历史记录：记录用户所输入的命令。
- 别名功能：能够让用户对命令起一个自己喜欢的别名，从而个性化自己的操作功能。

增强功能：

utshell 在 openEuler 24.03 LTS SP1 版本中是 0.5 版本，当前版本主要功能有：

- 实现对 shell 脚本的解析。
- 实现对第三方命令的执行。

应用场景

utshell 适用于传统服务器系统以及云原生环境，有人值守或无人值守下自动化脚本运维的生产环境。也适用于桌面命 令行爱好者的日常使用。

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 10.3 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

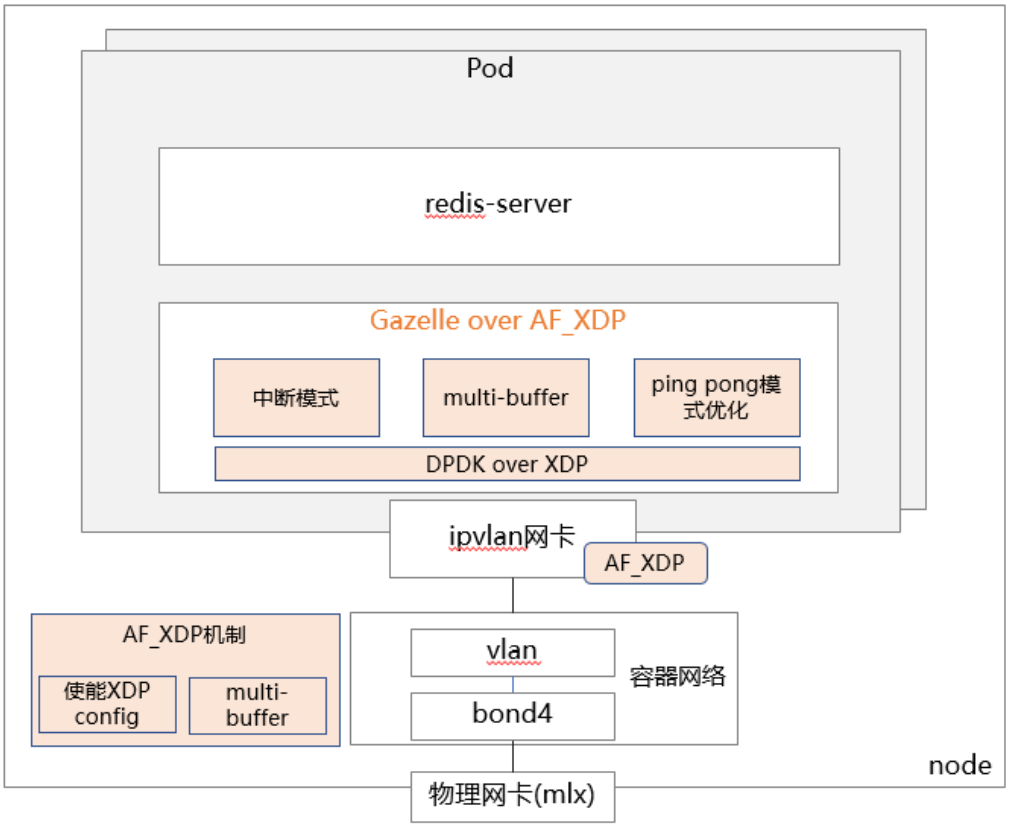
通用计算领域，运行 SPECCPU 2017 测试，相比于上游社区的 GCC 10.3 版本可获得 20%左右的性能收益。

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。

Gazelle 特性增强

Gazelle 是一款高性能用户态协议栈。它基于 DPDK 在用户态直接读写网卡报文，共享大页内存传递报文，使用轻量级 LwIP 协议栈。能够大幅提高应用的网络 I/O 吞吐能力。专注于数据库网络性能加速，兼顾高性能与通用性。本次版本新增容器场景 xdp 部署模式及 openGauss 数据库 tpcc 支持，丰富用户态协议栈。

功能描述



新增 xdp 网络部署模式

- 高性能（超轻量）：基于 dpdk、lwip 实现高性能轻量协议栈能力。
- 极致性能：基于区域大页划分、动态绑核、全路径零拷贝等技术，实现高线性度并发协议栈。
- 硬件加速：支持 TSO/CSUM/GRO 等硬件卸载，打通软硬件垂直加速路径。
- 通用性（posix 兼容）：接口完全兼容 posix api，应用零修改，支持 udp 的 recvfrom 和 sendto 接口。
- 通用网络模型：基于 fd 路由器、代理式唤醒等机制实现自适应网络模型调度，udp 多节点的组播模型，满足任意网络应用场景。
- 易用性（即插即用）：基于 LD_PRELOAD 实现业务免配套，真正实现零成本部署。
- 易运维（运维工具）：具备流量统计、指标日志、命令行等完整运维手段。

新增特性：

- 支持基于 ipvlan 的 l2 模式网卡上通过 xdp 的方式部署使用 Gazelle。
- 新增中断模式，可以支持在无流量或低流量场景下，lstack 不再占满 CPU 核。

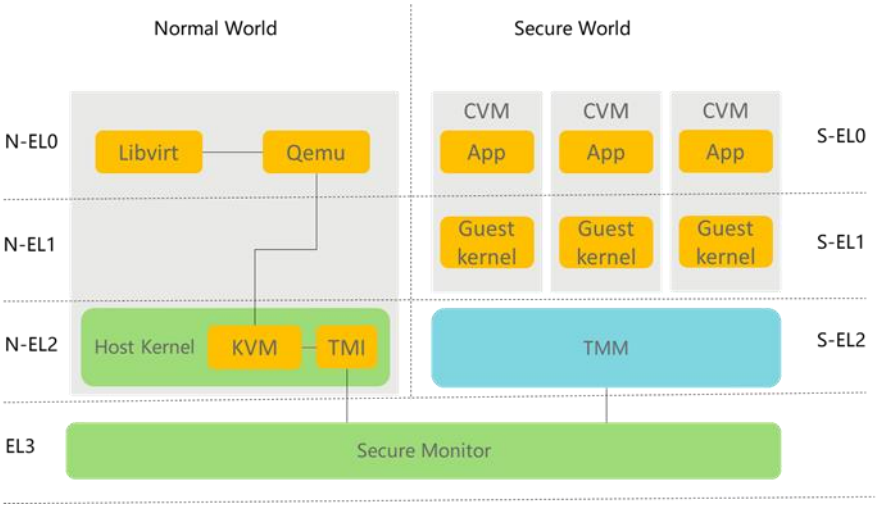
- 优化 pingpong 模式的网络，在数据包进行 pingpong 收发时，优化报文的发送行为。
- 新增对 openGauss 数据库的单机及单主备 tpcc 测试支持。

应用场景

适用于网络 IO 是性能瓶颈的应用，对 Redis、MySQL 等数据库场景有较好的性能提升效果。

virtCCA 机密虚机

virtCCA 机密虚机特性基于鲲鹏 920 系列 S-EL2 能力，在 TEE 侧实现机密虚机能力，实现现有普通虚机中的软件栈无缝迁移到机密环境中。



基于 Arm CCA 标准接口，在 Trustzone 固件基础上构建 TEE 虚拟化管理模块，实现机密虚机间的内存隔离、上下文管理、生命周期管理和页表管理等机制，支持客户应用无缝迁移到机密计算环境中。

技术约束:

1. 基于安全性考虑，宿主机 BIOS 中 CPU 超线程需要关闭；

功能描述

1. 设备直通：

设备直通是基于华为鲲鹏 920 系列通过预埋在 PCIE Root Complex 里的 PCIE 保护组件，在 PCIE 总线上增加选通器，对 CPU 与外设间的通信进行选通，即对 SMMU 的 Outbound Traffic 和 Inbound Traffic 的控制流和数据流进行控制，以此保证整体数据链路的安全性。

