



openEuler 21.09

技术白皮书

目录

CONTENTS

<u>01</u>	概述	<u>02</u>	<u>07</u>	增强特性	<u>21</u>
<u>02</u>	平台架构	<u>05</u>	<u>08</u>	社区治理	<u>25</u>
<u>03</u>	运行环境	<u>08</u>	<u>09</u>	著作权说明	<u>32</u>
<u>04</u>	内核创新	<u>10</u>	<u>10</u>	商标	<u>32</u>
<u>05</u>	云化基座	<u>14</u>	<u>11</u>	附录	<u>32</u>
<u>06</u>	场景创新	<u>18</u>			

01/ 概述



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

欧拉开源社区通过开放的社区形式与全球的开发者共同构建一个开放、多元和架构包容的软件生态体系, 孵化支持多种处理器架构、覆盖数字设施全场景, 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。

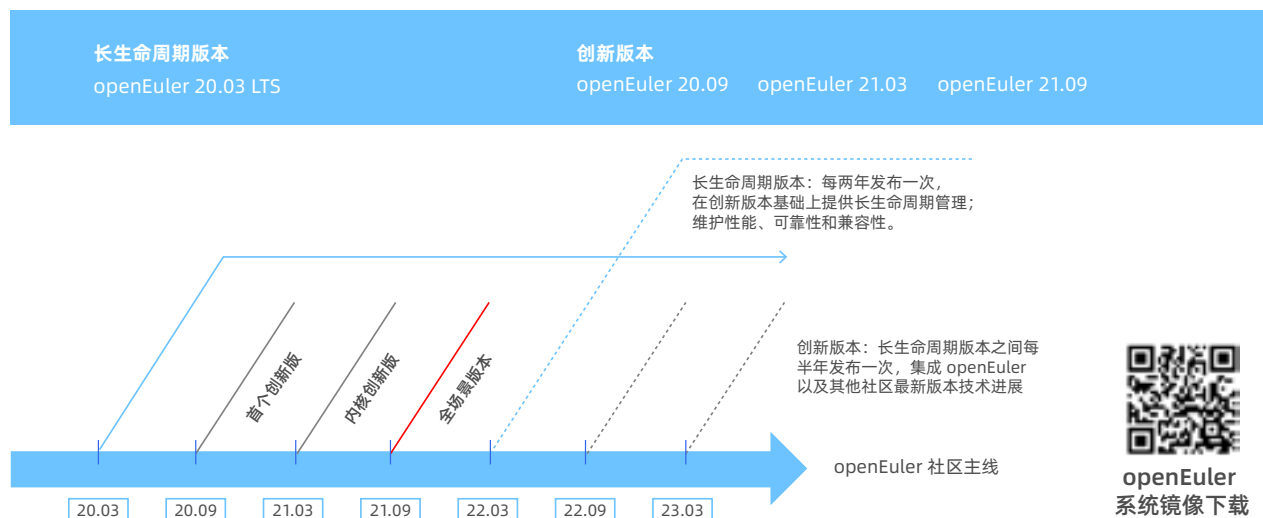
2020 年 3 月 30 日, openEuler 20.03 LTS (Long Term Support, 简称为 LTS, 中文为长生命周期支持) 版本正式发布, 为 Linux 世界带来一个全新的具备独立技术演进能力的 Linux 发行版。

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

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

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

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

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

openEuler 覆盖全场景的创新平台

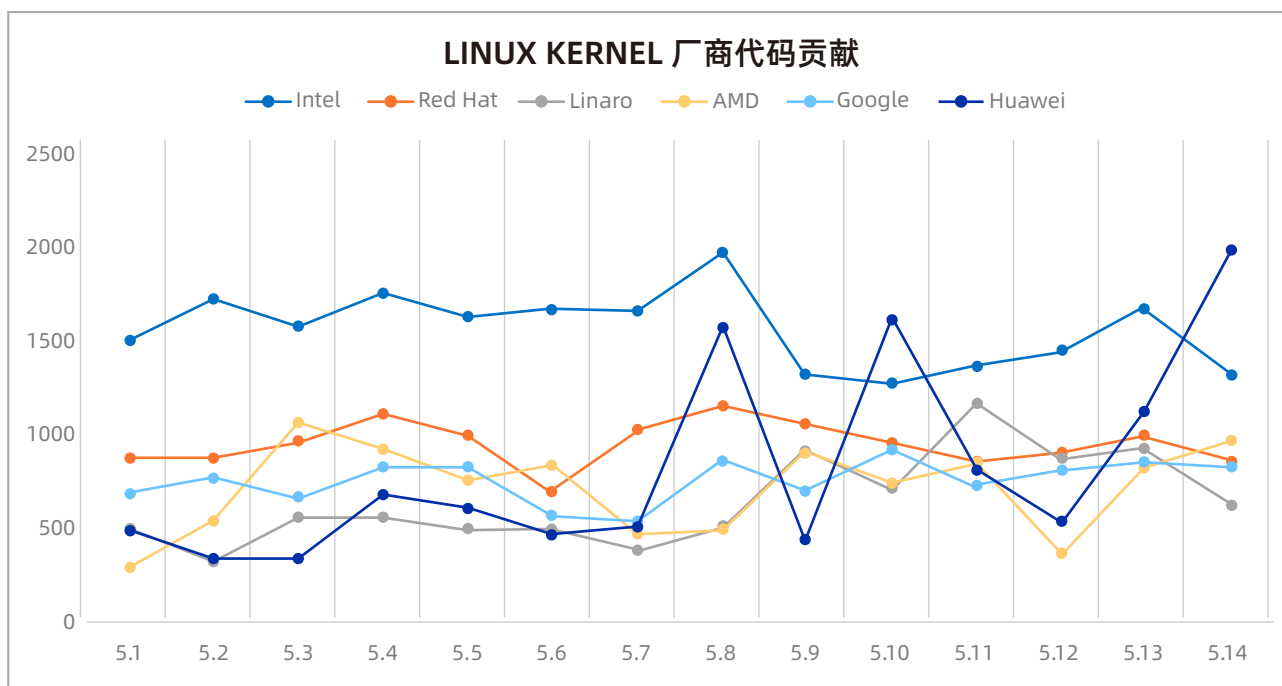


openEuler 已支持 X86、Arm、RISC-V 多处理器架构，未来还会扩展 PowerPC、SW64 等更多芯片架构支持，持续完善多样化算力生态体验。

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 正成为覆盖全场景的操作系统，将发布面向边缘计算的版本 openEuler 21.09 Edge、面向嵌入式的版本 openEuler 21.09 Embedded。

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

openEuler 对 Linux Kernel 的持续贡献



openEuler 内核研发团队持续贡献 Linux Kernel 上游社区，回馈主要集中在：芯片架构、ACPI、内存管理、文件系统、Media、内核文档、针对整个内核质量加固的 bug fix 及代码重构等内容。十余年来总计向社区贡献 17000+ 补丁。

在 Linux Kernel 5.10 和 5.14 版本中，openEuler 内核研发团队代码贡献量排名全球第一。坚持内核创新，持续贡献上游社区。

openEuler 开放透明的开源软件供应链管理

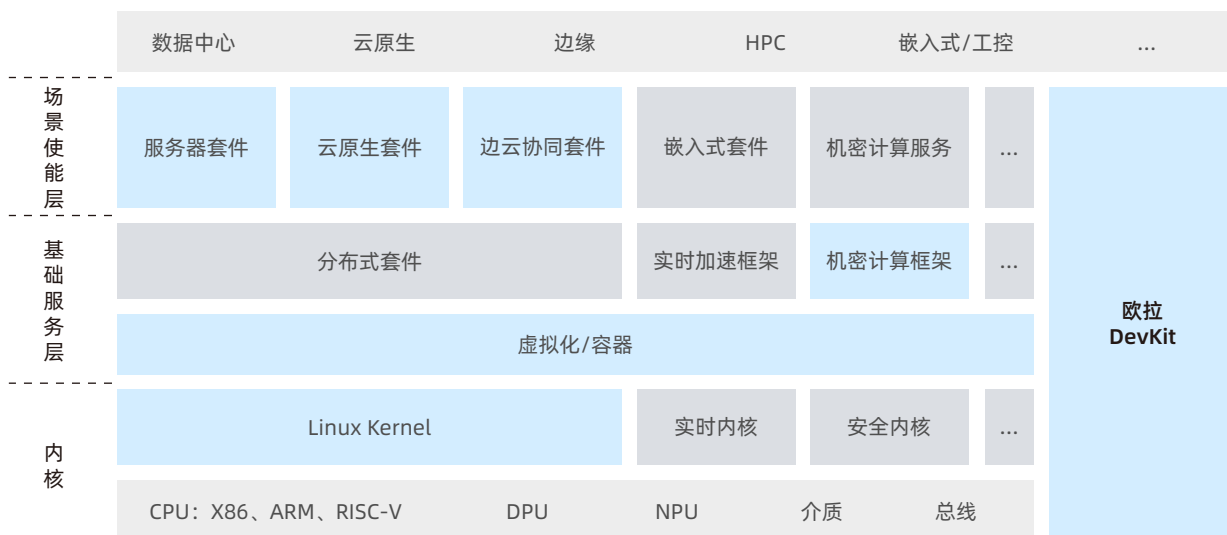
开源操作系统的构建过程，也是供应链聚合优化的过程。拥有可靠开源软件供应链，是大规模商用操作系统的基础。openEuler 从用户场景出发，回溯梳理相应的软件依赖关系，理清所有软件包的上游社区地址，源码和上游对应验证。完成构建验证、分发、实现生命周期管理。开源软件的构建、运行依赖关系，上游社区，三者之前形成闭环且完整透明的软件供应链管理。