基于 virtCCA PCIPC 的设备直通能力，实现对 PCIE 设备的安全隔离和性能提升，存在以下优势：

1）安全隔离

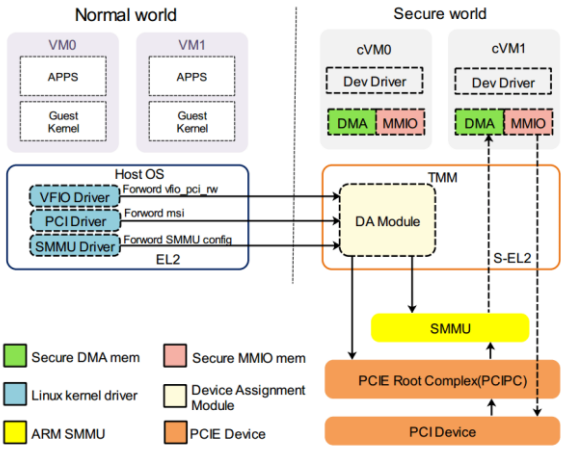
TEE 侧控制设备的访问权限，Host 侧软件无法访问 TEE 侧设备；

2）高性能

机密设备直通，相比业界加解密方案，数据面无损耗；

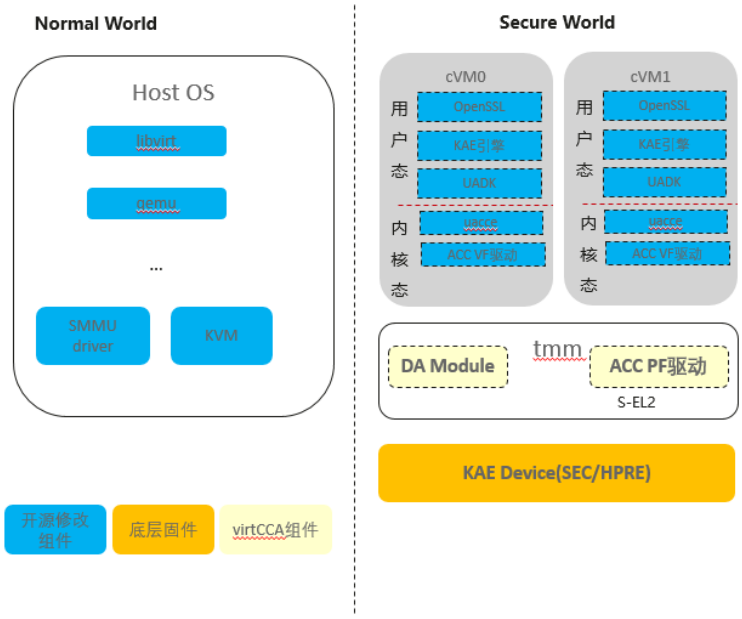
3）易用性

兼容现有开源 OS,无需修改开源 OS 内核驱动代码。



2. 国密硬件加速：

国密硬件加速是基于华为鲲鹏芯片，通过 KAE 加速器能力复用到安全侧，并采用 openEuler UADK 用户态加速器框架，提供客户机密虚机内国密加速性能提升以及算法卸载的能力。



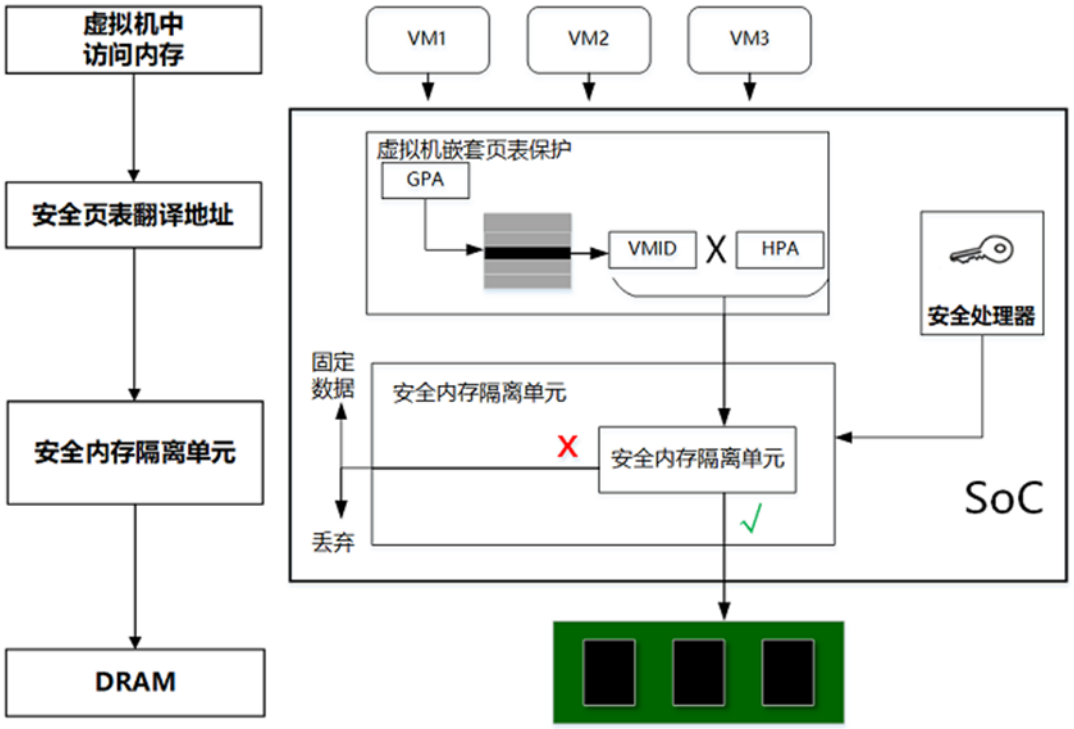
应用场景

面向密态数据库、安全云主机、机密多方计算、数据可信流通以及 AI 模型数据保护。

海光 CSV3 支持

海光第三代安全虚拟化技术(CSV3)在前二代技术的基础上继续增强，在CPU内部实现了虚拟机数据的安全隔离，禁止主机操作系统读写CSV3虚拟机内存，禁止主机操作系统读写虚拟机嵌套页表，保证了虚拟机数据的完整性，实现了CSV3虚拟机数据机密性和完整性的双重安全。

功能描述



CSV3架构图

安全内存隔离单元

安全内存隔离单元是海光第三代安全虚拟化技术的专用硬件，是实现虚拟机数据完整性的硬件基础。该硬件集成于CPU内部，放置于CPU核心访问内存控制器的系统总线路径上。该硬件可获取CSV3虚拟机安全内存的信息，包括内存物理地址，虚拟机VMID，及相关的权限等。CPU在访问内存时，访问请求先经过安全内存隔离单元做合法性检查，若访问允许，继续访问内存，否则访问请求被拒绝。

以CSV3架构图为例，在CSV3虚拟机运行过程中读取或写入内存时，先经过页表翻译单元完成虚拟机物理地址(GPA, Guest Physical Address)到主机物理地址(HPA, Host Physical Address)的转换，再向地址总线发出内存访问请求。访问请求中除了包含读写命令、内存地址(HPA)，还必须同时发出访问者的VM ID。

当CPU读取内存数据时，若安全内存隔离单元判断内存读取请求者的VMID错误，返回固定模式的数据。当CPU写入内存数据时，若安全内存隔离单元判断内存写入请求者的VMID错误，丢弃写入请求。

安全处理器

安全内存隔离单元是海光第三代安全虚拟化保护虚拟机内存完整性的核心硬件，对此硬件单元的配置必须保证绝对安全，无法被主机操作系统修改。

海光安全处理器是SoC内独立于主CPU之外的处理器，是CPU的信任根。安全处理器在CPU上电后，通过内置验签密钥验证固件的完整性，并加载执行。安全处理器具有独立硬件资源和封闭的运行环境，是CPU内最高安全等级硬件，管理整个CPU的安全。安全内存隔离单元的内容仅安全处理器有权限配置和管理，主机操作系统无权读写。

在虚拟机启动时，主机操作系统向安全处理器发送请求，安全处理器对安全内存隔离单元做初始配置。虚拟机运行期间，安全处理器固件更新安全内存隔离单元。虚拟机退出后，安全处理器清除安全内存隔离单元的配置。安全处理器会检查主机发来的配置请求是否合法，主机向安全处理器发起的任何非法配置请求，都会被拒绝。

虚拟机整生命周期内，安全内存隔离单元都在安全处理器的管理控制之下，保证了其配置的安全性。

Virtual Machine Control Block(VMCB)保护

Virtual Machine Control Block(VMCB)是虚拟机的控制信息块，保存了虚拟机ID(VMID)，虚拟机页表基地址，虚拟机寄存器页面基地址等控制信息，主机操作系统通过修改虚拟机控制信息能够影响并更改虚拟机内存数据。

CSV3增加了对虚拟机控制信息的保护，安全处理器负责创建VMCB，并配置于安全内存隔离单元的硬件保护之下。主机操作系统无法更改CSV3虚拟机VMCB的内容。

为更好的与主机操作系统的软件配合，CSV3创建了真实VMCB与影子VMCB页面。主机操作系统创建影子VMCB，填入控制信息，传递给安全处理器。安全处理器创建真实VMCB页面，复制影子VMCB中除虚拟机ID，虚拟机页表基地址，虚拟机寄存器页面基地址等关键信息之外的控制信息，并自行添加关键控制信息。虚拟机使用真实VMCB页面启动和运行，阻止了主机操作系统对虚拟机VMCB的攻击。

应用场景

海光CSV技术是基于安全虚拟化的TEE技术，其基于CPU硬件对TEE内的数据做安全保护。海光CSV使用国密SM4加密算法，保证虚拟机数据在内存中的机密性。经历3代演进，推出CSV1/CSV2/CSV3。CSV1使用C86处理器隔离的TLB、cache资源，数据在内存中被内存控制器上的SM4引擎加密；在CSV1的基础上，CSV2增加了虚拟机状态信息加密；CSV3在CSV2的基础上使用硬件隔离内存手段，进一步实现虚拟机数据的完整性。

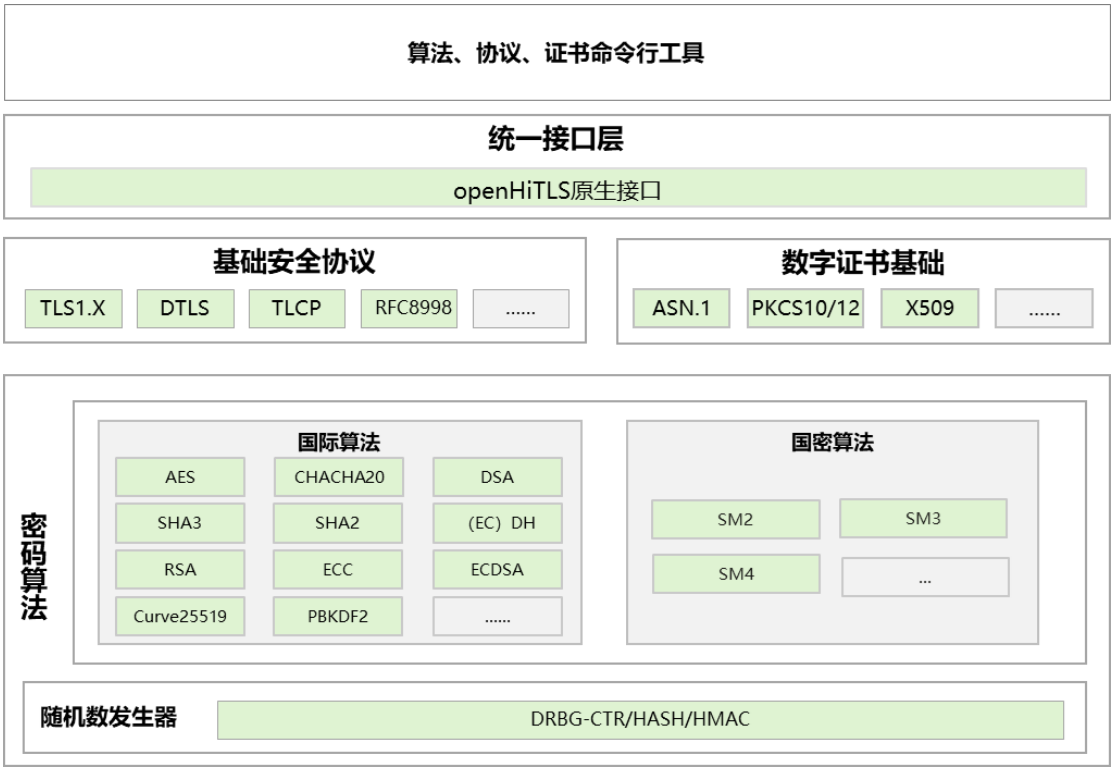
基于海光CSV技术的虚拟机支持启动度量、应用度量、远程认证、磁盘加密、秘钥封印等基础功能，广泛应用在云计算、隐私计算等行业。

在异构机密计算方面，海光推出CSV+DCU的完整闭环安全方案：DCU机密VRAM抵御非法访问、传输加密保证数据链路安全、远程认证证明DCU的身份信息。可以放心地应用

在政府机构、AI、医疗、金融、多方计算等异构加速场景中，为数据安全保驾护航。

密码套件 openHiTLS

openHiTLS 旨在通过提供轻量化、可裁剪的软件技术架构及丰富的国际主流及中国商用密码算法、协议，满足云计算、大数据、AI、金融等多样化行业的安全需求。它具备算法先进、性能卓越、安全可靠、开放架构及多语言平滑兼容等特点，为开发者提供了一套安全、可扩展的密码解决方案。通过社区共建与生态建设，openHiTLS 推动密码安全标准在各行各业的加速落地，同时构建以 openEuler 为核心的安全开源生态，为用户带来更加安全、可靠的数字环境。



功能描述

支持主流的密码协议和算法

支持国际主流及中国商用密码算法和协议，可根据场景需求选择合适的密码算法和协议。

- 支持中国商用密码算法：SM2、SM3、SM4；
- 支持国际主流算法：AES、RSA、(EC)DSA、(EC)DH、SHA3、HMAC 等；
- 支持 GB/T 38636-2020 TLCP 标准，即双证书国密通信协议；
- 支持 TLS1.2、TLS1.3、DTLS1.2 协议。

开放架构实现全场景密码应用覆盖

通过开放架构、技术创新及全产业链的应用实践，向产业界提供一站式全场景覆盖。

- 灵活南、北向接口：北向统一接口行业应用快速接入；南向设备抽象，广泛的运行在各类业务系统；
- 多语言平滑兼容：统一接口层（FFI）提供多语言兼容的能力，一套密码套件支持多种语言应用；
- 广泛的产品应用实践：全产业链应用场景密码技术实践，确保密码套件在不同场景下的高性能、高安全和高可靠。

分层分级解耦、按需裁剪，实现密码套件轻量化

加密内存成本无法回避，分层分级解耦实现密码算法软件极致成本。

- 分层分级解耦：TLS、证书、算法功能分层解耦、算法抽象、调度管理、算法原语分级解耦、高内聚低耦合、按需组合；
- 高级抽象接口：提供高级抽象接口，避免算法裁剪引入对外接口变更，降低软件成本前提下，保持对外接口不变；
- 极致成本：按需裁剪，特性依赖关系自动管理，可实现 PBKDF2 + AES 算法 BIN20K、堆内存 1K、栈内存 256 字节。

密码算法敏捷架构，应对后量子迁移

通过密码算法敏捷架构、技术创新实现应用快速迁移和先进算法的快速演进。

- 统一北向接口:提供对算法标准化、可扩展的接口，避免算法切换造成接口变动，上层业务应用需要大范围适配新接口；
- 算法插件化管理框架:实现对算法插件化管理，算法 Provider 层支持算法运行时动态加载，提供热加载算法的能力；
- 算法使用配置化:支持通过配置文件获取算法信息，可避免算法标识代码硬编码。