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。



引领内核创新

云原生调度增强：针对云场景在线和离线业务混合部署场景，创新 CPU 调度算法保障在线业务对 CPU 的实时抢占及抖动抑制，创新业务优先级 OOM 内存回收算法保障在线业务安全可靠运行。

- **新文件系统 EulerFS：**面向非易失性内存的新文件系统，采用软更新、目录双视图等技术减少文件元数据同步时间，提升文件读写性能。
- **内存分级扩展 etMem：**新增用户态 swap 功能，策略配置淘汰的冷内存交换到用户态存储，用户无感知，性能优于内核态 swap。

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。

探索场景创新

边缘计算：发布面向边缘计算场景的版本 openEuler 21.09 Edge，集成 KubeEdge+ 边云协同框架，具备边云应用统一管理和发放等基础能力。

- **嵌入式：**发布面向嵌入式领域的版本 openEuler 21.09 Embedded，镜像大小 <5M，启动时间 <5S。

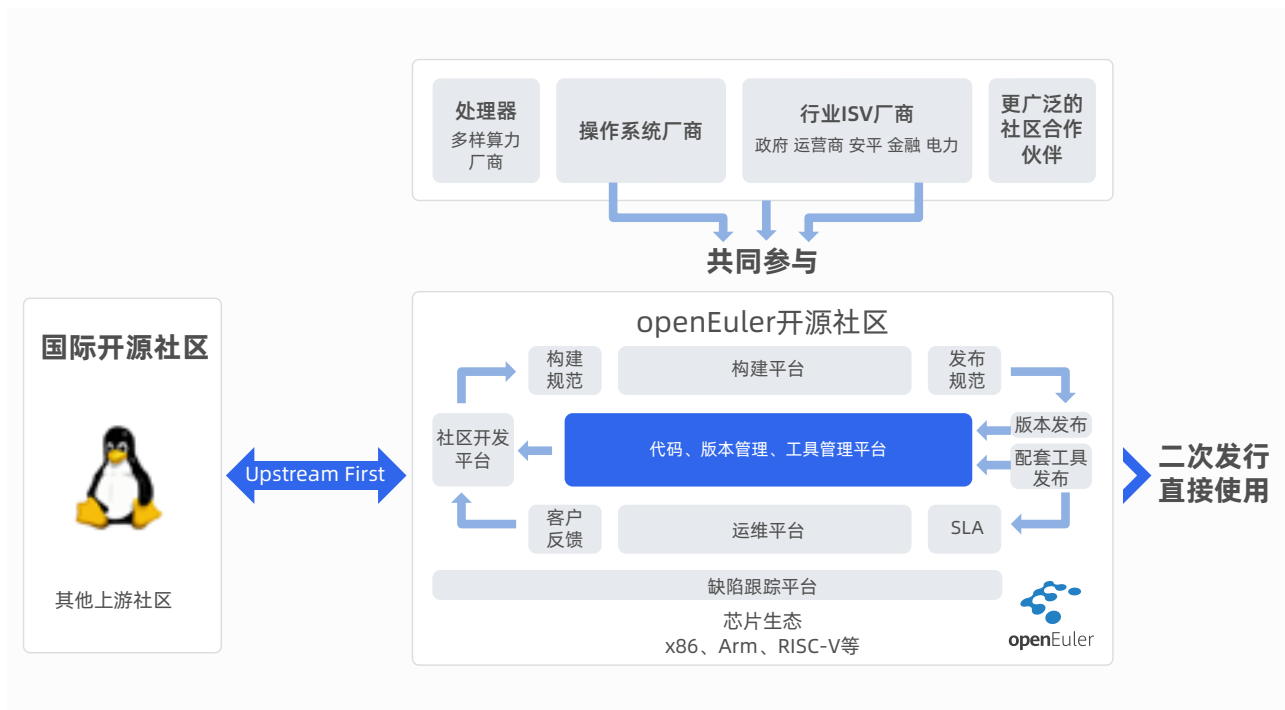
繁荣社区生态

友好桌面环境：UKUI、DDE、Xfce 桌面环境，丰富社区桌面环境生态。

- **欧拉 DevKit：**支持操作系统迁移、兼容性评估、简化安全配置 secPaver 等更多开发工具。

平台框架

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



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、ARM32
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，当前已支持的虚拟机为：

1. centos 7.9 qemu 1.5.3-175.el7 libvirt 5.0.0-1.el7
2. centos 8 qemu 5.1.0-20.el8 libvirt 6.6.0-7.3.el8
3. fedora 29 qemu 3.0.0-1.fc29 libvirt 4.7.0-1.fc29
4. fedora 32 qemu 4.2.0-7.fc32 libvirt 6.1.0-2.fc3

部件名称	最小硬件要求
架构	ARM64、ARM32
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、ARM32
内存	为了获得更好的体验，建议不小于 4GB
硬盘	为了获得更好的体验，建议不小于 20 GB

嵌入式

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

部件名称	最小硬件要求
架构	ARM64、ARM32
内存	为了获得更好的体验，建议不小于128MB
存储	为了获得更好的体验，建议不小于256MB

04/

内核创新



openEuler 内核中的新特性

openEuler 21.09 基于 Linux Kernel 5.10 内核构建，在进程调度、内存管理、网络等方面带来 12 处如下创新：

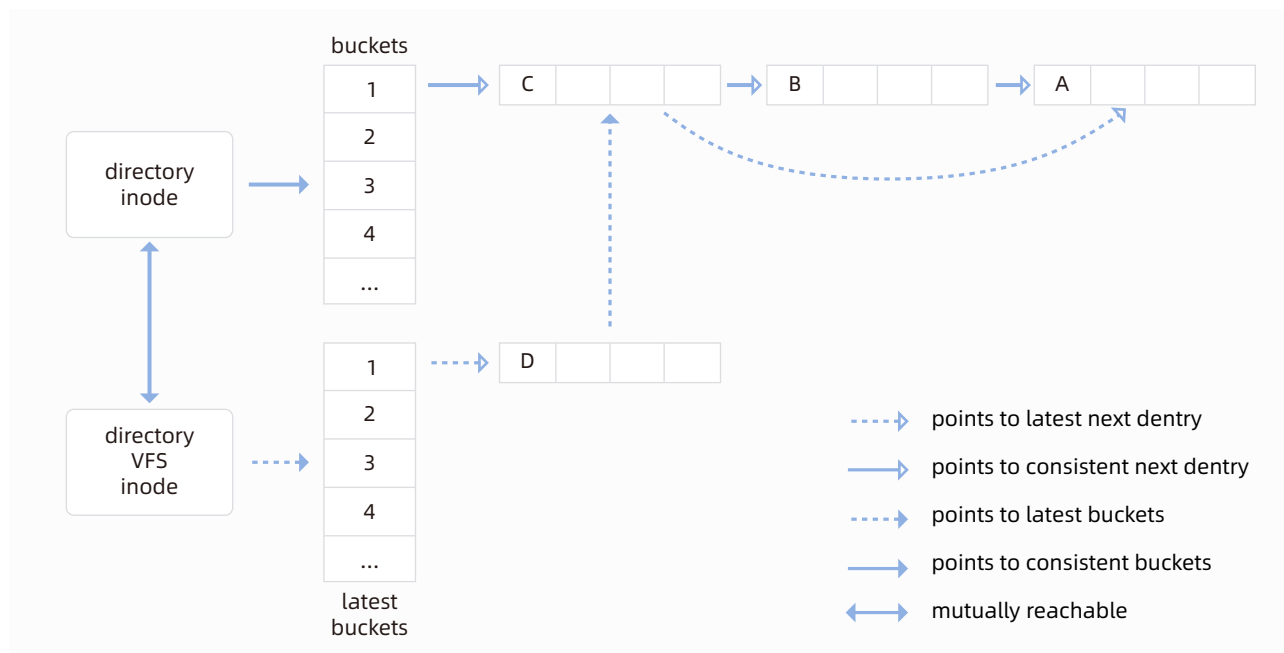
- 01/ 进程调度优化：**优化进程负载均衡算法，减少负载均衡过程中的开销，提升性能；
- 02/ 内核动态抢占：**新增启动选 preempt=none/voluntary/full，允许内核动态切换抢占模式。
- 03/ mremap 性能优化：**通过移动 PMD/PUD 级别的表项，加速映射大块内存的速度。
- 04/ per memcg lru lock：**采用 per memcg lru_lock，减少云原生容器实例锁竞争，提升系统性能。
- 05/ 大页内存管理优化：**通过共享映射方式将 HugeTLB 管理页中无实际作用的 tail 页释放掉，降低大页内存管理结构的开销。
- 06/ TLB 并发刷新支持：**本地 TLB 和远端 TLB 刷新并行，优化 TLB shutdown 流程加速 TLB 刷新，提升业务性能。
- 07/ 大页 vmalloc 性能优化：**对于超过 huge page 的最小 size 的空间进行 vmalloc() 分配时，将会尝试使用 huge page 而不是 base page 来映射内存，改善 TLB 的利用，降低 TLB miss。
- 08/ OOM 内存回收算法：**在发生 OOM 时，优先对低优先级的进程组进行内存回收，保障在线业务的正常运行。
- 09/ 支持 PAC (Pointer Authentication Code) 特性：**在使用寄存器的值作为指针访问数据或代码之前验证其内容，抵御 ROP/JOP 攻击。
- 10/ 支持 BTI (Branch Target Identifiers) 特性：**对间接跳转的目标进行限制。与 PA 结合使用减少控制流攻击。
- 11/ XDP (eXpress Data Path) 支持：**基于 eBPF 的一种高性能、用户可编程的网络数据包传输路径，在网络报文还未进入网络协议栈之前就对数据进行处理，提升网络性能。可用于 DDOS 防御、防火墙、网络 QOS 等场景。
- 12/ SVA (Shared Virtual Addressing) 支持：**进程虚拟地址在主机进程和设备间共享，实现资源跨主机与设备免拷贝复用，提升跨主机和设备业务通讯性能。

新介质文件系统

非易失性内存（NVDIMM，比如 Intel Optane）是一种提供字节访问粒度的新型高速存储介质，现有内核文件系统 EXT4，可以协同 DAX 特性改善 NVDIMM 新介质数据读写性能，但在元数据管理方面，基于现有 journal 同步机制，元数据管理开销大，且容易出现写放大问题，NVDIMM 优势无法充分发挥。

EulerFS 创新元数据软更新技术（Soft Update），基于指针的目录双视图计数机制，减少元数据同步开销，有效提升文件系统 create、unlink、mkdir、rmdir 系统调用性能，较 ext4/dax，元数据操作延时降低 1~4 倍，带宽高 20%~4 倍。

功能描述



1. **哈希表目录：**采用哈希表来管理目录项，提高线性查找效率，减少伪共享。
2. **统一的分配器：**数据结构使用统一的分配器，这样可以打破不同数据结构之间的界限，使得内存管理更加地简单与灵活。
3. **采用软更新技术，简化了实现复杂度：**软更新 (Soft Update) 是一种轻量级的保证文件系统一致性的技术。
4. **基于指针的目录双视图计数机制，**减少元数据同步开销，有效提升文件系统读写性能。
5. **依赖跟踪：**目录项的新建、删除等操作并不是立刻持久化的，在进行相应的操作后，只是在 inode 中跟踪依赖的持久化操作，后续通过异步的方式进行持久化，可以大幅提高性。

应用场景

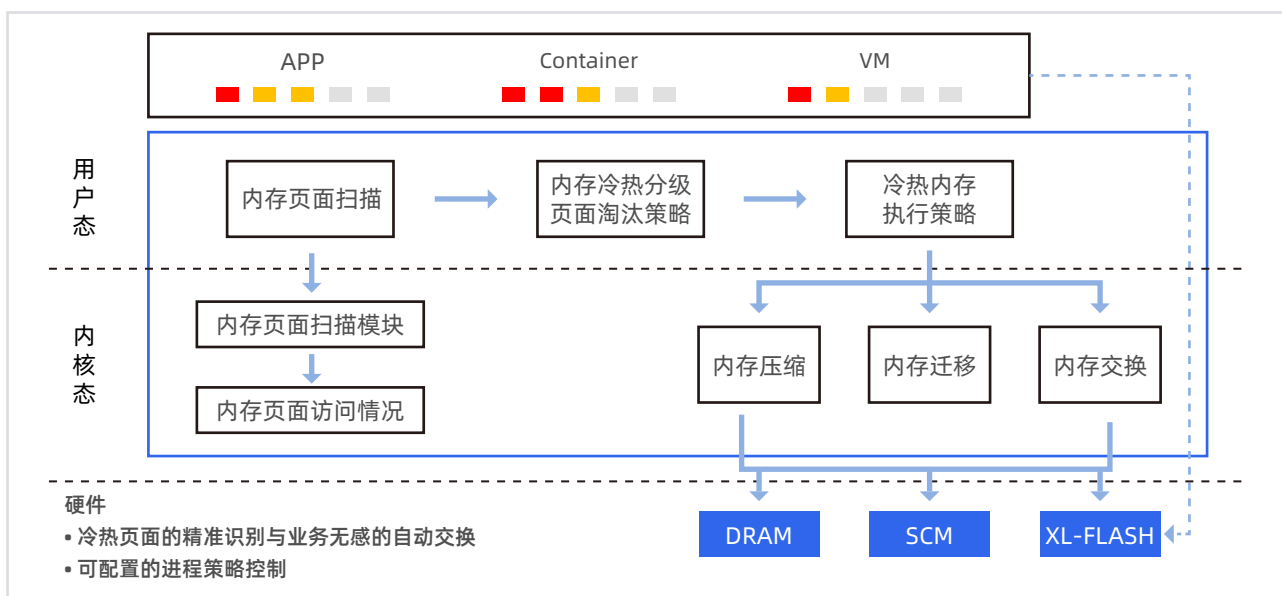
通用的针对 NVDIMM 新介质的文件系统，可代替 EXT4，XFS 等文件系统，满足单机应用、云原生分布式应用高性能数据存储诉求。

内存分级扩展

当前内存制造工艺已经达到瓶颈，Arm 生态发展让每个 CPU 核的成本越来越低。数据库、虚拟机、大数据、人工智能、深度学习场景同时需要算力和内存的支持。内存容量成为了制约业务和算力的问题。

内存分扩展通过 DRAM 和低速内存介质，如 SCM、AEP，以及 RDMA 远端内存等形成多级内存，通过内存自动调度让热数据在 DRAM 高速内存区中运行，让冷数据交换到低速内存区，从而增加内存容量，保证核心业务高效平稳运行。该特性适用于内存使用量大，且使用相对不频繁的应用进程上，在这些场景中的效果好收益大，实测等成本条件下 MySQL 性能提升 40%。针对用户态存储框架和用户需求，新增用户态的内存交换机制。

功能描述



继承已有功能：

1. **进程级控制：**etMem 支持通过配置文件来进行内存扩展的进程，相比于操作系统原生的基于 LRU 淘汰的 kswap 机制，更加灵活和精准。
2. **冷热分级：**用户态触发对指定进程进行内存访问扫描，根据分级策略配置文件，对内存访问结果进行分级，区分出热内存和冷内存。
3. **淘汰策略：**根据配置文件和系统环境配置，对冷内存进行淘汰，淘汰流程使用内核原生能力，安全可靠，用户无感知。