应用场景

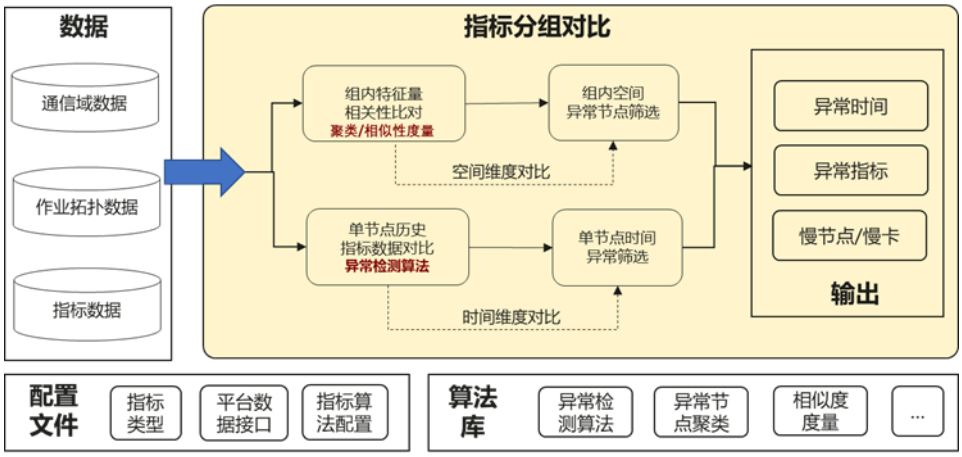
openHiTLS 密码套件的可应用业务场景广泛，典型应用场景如下：

- 云计算：openHiTLS 通过模块化和可裁剪的设计，能够灵活适应云计算环境的安全需求。它可以为云平台提供高强度的数据加密、身份认证和完整性校验等功能，确保云上数据的安全性和完整性。
- 大数据：在大数据处理过程中，数据的安全性和隐私保护至关重要。openHiTLS 密码套件提供了丰富的密码协议和算法选择，可以实现对大数据的加密存储、传输和处理，有效防止数据泄露和非法访问。
- 物联网（IoT）：物联网设备数量庞大，且通常资源受限。openHiTLS 通过轻量级、可剪裁的软件技术架构，可以适应物联网设备的低功耗、低资源需求的特点，为物联网设备提供安全可靠的通信和数据保护。
- 金融：金融行业对安全性有着极高的要求。openHiTLS 密码套件支持中国商用密码算法，并符合国际安全标准，可以为金融系统提供高强度的数据加密、身份认证和交易安全等功能，确保金融交易的安全性和合规性。

AI 集群慢节点快速发现 Add Fail-slow Detection

AI 集群在训练过程中不可避免会发生性能劣化，导致性能劣化的原因很多且复杂。现有方案是在发生性能劣化之后利用日志分析，但是从日志收集到问题定界根因诊断以及现网闭环问题需要长达 3-4 天之久。基于上述痛点问题，我们设计了一套在线慢节点定界方案，该方案能够实时在线观测系统关键指标，并基于模型和数据驱动的算法对观测数据进行实时分析给出劣慢节点的位置，便于系统自愈或者运维人员修复问题。

功能描述



基于分组的指标对比技术提供了 AI 集群训练场景下的的慢节点/慢卡检测能力。这项技术通过 gala-anteater 实现，新增内容包括配置文件、算法库、慢节点空间维度对比算法和慢节点时间维度对比，最终输出慢节点异常时间、异常指标以及对应的慢节点/慢卡 ip，从而提高系统的稳定性和可靠性。该服务主要功能如下：

- 配置文件：主要包括待观测指标类型、指标算法配置参数以及数据接口，用于初始化慢节点检测算法。
- 算法库：包括常用的时序异常检测算法 spot 算法，k-sigma 算法，异常节点聚类算法和相似度度量算法。
- 数据：包括指标数据、作业拓扑数据以及通信域数据，指标数据表示指标的时序序列，作业拓扑数据表示训练作业所用的节点信息，通信域数据表示节点通信的连接关系，包括数据并行、张量并行和流水线并行等。
- 指标分组对比：包括组内空间异常节点筛选和单节点时间异常筛选。组内空间异常节点筛选根据异常聚类算法输出异常节点；单节点时间异常筛选根据单节点历史数据进行时序异常检测判断节点是否异常。

应用场景

gala-anteater支持慢节点异常上报，告警展示，异常信息展示。

AI模型训练场景：适用于AI模型大规模集群训练任务，通过该功能可快速发现慢节点，便于系统自愈或者运维人员修复问题。

AI模型推理场景：适用于单一模型的多实例性能劣化检测，通过多实例间应用资源的

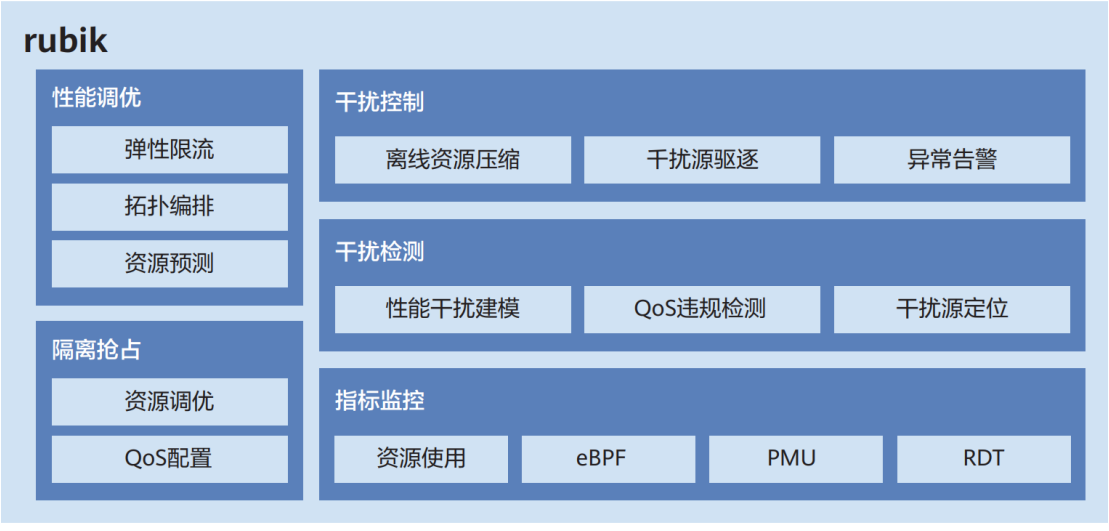
对比情况，可快速发现性能劣化的实例，便于推理作业的调度和资源利用率的提升。

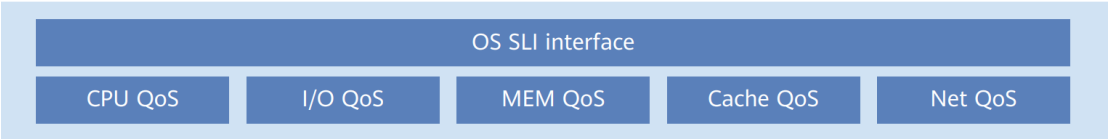
rubik 在离线混部调度协同增强

云数据中心资源利用率低（<20%）是行业普遍存在的问题，提升资源利用率已经成为了一个重要的技术课题。将业务区分优先级混合部署（简称混部）运行是典型有效的资源利用率提升手段。然而，将多种类型业务混合部署能够显著提升集群资源利用率，也带来了共峰问题，会导致关键业务服务质量（QoS）受损。因此，如何在提升资源利用率之后，保障业务 QoS 不受损是技术上的关键挑战。

rubik 是 openEuler 提供的容器混部引擎，提供一套自适应的单机算力调优和服务质量保障机制，旨在保障关键业务服务质量不下降的前提下，提升节点资源利用率。