新增功能：

1. **进程级控制：**etMem 支持通过配置文件来进行内存扩展的进程，相比于操作系统原生的基于 LRU 淘汰的 kswap 机制，更加灵活和精准。

应用场景

应用场景：节点内业务进程内存分级扩展

适用于对内存使用较多，且访问相对不频繁的业务软件，扩展效果较好，比如 MySQL、Redis、Nginx 等，内存扩展操作均在节点内部，不涉及跨节点远端操作。

在用户态存储框架的场景中，可通过策略框架的用户态 userswap 功能，使用用户态存储作为交换设备。

05/

云化基座



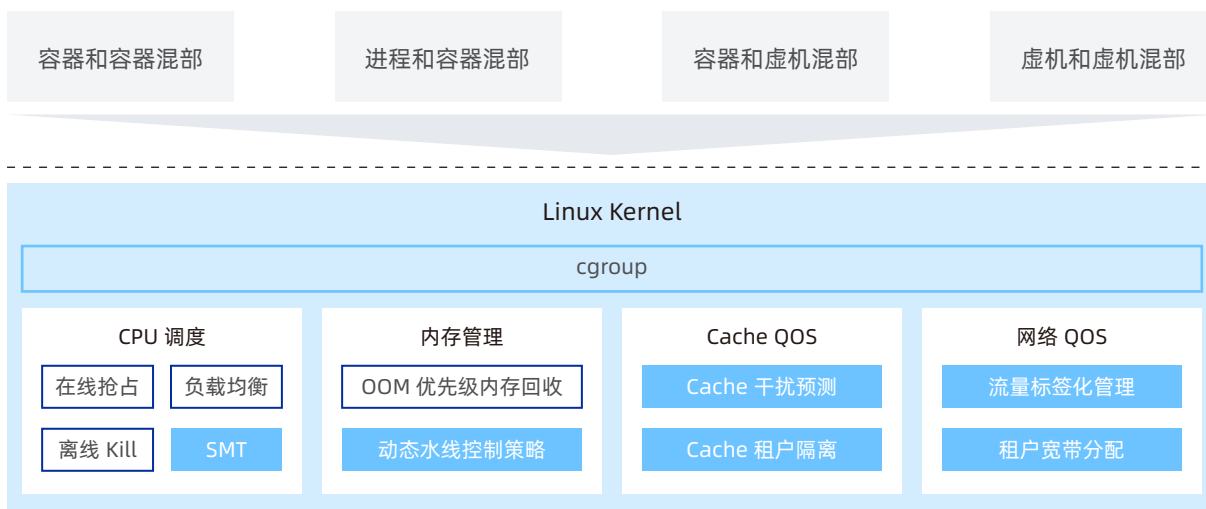
云原生调度增强

在云业务场景中，交互类延时敏感在线业务存在潮汐现象，CPU 资源利用率普遍较低 (~15%)，在线和离线业务混合部署是提升资源利用率的有效方式。在现有的内核资源分配和管理机制，混部后的在线业务的性能抖动大，服务质量无法得到有效保障。openEuler 面向云原生业务混部场景，创新 CPU 调度算法和内存回收算法，支撑提升系统的 CPU 利用率和保证在线业务的服务质量。

QAS (Quality aware scheduler) 是一种适用于云原生场景，业务混合部署的全新调度算法，可以确保在线任务对 CPU 的快速抢占，确定性的调度运行，同时压制离线任务干扰。

优化 OOM 时内存回收调度算法，在发生 OOM 时，优先对低优先级的进程组进行内存回收，保障在线业务的正常运行。

功能描述



1. **进程属性设置**：支持通过 cgroup 接口来设置任务运行级别，可以配置成在线或离线属性。
2. **任务优先运行**：在 cpu 中有在线任务和离线任务场景下，优先选择在线任务，在线任务可以 us 级抢占离线任务运行。
3. **离线 kill 机制**：任务收到 kill 信号后会快速响应 kill 信号处理流程，加快离线任务的退出流程。
4. **OOM 内存回收优先级机制**：在发生 OOM 时，优先对低优先级的进程组进行内存回收，保障在线业务的正常运行。

应用场景

适用于对交互类等时延敏感型业务（比如 MySQL、Redis、Nginx 等）和 CPU 消耗且时延不敏感的业务（如 AI 离线训练）混合部署，包括容器与容器、容器与进程、容器与虚拟机、虚拟机与虚拟机混合部署等多种场景。

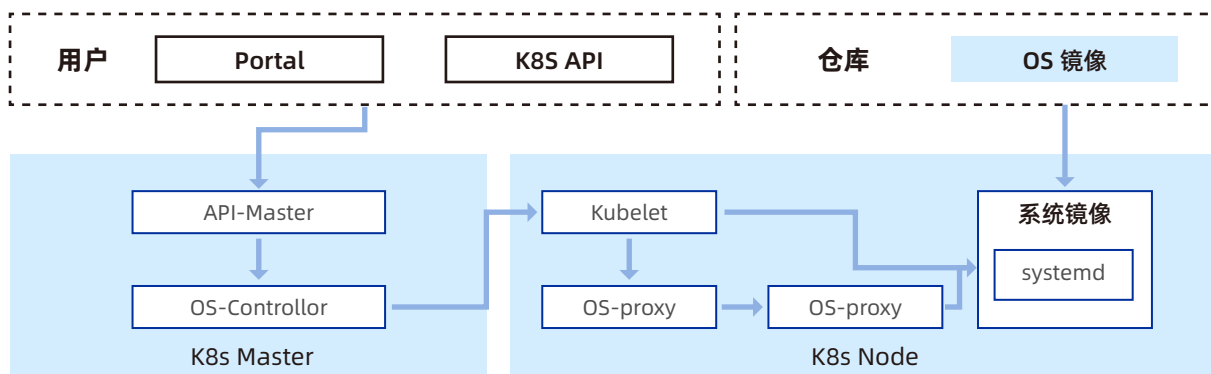
容器操作系统

云原生是云计算发展的下一跳，K8s 事实上已经成为云原生软件基础设施的底座。业界主流操作系统厂商都推出了针对云原生场景的 OS，如 Red Hat RCHOS，AWS BottleRocket 等，实现 OS 容器化部署、运维，提供与业务容器一致的管理和运维体验。

openEuler 适应云原生发展趋势，推出容器化操作系统 KubeOS，实现云原生集群 OS 的统一容器化管理，具备如下特点：

1. OS 容器化管理、对接 K8S，原子化的生命周期管理；
2. OS 轻量化裁剪，减少不必要的冗余包，可实现快速升级、替换等。

功能描述



1. 全局管理器 OS-Controller，监控所有节点上的 OS 实例，收集所有节点 OS 信息，实现全局 OS 的生命周期管理，包括升级、重启、配置，以及关机时驱逐 Pod。OS 轻量化裁剪，减少不必要的冗余包，可实现快速升级、替换等。
2. 节点代理 OS-proxy，部署和运行在工作节点上，监控单节点的 OS 实例，收集单节点 OS 信息，并上报到 OS-agent。
3. 节点 OS-agent，接受 os-proxy 相关命令，容器 OS 生命周期管理的执行单元，维护容器 OS 安装、升级等。

适用场景

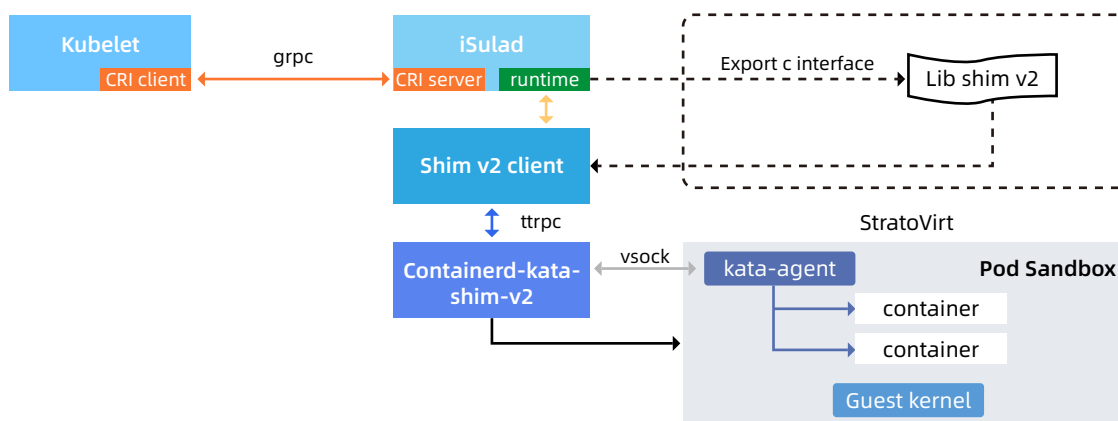
基于 K8S 容器云平台的业务节点服务器主机 OS 的容器化管理，提供容器业务相同的生命周期管理和运维体验；

安全容器方案

容器作为云计算广泛使用的技术，在简化应用封装，加快应用部署、减少环境依赖方面优势明显。容器可以直接访问主机资源，容器间共享主机内核，存在容器逃逸等诸多安全问题，无法满足典型金融多租户安全隔离诉求。业界阿里和 Intel 主导 kata 开源项目，使用虚拟化隔离层容器形成安全容器方案，Google 推出 GVisor 安全沙箱，结合进程级虚拟化隔离容器应用安全风险。

openEuler 结合虚拟化运行时 StratoVirt，容器管理引擎 isulad 形成安全容器方案，较传统 docker+qemu 方案，底噪和启动时间优化 40%+，为应用提供一个轻量、安全的执行环境，隔离容器和宿主机操作系统间、容器间的安全风险。

功能描述



安全容器 StratoVirt:

1. **强安全性**: 基于 Rust 实现语言级安全, 模块按需组合最小化攻击面, 支持多租户物理隔离。
2. **轻量低噪**: 采用极简设备模型时, 启动时间小于 50ms, 内存底噪小于 4M。
3. **软硬协同**: 支持 x86 的 VT, 支持鲲鹏的 Kunpeng-V。
4. **极速伸缩**: 毫秒级设备扩缩能力, 为轻量化负载提供灵活的资源伸缩能力。
5. **多场景支持**: 实现一套架构支持 serverless、安全容器、标准虚拟机等多种应用场景。

容器引擎 iSulad:

1. **轻量引擎**: C/C++ 编程语言重构轻量容器底座, 适应边、云多样场景;
2. **混合调度**: 通过 containerd-shim-kata-v2 对接 StratoVirt, 安全容器 StratoVirt 和普通容器 K8S 统一部署和调度。

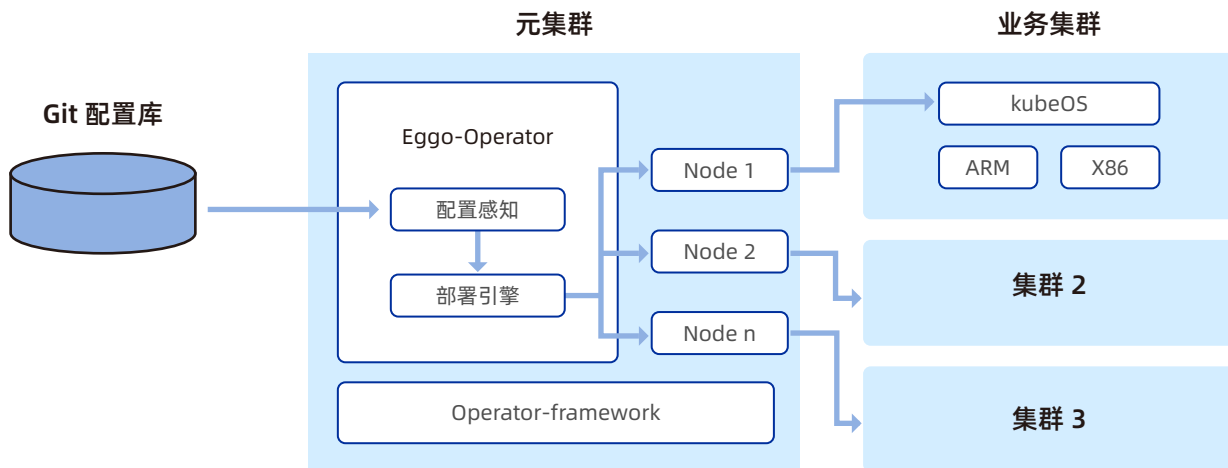
应用场景

替代 Docker 普通容器, 保证容器业务安全和隔离性, 可应用于金融、电信、安全等行业及面向多租户的业务场景;

双平面部署工具

eggo 是 openEuler 云原生 Sig 组 K8S 集群部署管理项目, 提供高效稳定的集群部署集群的能力。支持单集群多架构、支持在线和离线部署模式多种部署模式, 结合 GitOps 管理能力、感知集群配置变化, 驱动集群 OS 统一高效部署。

功能描述



1. **集群配置版本化管理**: 配置统一 Git repo 版本化管理, 使用仓库汇总和跟踪集群的配置信息;
2. **配置感知**: GitOps 会感知 git 配置库中集群配置信息的变化, 给部署引擎发起集群相应的操作请求;
3. **部署引擎**: 部署引擎负责下发任务给业务集群, 触发部署业务集群、销毁业务集群、新增节点和删除节点等任务。

适用场景

X86/ARM 双平面云基础设施, 基于 K8S 云原生框架, 实现 OS 统一集群化部署、监控、审计等场景。

06/ 场景创新



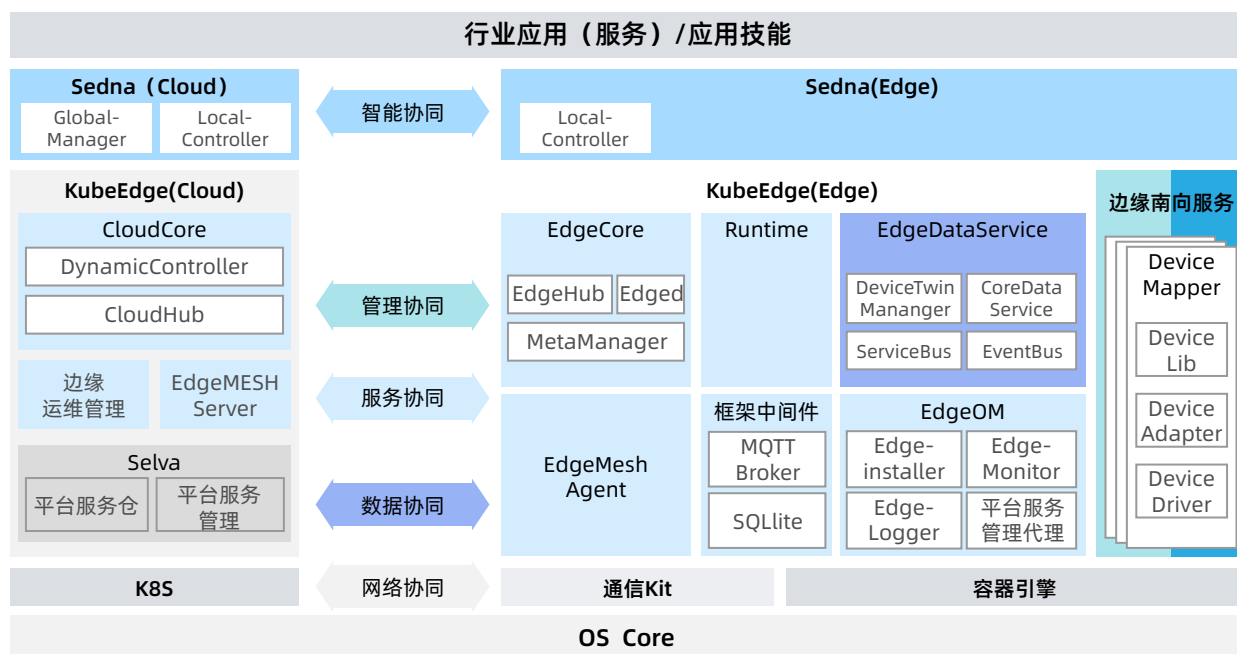
边缘计算和嵌入式是 openEuler 场景创新的第一次尝试，提供预览性质的技术创新体验，欢迎使用并反馈您的建议，帮助我们改进。

边缘计算

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

openEuler 发布面向边缘计算的版本 openEuler 21.09 Edge，集成 KubeEdge+ 边云协同框架，具备边云应用统一管理和发放等基础能力，并将通过增强智能协同提升 AI 易用性和场景适应性，增强服务协同实现跨边云服务发现和流量转发，以及增强数据协同提升南向服务能力。

功能描述



版本功能如下：

1. 提供统一的跨边云的协同框架（KubeEdge+），实现边云之间的应用管理与部署，跨边云的通信，以及跨边云的外设管理等基础能力。配置感知：GitOps 会感知 git 配置库中集群配置信息的变化，给部署引擎发起集群相应的操作请求。

未来还将提供：

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

应用场景

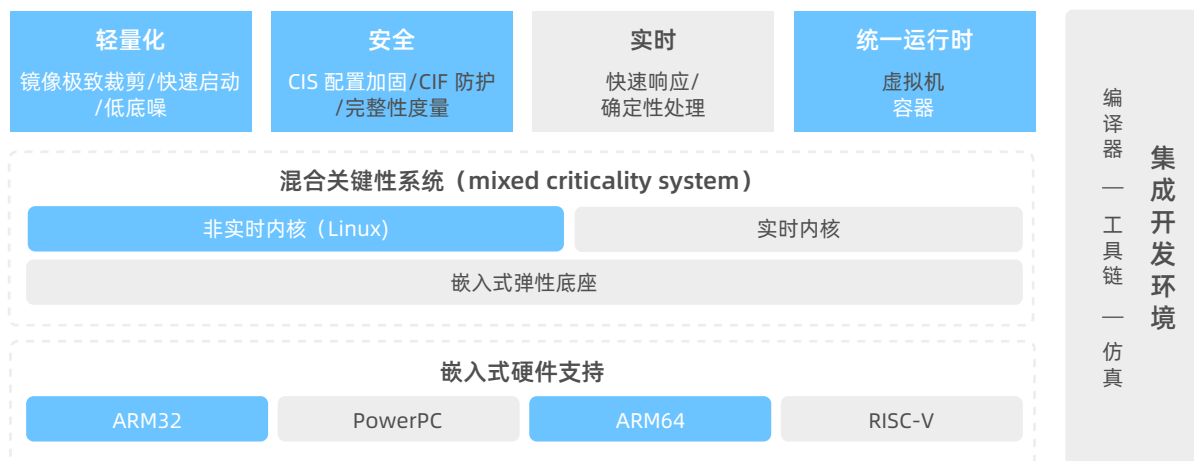
可应用智能制造、城市交通、高速收费稽查、智慧加油站、医疗影像识别、智慧园区等广泛的边云协同场景。

嵌入式

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

openEuler 发布面向嵌入式的版本 openEuler 21.09 Embedded，提供轻量化、安全和轻量容器等基础能力，支持 ARM32、ARM64 芯片架构。并将协同 openEuler 社区生态伙伴、用户、开发者，逐步扩展支持 PowerPC、RISC-V 等芯片架构，增加确定性时延、工业中间件、仿真系统等能力，打造嵌入式领域操作系统解决方案。