功能描述





- Cache 及内存带宽控制：支持对低优先级虚拟机的 LLC 和内存带宽进行限制，当前仅支持静态分配。
- CPU 干扰控制：支持 CPU 时间片 us 级抢占及 SMT 干扰隔离，同时具有防优先级反转能力。

- 内存资源抢占：支持在节点 OOM 时优先杀死离线业务，从而保证在线业务的服务质量。
- memcg 异步内存回收：支持限制混部时离线应用使用的总内存，并在在线内存使用量增加时动态压缩离线业务内存使用。
- QuotaBurst 柔性限流：支持关键在线业务被 CPU 限流时允许短时间突破 limit 限制，保障在线业务运行的服务质量。
- PSI 指标观测增强：支持 cgroup v1 级别的压力信息统计，识别和量化资源竞争导致的业务中断风险，支撑用户实现 硬件资源利用率提升。
- IOCost 限制业务 io 权重：支持限制混部是离线业务的磁盘读写速率，防止离线业务争抢在线业务的磁盘带宽，从而提升在线业务服务质量。
- CPI 指标观测：支持通过观察 CPI 指标统计当前节点的压力，识别并驱逐离线业务以保证在线应用的服务质量。

此版本新增以下特性：

- 节点 CPU/内存干扰控制和驱逐：支持通过观测当前节点的 CPU 和内存水位，在资源紧张情况下通过驱逐离线业务，保证节点水位安全。

应用场景

业务可根据时延敏感性分为高优先级业务和低优先级业务，将业务区分优先级混合部署以提高资源利用率。高优先级 虚拟机业务推荐：时延敏感类业务，如 web 服务、高性能数据库、实时渲染、机器学习推理等。低优先级虚拟机业务推荐： 非时延敏感类业务，如视频编码、大数据处理、离线渲染、机器学习训练等。

CFGO 反馈优化特性增强

日益膨胀的代码体积导致当前处理器前端瓶颈成为普遍问题，影响程序运行性能。编译器反馈优化技术可以有效解决此类问题。

CFGO（Continuous Feature Guided Optimization）是 GCC for openEuler 和毕昇编译器的反馈优化技术名，指多模态（源代码、二进制）、全生命周期（编译、链接、链接后、运行时、OS、库）的持续反馈优化，主要包括以下两类优化技术：

- 代码布局优化：通过基本块重排、函数重排、冷热分区等技术，优化目标程序的二进制布局，提升 i-cache 和 i-TLB 命中率。
- 高级编译器优化：内联、循环展开、向量化、间接调用等提升编译优化技术受益于反馈信息，能够使编译器执行更精确的优化决策。

功能描述

GCC CFGO 反馈优化共包含三个子特性：CFGO-PGO、CFGO-CSPGO、CFGO-BOLT，通过依次使能这些特性可以缓解处理前端瓶颈，提升程序运行时性能。为了进一步提升优化效果，建议 CFGO 系列优化与链接时优化搭配使用，即在 CFGO-PGO、CFGO-CSPGO 优化过程中增加-flto=auto 编译选项。

- CFGO-PGO

CFGO-PGO 在传统 PGO 优化的基础上，利用 AI4C 对部分优化遍进行增强，主要包括 inline、常量传播、去虚化等优化，从而进一步提升性能。

- CFGO-CSPGO

PGO 的 profile 对上下文不敏感，可能导致次优的优化效果。通过在 PGO 后增加一次 CFGO-CSPGO 插桩优化流程，收集 inline 后的程序运行信息，从而为代码布局和寄存器优化等编译器优化遍提供更准确的执行信息，实现性能进一步提升。

- CFGO-BOLT

CFGO-BOLT 在基线版本的基础上，新增 aarch64 架构软件插桩、inline 优化支持等优化，最终进一步提升性能。

应用场景

CFGO 通用性较好，适用于整机性能瓶颈在 CPU 上，且 CPU 瓶颈在前端的 C/C++应用，如数据库、分布式存储等场景，一般可以取得 5~10%性能提升。

AI4C 编译选项调优和 AI 编译优化提升典型应用性能

AI4C（AI for Compiler）代表 AI 辅助编译优化套件，是一个使用 AI 技术优化编译选项和优化遍关键决策的软件框架，旨在突破当前编译器领域的两个关键业务挑战：

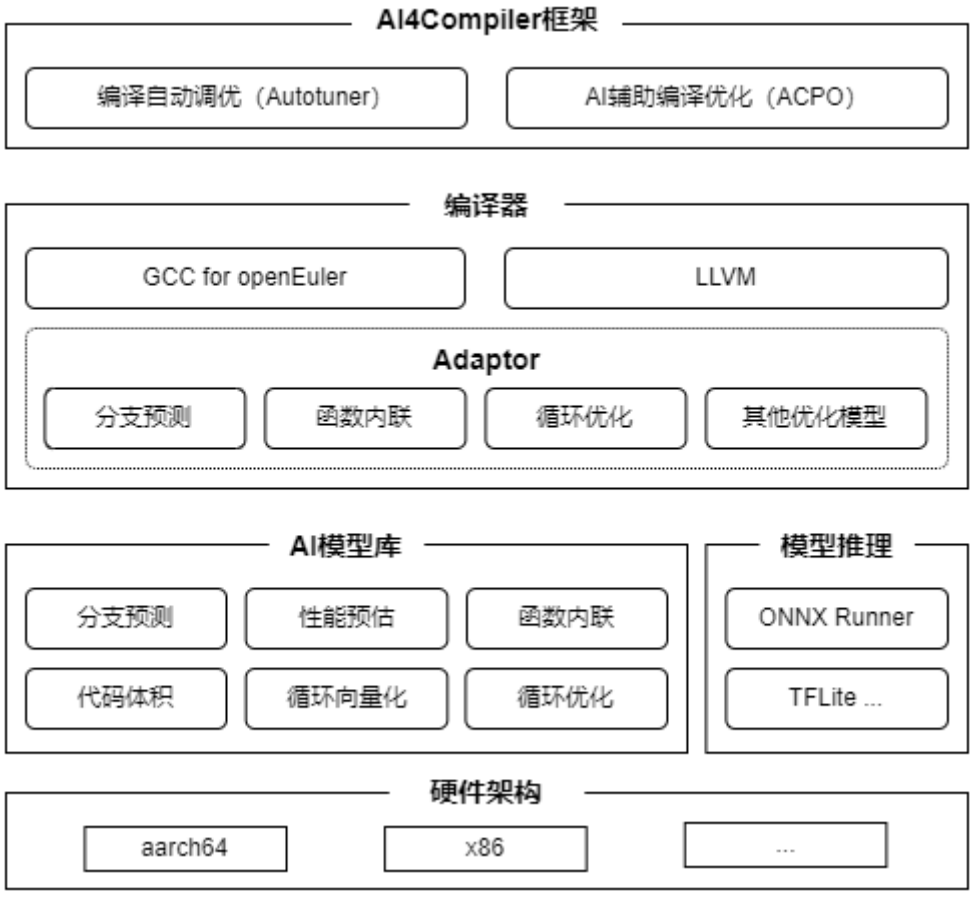
1. 性能提升困难：传统编译器优化开发周期长，新的编译优化技术与已有编译优化过程难以兼容达到 1+1>=2 的效果，导致性能提升无法达到预期效果。