功能描述



版本功能如下：

- 轻量化能力：**开放 yocto 编译框架，支撑 OS 镜像轻量化定制，提供 OS 镜像 <5M, 运行底噪 <15 M, 以及 <5S 快速启动等能力。
- 安全加固：**对账户口令、文件权限等资源安全加固，OS 默认安全使能。
- 轻量容器：**面向嵌入式场景的轻量容器运行时，支持标准容器镜像的部署运行。
- 多架构支持：**支持 ARM32 和 ARM64 芯片架构。

未来还将提供：

- 确定性时延：**基于分域多内核架构的确定性时延处理能力，满足工业控制领域多层次的时延诉求；
- 行业安全认证：**逐步支持面向行业安全认证，如面向 IEC61508、EC62443 等

应用场景

嵌入式系统可广泛应用于航空航天、工业控制、电信设备、汽车及医疗等领域；随着 5G、AI 新型技术的成熟，还可应用于物联网 IOT 设备，边缘智能计算设备等。

分布式内存（敬请期待）

海量数据的时效性处理推动了应用的规模化扩展，大规模集群分布式应用加剧了复杂性和性能挑战，凸显了现有计算架构的瓶颈。分布式内存套件是集群分布式应用加速平台，针对大数据、HPC、AI、分布式存储、数据库、云与虚拟化等分布式应用场景，通过内存数据处理、近数计算等关键技术实现应用性能倍数级提升。分布式内存套件在多样化算力和端边云协同场景也可以实现应用的加速。该特性将在后续版本推出，敬请期待。



07/

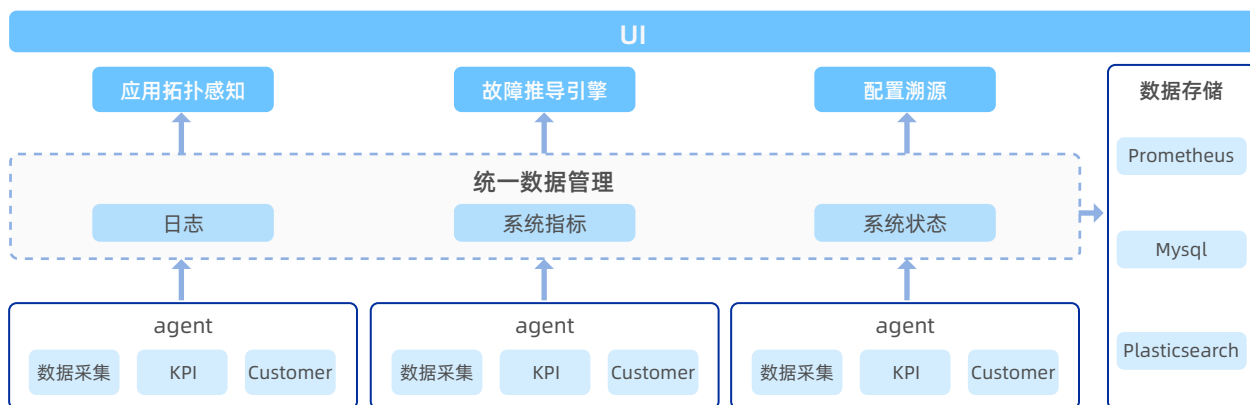
增强特性

智能运维 A-OPS

IT 基础设施和应用产生的数据量快速增长（每年增长 2~3 倍），应用大数据和机器学习技术日趋成熟，驱动高效智能运维系统产生，助力企业降本增效。业界 Redhat Insights 以数据驱动结合智能引擎实现故障自动预测、诊断及定位，将操作系统安全、性能等疑难问题解决周期，从数小时降到几分钟。

openEuler 智能运维推出智能运维基本框架，提供配置溯源，应用拓扑感知，故障定位基础能力，支持快速排障和运维成本降低。

功能描述



1. **应用拓扑感知**：是基于 eBPF 的低负载探针框架，提供应用级别的网络拓扑自动感知和检测能力，通过 web ui 提供网络状态可视化，便于运维人员快速发现网络问题
2. **配置溯源**：配置变更是维护阶段变动最多也最容易引入问题的操作，提供集群式 OS 配置管理能力，通过用户自定义域范围，实现配置基线和比较功能，快速排除配置问题。
3. **故障定位**：提供专家模式引擎，对系统故障实时感知，及时修复系统故障，减少宕机时间和运维投入。

应用场景

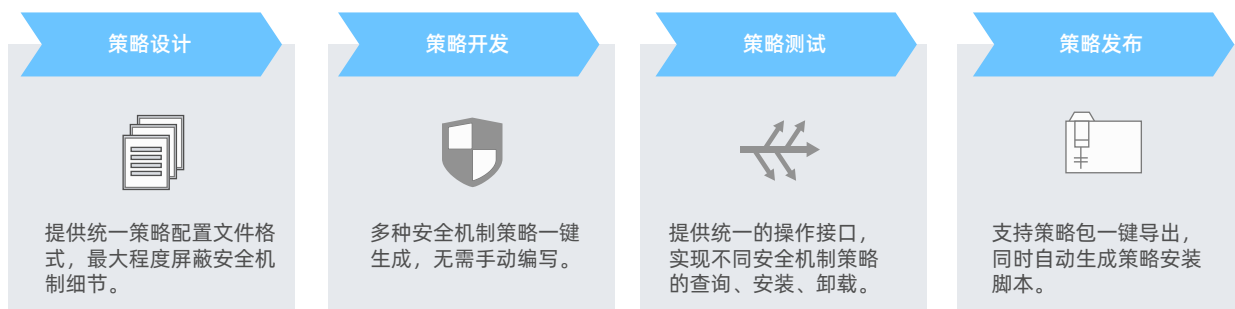
适用于具备较多经验的 OS 维护团队，采用 A-OPS 现有故障树或者新增故障树，可有效提升维护效率，减少宕机时间。

安全策略配置工具

secPaver 是一款 SELinux 安全策略开发工具，用于辅助开发人员为应用程序开发安全策略。secPaver 通过抽象通用的策略配置文件，封装统一的策略操作接口，对策略开发人员屏蔽安全机制细节，降低了策略开发门槛，提高策略开发效率。secPaver 当前支持 SELinux 的策略开发，后续可扩展支持 AppArmor 等其它安全机制。

功能描述

secPaver 功能定位：端到端的策略开发工具



主要功能包括：

1. **策略设计：**提供命令创建工程，在工程下编辑策略配置文件完成策略设计

```
pav project create project_name /opt/policy
```

2. **策略生成：**提供命令根据策略配置文件内容生成 SELinux 策略文件

3. **策略测试：**提供命令完成策略的安装、查询、卸载；

```
pav policy list
```

```
pav policy install/uninstall policy_name
```

4. **策略发布：**提供命令生成策略包（策略文件及安装卸载脚本文件）

```
pav policy export policy_name
```

应用场景

开发人员、安全策略配置人员使用 secPaver 为应用程序开发 SELinux 策略文件。

Kunpeng GCC

Kunpeng GCC 编译器基于开源 GCC 10.3 版本开发，通过软硬件协同，内存优化、SVE 向量化、数学库等，打造鲲鹏 920 处理器的高性能编译器：

1. 持 ARM 架构下 mcmodel=medium 的寻址方式，可以正确访问体积超过 4GB 的符号，解决程序地址溢出导致的错误。
2. 支持 ARM 架构下 128 位四精度浮点数库，有效提升浮点数运算的精度。
3. 支持矢量化 mathlib 数学库使能，在矢量化阶段自动寻找可以使用矢量化版本数学库的机会。
4. 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。
5. 支持 SLP 矢量化优化，进行 reduction chains group 的分析以及矢量化，提升程序运行的性能。
6. 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。
7. 支持 fp-model 精度控制选项，精细化控制浮点数计算精度。

应用场景

在高性能计算领域的测试中，运行 WRF、NEMO 应用，相比于上游社区的 GCC 10.3 版本能够获得 10% 左右的性能收益。还可用于 SPEC CPU 2017 基准测试中，相比于上游社区的 GCC 10.3 版本在鲲鹏平台能够获得 15% 左右的性能收益。

毕昇 JDK

毕昇 JDK 是基于 OpenJDK 开发的增强版本, 具备高性能、高可用等优点, 可用于生产环境。毕昇 JDK 积累了大量使用场景, 并针对 Arm 进行性能优化。支持 OpenJDK8 和 OpenJDK 11 两个版本, 其中毕昇 JDK 8 与 Java SE 标准兼容, 毕昇 JDK 具备如下优势:

1. **稳定高效:** 在 SPECjbb2015 等基准测试中性能优于 OpenJDK。
2. **软硬结合:** 充分发挥鲲鹏服务器硬件特性, 运行效率更高。
3. **安全有保障:** 毕昇 JDK 和 OpenJDK 社区版本同步更新, 同时针对安全问题做更为严格的分析和把控, 及时合入 CVE 安全漏洞补丁。
4. **开源贡献:** 毕昇 JDK 提供发行包和开源代码, 并持续贡献上游社区。

功能描述

1. **支持 Java Flight Recorder:** 低负载的在线性能诊断工具, 对于长时间运行的应用程序, 性能损耗小于 1%, 生产环境中可使能, 通过 `-XX:+FlightRecorder` 命令启用, 生成 dump 文件, 并可借助 JMC 工具可视化。
2. **支持 AppCDS 技术 (毕昇 JDK 8):** 该技术是 CDS 的扩展版本, 可以将应用类转储为 jsa 文件, 增加类使用范围, 提升应用启动加载速度。
3. **支持 G1GC 内存伸缩特性 (毕昇 JDK 8):** JVM 能够检测应用负载下降和 Java 堆有空闲内存的情况, 并自动减少 Java 堆占用情况, 将空闲内存资源归还给操作系统, 在按资源使用量付费的容器场景中可以节省开销。
4. **支持 ZGC 垃圾回收算法 (毕昇 JDK 11):** 在 ARM64 上补齐该功能。支持 TB 级大小, 最大 10ms 时延, 停顿时间不会随着堆增大而增大。
5. **G1GC 支持 NUMA-Aware 特性 (毕昇 JDK8 和 毕昇 JDK11):** JVM 能充分发挥硬件特性, 在应用运行进行对象分配会优先选择使用本地节点的内存, 在垃圾回收进行内存复制时优先在同一节点进行, 保证垃圾回收后应用的数据亲和性。
6. **支持新的快速序列化 (毕昇 JDK8 和 毕昇 JDK11):** 减少冗余新的传输, 提高序列化和反序化的效率。
7. **支持 KAE Provider 特性 (毕昇 JDK8):** 毕昇 JDK8 通过利用 Provider 机制, 实现对鲲鹏服务器 KAE 加解密特性的支持, 以帮助用户提升在鲲鹏 AArch64 服务器加解密业务的竞争力, 较 JDK 原生加解密性能提升 90%。
8. **支持 JMap 并行扫描 (毕昇 JDK8 和 毕昇 JDK11):** OpenJDK JMap 工具默认是单线程执行 java 程序的 dump 操作, 毕昇 JDK8 & JDK11 实现其并行化、增量式扫描, 有效提升 JMap 在大堆场景的扫描速度, 已支持 G1GC & ParallelGC & CMS。

应用场景

应用场景 1: 大数据应用

针对大数据优化 GC 内存的分配和回收, 消除 JIT 代码中冗余内存屏障等。在大数据应用的基准测试性能较 OpenJDK 有 5%~20% 的性能提升。

应用场景 2: 常见 Java 应用

针对鲲鹏服务器的弱内存模型进行优化, 无效内存屏障, 软硬结合使能 NUMA-Aware 特性, 提高应用访问内存的效率, 提升应用性能。增强 Java 开发者工具、引入 JFR 支持和 JMap 优化等功能帮助开发者快速分析性能、定位故障。



08/

社区治理

社区愿景

openEuler 社区的愿景是：通过社区合作，打造创新平台，构建支持多处理器架构、统一开放的操作系统社区，推动软硬件生态繁荣。

社区沟通和交流

openEuler 包含许多项目，这些项目被组织成社区团体。这些团体的沟通和交流渠道，包括邮件列表等，可以在这些团队的 README 上找到。

邮件列表

如果要开始一个开放主题的讨论，将电子邮件发送到相关的邮件列表是另一个不错的选择。

访问该地址 <https://openeuler.org/zh/community/mailling-list/> 找到社区可用的邮件列表，请根据您的兴趣参照下面步骤去加入某个邮件列表。

这里有两种加入方式：

网页

- ① 点击 <https://openeuler.org/zh/community/mailling-list/> 中的列表名字进入订阅页面。
- ① 输入订阅邮箱并点击“订阅”按钮。
- ① 登录邮箱并回复从 openeuler.org 发来的确认邮件。

如果接收到从 @openeuler.org 发来的类似“Welcome”的邮件就意味着您已经订阅成功了。

注意：如果没有收到“Welcome”邮件，请确保回复邮件时标题保持与原邮件一致再发一次。

邮箱

- ② 发送一封以“subscribe”为标题的邮件到每个邮件列表的提供的订阅地址（订阅地址是列表地址加“-join”后缀组成）。
- ③ 回复从 openeuler.org 发送的确认邮件。

以订阅 Dev（dev@openeuler.org）作为例子，邮件如下：

receivers: dev-join@openeuler.org

subject: subscribe

content: NA

当前有 Announce 和 Discussion 两种列表类型，对于 Discussion 列表的发送方式，跟发送一般的邮箱到私有地址没有区别，按照平常的方式发送即可，用一对中括号加一个主题作为前缀添加到邮件主题里会是一种很好的实践，但是不是必须的，Announce 类型的邮件列表只是用于宣布消息或者注意事项，不接受邮件发布。

注意：如果你不能在收件箱中接收到任何邮件信息，请优先检查是否将其归并到垃圾邮箱里了。

如果您想退订某个邮件列表，请参照如下步骤：

- ① 发送一封以“unsubscribe”为标题的邮件到该列表的退订地址（退订地址一般是列表地址加“-leave”后缀组成）。
- ② 直接回复确认邮件。

当您收到一封退订确认邮件表示您已经退订成功。

获取帮助

如果过程中遇到问题，请联系基础设施团队：

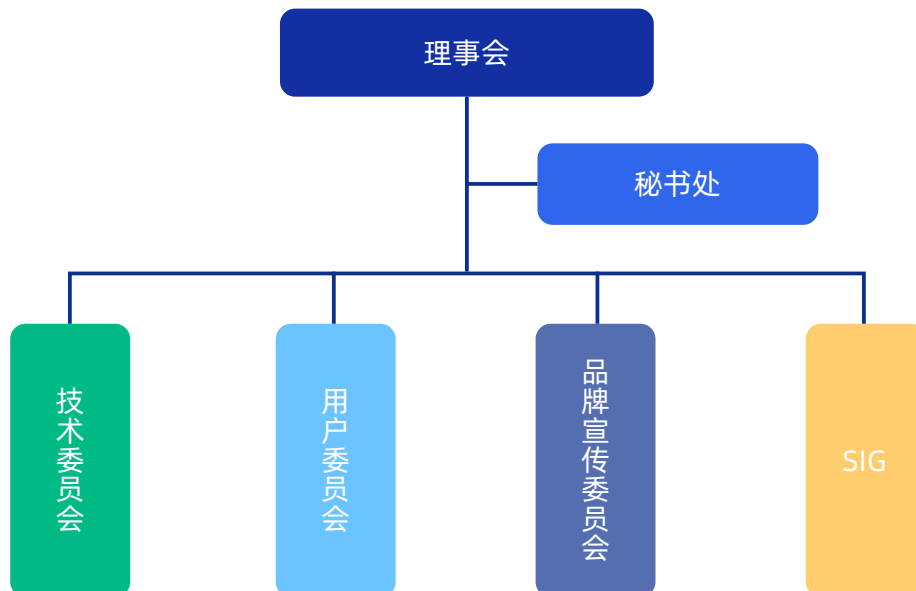
邮件：infra@openeuler.org

如果您发现任何有关邮件列表相关的 bug，请直接提交 issue 到 [infrastructure](#)

社区治理组织架构

社区治理组织架构包含理事会、秘书处、技术委员会、用户委员会、品牌宣传委员会、

特别兴趣小组 (Special Interest Group, 以下简称 SIG)：



理事会指导 openEuler 社区的发展方向，制定长期发展规划和实施指导意见；审视用户委员会、品牌宣传委员会的工作，对用户委员会、品牌宣传委员会的工作规划和内容进行决策。同时理事会面向全球各行业宣传和推广 openEuler 社区及基于社区技术的操作系统发行版，促进其广泛使用和生态建设。

秘书处

秘书处在 openEuler 理事会的指导下开展日常工作：

1. 执行理事会的各项决议，筹备和落实理事会会议召开，协调社区各个机构开展工作。
2. 编写社区季度、年度的工作报告，经理事会批准后对外发布。

技术委员会

openEuler 技术委员会（Technical Committee，以下和 TC 互换使用）是 openEuler 社区的技术领导机构。

技术委员会的主要职责如下：

1. 技术委员会拥有技术决策的最终裁决权。
2. 决策社区技术的发展愿景和方向。
3. 决策社区 SIG 的成立、撤销、合并等事务。解决 SIG 组之间的协作冲突，辅导、审视和监督 SIG 组的日常运作。
4. 落实社区日常开发工作，保证 openEuler 操作系统版本高质量发布。
5. 导社区在体系架构、内核、虚拟化、云原生、安全等领域技术创新，保证社区具有持续的技术竞争力。
6. 导社区建立原创性开源项目，持续构建社区技术影响力。