2. 调优效率低下：硬件架构或者软件业务场景变更，需要根据新的负载条件投入大量人力成本，重新调整编译优化的成本模型，导致调优时间长。

功能描述

AI4Compiler 框架提供编译选项自动调优和 AI 模型辅助编译优化两个主要模块。

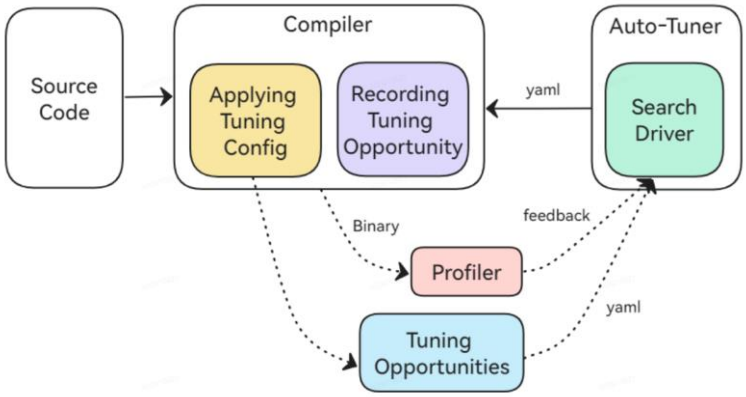
软件三层架构如下图所示，框架上层调度模块驱动中层编译器核心优化过程，通过不同编译器各自的适配模块调用底层 AI 模型和模型推理引擎，以优化特性相关数据和硬件架构参数作为模型输入特征运行模型推理，获得编译过程关键参数最佳决策，从而实现编译优化。



- 编译自动调优（Autotuner）

AI4C 的自动调优基于 OpenTuner（2015 Ansel et al.）开发，通过插件驱动编译器采集优化特性相关参数信息，通过搜索算法调整关键决策参数（例如循环展开系数），通过插件注入编译过程修改决策，运行编译输出二进制获得反馈因子，迭代自动调优。

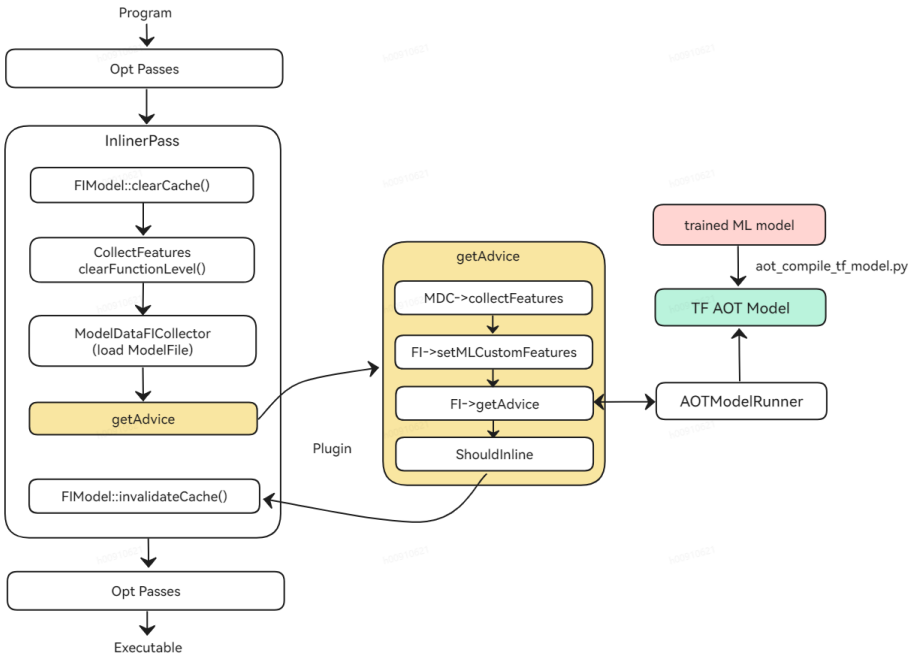
- 已集成一系列搜索算法，动态选择算法并共享搜索进；
- 支持用户配置 yaml 自定义搜索空间和扩展底层搜索算法；
- 支持细粒度代码块调优与粗粒度编译选项自动调优；
- 在 cormark、dhrystone、Cbench 等 benchmark 上获得 3%~5%不等的收益。



- AI 辅助编译优化（ACPO）

ACPO 提供全面的工具、库、算法，为编译器工程师提供简单易用的接口使用 AI 模型能力，替代或增强编译器中启发式优化决策算法。在编译器优化过程中，使用插件提取优化遍的输入结构化数据作为模型输入特征，getAdvice 运行预训练模型获得决策系数，编译器使用模型决策结果替代部分启发式决策，获得更好的性能。

- 解耦编译器与 AI 模型和推理引擎，帮助算法开发者专注 AI 算法模型开发，简化模型应用成本，同时兼容多个编译器、模型、AI 推理框架等主流产品，提供 AI 模型的热更新能力；
- 实践落地 IPA Function Inline、RTL Loop Unroll 等不同优化阶段和优化过程，获得相对显著的收益。



应用场景

AI4C 提供优化模型和调优插件实现不同应用的编译优化，支持开发者根据不同应用的性能瓶颈开发相应的优化模型，使用 AI4C 框架能力注入编译过程，实现性能提升，当前 AI4C 主要应用于数据库等互联网后端领域。

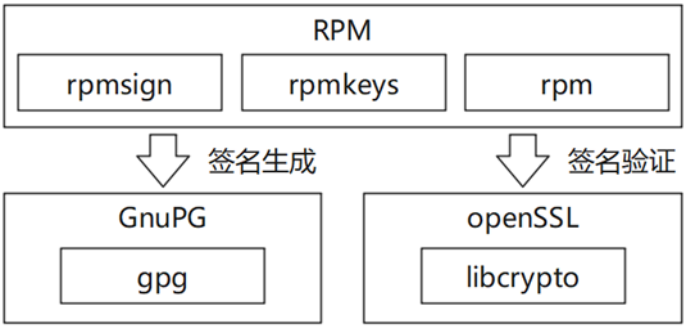
- RocksDB
- MySQL
- Doris
- Nginx
- Python
- ...

RPM 国密签名支持

根据国内相关安全技术标准，在某些应用场景中需要采用国密算法实现对重要可执行程序来源的真实性和完整性保护。openEuler 当前采用 RPM 格式的软件包管理，软件包签名算法基于 openPGP 签名规范。openEuler 24.03 LTS SP1 版本基于 RPM 包管理机制扩展对于 SM2 签名算法和 SM3 摘要算法的支持。

功能描述

本特性主要基于RPM组件以及其调用的GnuPG2签名工具，在现有openPGP签名体系的基础上，进行国密算法使能。特性涉及的软件包如下：



在RPM软件包签名场景，用户可调用gpg命令生成SM2签名私钥和证书，并调用rpmsign命令为指定的RPM包添加基于SM2+SM3算法的数字签名。

在RPM软件包验签场景，用户可调用rpm命令导入验签证书，并通过校验RPM包的数字签名信息从而验证软件包的真实性和完整性。

应用场景

用户可基于本特性提供的软件包国密签名/验签技术，构建基于国密算法的软件包验证能力，满足国内部分合规场景中的密码应用安全要求。

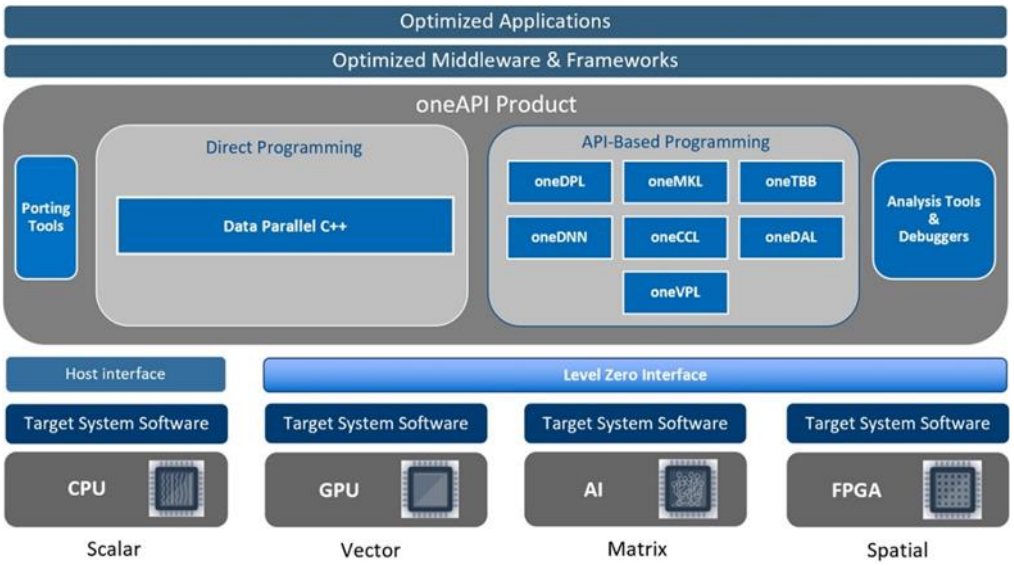
oneAPI 框架支持

Unified Acceleration Foundation（UXL）正在推动构建开放的异构加速软件框架的标准化。其中oneAPI作为初始项目的目标是提供一种跨行业、开放、基于标准的统一编程模型，并为异构加速器（如 CPU、GPU、FPGA 和专用加速器）提供统一的开发体验。oneAPI规范扩展了现有的开发者编程模型，通过并行编程语言（Data Parallel C++）或者一组加速软件库以及底层的硬件抽象接口（Level Zero）来支持跨架构的编程，从而支持多种加速硬件和处理器平台。为了提供兼容性并提高开发效率，oneAPI 规范基于行业标准，提供了多种开放的、跨平台和易用的开发者软件套件。

功能描述

为了在openEuler上完整的支持oneAPI，我们从openEuler 24.03 LTS开始分别集成了

oneAPI的开发环境Basekit和运行态环境的Runtime容器镜像。并从openEuler 24.03 LTS SP1开始，openEuler原生支持了oneAPI系列底层框架库的适配和集成，包括oneAPI运行所需的各类依赖库，以及英特尔的图形加速编译器，以及OpenCL的runtime，和支持不同平台（x86_64和aarch64）的底层硬件抽象层（Level-Zero）等。同时为了完整支持Data Parallels C++和基于加速库的API的编程模式，我们也对oneAPI官方提供的各类软件包在openEuler上做了相应的适配和验证的工作，这样我们可以方便的通过在openEuler中增加oneAPI的官方DNF/YUM源来安装和更新所有的oneAPI的运行依赖、开发工具和调试工具等。



应用场景

- 跨平台的编程框架：oneAPI的统一编程框架可以将DPC++实现的程序快速的通过编译器移植到不同的平台上运行。同时，因为oneAPI是一套跨架构编程和高性能与可用性结合的编程框架，所以oneAPI可以用于结合CPU、GPU、FPGA和NPU等的异构加速器能力，从而应用于各类跨平台的高性能计算场景。
- 人工智能与深度学习：oneAPI已经和主流的AI框架进行了整合，在需要使用异构的硬件加速的场景下可以很容易的通过适配 oneAPI Level Zero，从而充分利用硬件的加速能力运行各类AI框架和模型。
- 高性能计算、科学计算与仿真：oneAPI 提供了丰富的数学库和并行计算工具，可以加速线性代数、微分方程等数值计算任务。而且在数学、物理学、工程学等科研和工业领域，oneAPI 可以用于构建复杂系统的仿真模型和利用硬件（如Intel Xeon处理器的AMX加速指令）对计算进行加速。oneAPI 适配了业界常用的各类加速库，从而也可以用于优化类似CAE等工业软件的加速，并提高仿真分析的速度和精度。

仓库地址

https://gitee.com/openeuler/intel-oneapi

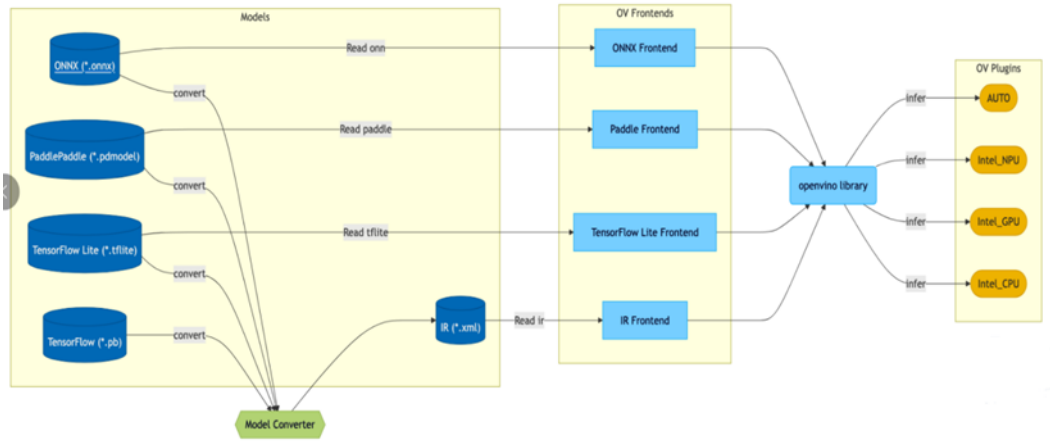
OpenVINO 支持

OpenVINO是一套开源的AI工具套件和运行库，其能用于优化几乎各类主流框架的深度学习模型，并在各种Intel处理器和加速器以及其他硬件平台如ARM上以最佳性能进行部署并高效提供AI服务。我们从openEuler 24.03 LTS SP1开始，提供了OpenVINO原生的适配和集成，从而在openEuler上提供了完整的OpenVINO的计算能力。

功能描述

OpenVINO的模型转化功能可以将已经使用流行框架（如 TensorFlow、PyTorch、ONNX和PaddlePaddle等）训练的模型进行转换和优化。并在各类的Intel处理器和加速器或者ARM的硬件环境中进行部署，包括在本地、设备端、浏览器或云端等场景下提供服务能力。其主要功能包括：

- 预训练模型库、模型转化和优化：支持OpenVINO的Open Model Zoo提供了数百种开源的预训练模型，可以快速的运行在OpenVINO运行框架下。OpenVINO支持将多种深度学习框架训练得到的模型转换为中间表示（IR）格式，以便在 OpenVINO 中进行优化。比如OpenVINO通过NNCF对模型剪枝或者稀疏性压缩优化，减少模型大小，提高推理速度等。同时OpenVINO也提供了性能分析工具用于分析模型性能和瓶颈等。
- 硬件加速：将优化后的模型部署到Intel或者其他支持的硬件上，充分利用硬件的并行计算能力，加速推理过程。同时，OpenVINO提供了Plugin机制，可以扩展和实现第三方的硬件计算加速能力。
- OpenVINO运行时（Runtime）支持：OpenVINO推理引擎运行时支持跨硬件和跨操作系统和多语言（C、C++、Python等），并能提供异步推理能力，提高系统吞吐量。
- OpenVINO GenAI 是 OpenVINO 的一个新分支，旨在简化生成式 AI 模型的推理过程。它隐藏了生成过程的复杂性，并最大限度地减少了所需的代码量。OpenVINO GenAI 是一个由流水线和方法组成的库，扩展了 OpenVINO 运行时，使其能够更有效地处理生成式 AI 模型，从而快速的提供生成式AI的能力。



应用场景

OpenVINO是一套支持现有主流训练模型的推理加速和应用场景部署的框架，所以其可以广泛的应用在本地、设备端、浏览器或云端的AI模型部署的场景。

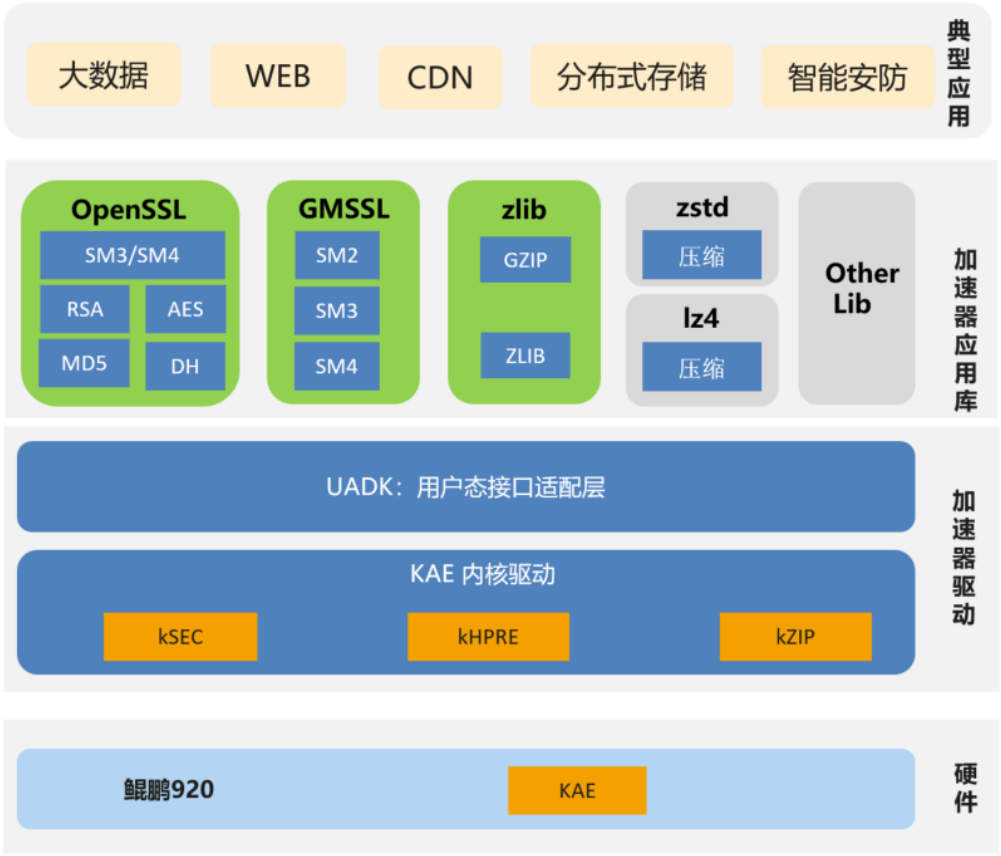
- 用于AI PC的基础AI框架：OpenVINO包含了一套成熟的开源和开放的推理优化和部署框架，由于其跨操作系统、跨硬件平台和跨语言编程的能力，OpenVINO完全可以作为一套主流的AI PC的基础AI框架，用于在AI PC中深度集成各类预训练的模型，并用于支撑AI PC上从操作系统到应用软件的底层AI能力的实现。
- 生成式AI：新的Gen AI API可以快速的结合已有的生成式AI模型来部署文本到图形、语音识别或转文字、视觉语言模型等各类的生成式AI的应用场景。
- 行业AI和边缘计算AI：OpenVINO也提供了边缘计算AI的开发套件，利用 OpenVINO 工具包可以高效的开发零售、制造和医疗等行业的应用。

  仓库地址

  https://gitee.com/openeuler/intel-openvino

鲲鹏 KAE 加速器

鲲鹏加速引擎 KAE(Kunpeng Accelerator Engine)是基于鲲鹏 920 处理器提供的硬件加速器解决方案，包含了 KAE 加解密和 KAEZip。KAE 加解密和 KAEZip 分别用于加速 SSL(Secure Sockets Lyaer) / TLS(Transport Layer Security)应用和数据压缩，可以显著降低处理器消耗，提高处理器效率。此外，鲲鹏加速引擎对应用层屏蔽了其内部细节，用户通过 OpenSSL、zlib 标准接口即可实现快速迁移现有业务。鲲鹏加速引擎的系统逻辑架构如下图所示。



各 KAE 相关模块详细功能描述如下表所示。

模块名称	功能描述
UADK	KAE 的用户态接口适配层，提供用户态的 API 接口
ksec/khpre/kzip	KAE 的内核态设备驱动
openSSL	KAE-engine 适配的 openSSL 库使能 KAE 加解密加速。
zlib	标准开源 zlib 库适配 KAE 压缩加速，提供标准开源 zlib 库 API 接口。

功能描述

KAE 加解密是鲲鹏加速引擎的加解密模块，使用鲲鹏加速引擎实现 RSA/SM3/SM4/DH/MD5/AES 算法，结合无损用户态驱动框架，提供高性能对称加解密、非对称加解密算法能力，兼容 OpenSSL 1.1.1x 系列版本，支持同步&异步机制。目前主要支持以下算法：

- 摘要算法 SM3/MD5，支持异步模型。
- 对称加密算法 SM4，支持异步模型，支持 CTR/XTS/CBC/ECB/OFB 模式。
- 对称加密算法 AES，支持异步模型，支持 ECB/CTR/XTS/CBC 模式。
- 非对称算法 RSA，支持异步模型，支持 Key Sizes 1024/2048/3072/4096。

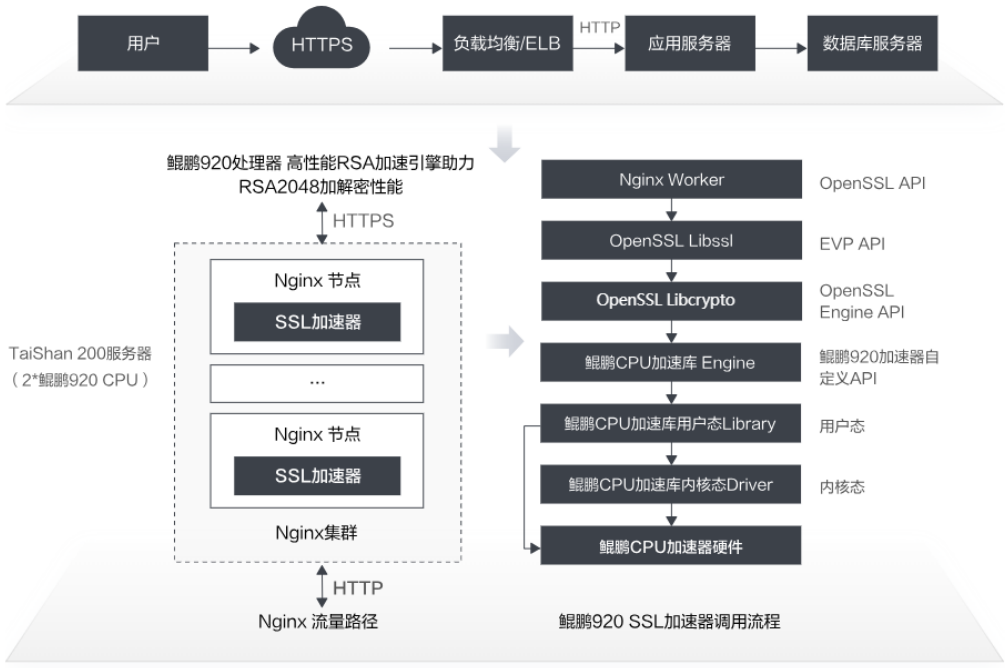
- 密钥协商算法 DH，支持异步模型，支持 Key Sizes 768/1024/1536/2048/3072/4096。

KAEzip 是鲲鹏加速引擎的压缩模块，使用鲲鹏硬加速模块实现 deflate 算法，结合无损用户态驱动框架，提供高性能 Gzip/zlib 格式压缩接口。通过加速引擎可以实现不同场景下应用性能的提升，例如在分布式存储场景下，通过 zlib 加速库加速数据压缩和解压。

- 支持 zlib/Gzip 数据格式，符合 RFC1950/RFC1952 标准规范。
- 支持同步模式。
- 单处理器（鲲鹏 920 处理器）最大压缩带宽 7GB/s，最大解压带宽 8GB/s。
- 支持的压缩比≈2，与 zlib 1.2.11 接口保持一致。

应用场景

鲲鹏 KAE 加速引擎可应用于 web 的 SSL 加速、大数据的 HDFS 压缩存储、分布式存储的压缩存储等解决方案场景。如下图所示，鲲鹏 KAE 加速 web 应用场景，鲲鹏 KAE 提供 openSSL 配套 engine，实现 RSA/AES/SM3/SM4 的加密算法硬加速，客户应用无需修改直接调用原有 openSSL 接口，实现 https 的安全连接加速。



8. 著作权说明

9. 商标

10.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