组织会议

公开的会议时间：当前 TC 在北京时间，每双周周三的上午 10:00 召开公开的例行线上讨论会议。

具体会议参会方式，会提前一天在邮件列表中讨论确定。

欢迎任何感兴趣的开发者参加旁听

SIG

专注于一个领域的持久和开放的团队，该团队通过定期的任务和活动实现特定的交付目标。SIG 具有公开透明的程序，要遵循 openEuler 的行为准则。任何人都可以参与并作出贡献。所有的 SIG 都存在于 <https://openeuler.org/zh/sig/sig-list/>。

用户委员会

用户委员会是社区的最终用户沟通机构，其主要职权如下：

- 收集对于 openEuler 社区版、基于 openEuler 社区技术的操作系统发行版、其它所属 openEuler 社区开源项目的技术和产品需求，反馈给技术委员会和理事会，推动技术委员会的技术路线符合最终用户的需求，为理事会的发展方向和长期规划提供市场意见。
- 整理 openEuler 社区版、基于 openEuler 社区技术的操作系统发行版、其它所属 openEuler 社区开源项目的用户案例，在获得最终用户授权的情况下配合品牌宣传委员会进行案例宣传。

品牌宣传委员会

品牌宣传委员会是社区的品牌宣传机构，其主要职权是：

- 推广 openEuler 操作系统技术和社区，提升 openEuler 品牌的影响力。
- 引导 openEuler 操作系统技术的广泛使用，构建 openEuler 全球生态。

贡献

做出贡献的第一步是从 openEuler 的 SIG/ 项目列表中选择。开始参加 SIG/ 项目会议，并订阅邮件列表。SIG/ 项目通常会由一系列 help-wanted 的 Issue，这些 Issue 可以帮助新的贡献者参与进来。

签署 CLA

您必须首先签署“贡献者许可协议”（CLA），然后才能参与社区贡献。

社区行为准则

openEuler 社区遵守开源社区《贡献者公约》V1.4 中规定的行为守则，请参考 V1.4 版本。

如需举报侮辱、骚扰或其他不可接受的行为，您可以发送邮件至 tc@openeuler.org，联系 openEuler 技术委员会处理。

贡献者们的承诺

为建设开放友好的环境，我们贡献者和维护者承诺：不论年龄、体型、身体健全与否、民族、经验水平、教育程度、社会地位、国籍、相貌、种族等，我们项目和社区的参与者皆不受此骚扰。

我们的准则

有助于创造积极环境的行为包括但不限于：

- 措辞友好且包容。
- 尊重不同的观点和经验。
- 耐心接受有益批评。
- 关注对社区最有利的事情。

与社区其他成员友善相处参与者不应采取的行为包括但不限于：

- 发布与色情、暴力等有关的言论或图像。
- 捣乱 / 煽动 / 造谣行为、侮辱 / 贬损的评论、人身及政治攻击。
- 公开或私下骚扰。
- 未经明确授权便发布他人的资料，如住址、电子邮箱等。• 其他有理由认定为违反职业操守的不当行为。

我们的义务

社区项目维护者（Maintainer）有义务诠释何谓“妥当行为”，并妥善公正地纠正已发生的不当行为。社区项目维护者有权利和义务去删除、编辑、拒绝违背本行为标准的评论（Comments）、提交（Commits）、代码、wiki 编辑、问题（Issues）等贡献；社区项目维护者可暂时或永久地封禁任何他们认为行为不当、威胁、冒犯、有害的参与者。

适用范围

本行为标准适用于本社区。当有人代表本社区时，本标准亦适用于此人所处的公共平台。

代表本社区的情形包括但不限于：使用社区的官方电子邮件、通过官方媒体账号发布消息、作为指定代表参与在线或线下活动等。

代表本社区的行为准则可由秘书处进一步定义及解释，并报理事会审批发布。

监督与调查

可以致信秘书处 secretary@openeuler.io，向项目团队举报滥用、骚扰及不当行为。

秘书处将审议并调查全部投诉，妥善地予以必要的回应。项目团队有义务保密举报者信息。

贯彻落实

可以致信 tc@openeuler.org，向项目团队举报滥用、骚扰及不当行为。

维护团队将审议并调查全部投诉，妥善地予以必要的回应。项目团队有义务保密举报者信息。具体执行方针或将另行发布。

未切实遵守或执行本行为标准的项目维护人员，经项目负责人或其他成员决议，可能被暂时或永久地剥夺参与本项目的资格。

社区贡献

在社区上总是有可以改进的文档（比如您正在阅读的），需要检视的代码，可以重构或注释的函数或变量，可以持续补充和优化的测试用例。我们将帮助您了解 openEuler SIG 的组织方式，并引导您顺利的开始您的第一个贡献。

了解 SIG

SIG 就是 Special Interest Group 的缩写，openEuler 社区按照不同的 SIG 来组织，以便于更好的管理和改善工作流程。

- SIG 组是开放的，欢迎任何人加入并参与贡献。
- SIG 都是针对特定的一个或多个技术主题而成立的。SIG 内的成员推动交付成果输出，并争取让交付成果成为 openEuler 社区发行的一部分。
- SIG 的核心成员主导 SIG 的治理。您可以在贡献的同时积累经验和提升影响力。
- 每一个 SIG 在 Gitee 上都会拥有一个或多个项目，这些项目会拥有一个或多个 Repository。SIG 的交付成果会保存在这些 Repository 内。
- 可以在 SIG 对应的 Repository 内提交 Issue、针对特定问题参与讨论，提交和解决问题，参与评审等。
- 您也可以通过邮件列表、视频会议和 SIG 内的成员进行交流。

找到您感兴趣的 SIG 组，可以帮助您在正确的地方提出问题，并得到更快的社区响应。

- 方式一：如果您不了解有哪些 SIG 或项目，您可以查看 SIG 列表，它包含当前 openEuler 社区成立的所有 SIG 团队的清单。您可以通过该列表快速的定位到您感兴趣的领域所对应 SIG 团队。同时还会为您提供该 SIG 团队的如下信息：
 - SIG 下的项目，以及项目的 Repository 地址
 - SIG 内的交流方式，包括邮件列表、视频会议等
 - Maintainer 的联系方式
- 方式二：如果您知道感兴趣的项目名称，可以在 openEuler 的 Repository 列表下进行模糊搜索，从而快速定位到对应项目的首页地址。通常情况下，在该项目首页地址的 README.md 文件中，可以找到该项目所属的 SIG 信息、交流方式、成员和联系方式等。

如果上述两种方式都定位不到您感兴趣的 SIG，您可以向 community@openeuler.org 发求助邮件。建议您在邮件列表内用“【开发过程疑问】”作为标题，在内容中写出你寻找的 SIG 或项目的特征，我们会为您提供帮助。

开始贡献

找一个 Issue

- 找到 Issue 列表：在您感兴趣的项目的首页内（Gitee 上的项目的 Repository）的工具栏，点击“Issues”，您可以找到该 SIG 的 Issue 列表（如 Community 团队的 Issue 列表地址为 <https://gitee.com/openeuler/community/issues>）。
- 找到愿意处理的 Issue：如果您愿意处理其中的一个 Issue，可以将它分配给自己。只需要在评论框内输入 `/assign` 或 `/assign @ yourself`，机器人就会将问题分配给您，您的名字将显示在负责人列表里。
- 参与 Issue 内的讨论：每个 Issue 下面可能已经有参与者们的交流和讨论，如果您感兴趣，也可以在评论框中发表自己的意见。

提出问题或建议

- 提出问题：如果您发现并向社区上报问题或缺陷，问题提交的方式就是创建一个 Issue。您只要将问题以 Issue 的方式提交到该项目 Repository 的 Issue 列表内。提交问题时，请尽量遵守问题提交准则。
- 提出建议：如果您想对 SIG 领域内贡献出自己的意见或建议，也可以通过提交 Issue 的方式分享给大家。大家可以在该 Issue 上充分的交流和讨论。为了吸引更多广泛的注意，您也可以把 Issue 的链接附在邮件内，通过邮件列表发送给所有人。

搭建开发环境（具体详情参见附录 1）

1. 安装 openEuler
2. 开发环境准备
3. 下载和构建软件包

参与编码贡献

- 了解 SIG 和项目内的开发注意事项

每个 SIG 内的项目使用的编码语言、开发环境、编码约定等都可能存在差异的。如果您想了解并参与到编码类贡献，可以先找到该项目给开发者提供的贡献者指南——这个指南一般是在该 SIG 的首页地址内，以 CONTRIBUTING.md 文件的形式提供，或者就直接在该项目的 README.md 内。除了这些文件外，SIG 可能还会提供其他指南信息。这些信息位于 SIG 或其项目的特定社区目录中。如果您未找到相关信息，或者对相关信息有疑问，可以在该 SIG 内提交 Issue，或者把问题或疑问发送到该项目所属 SIG 的邮件列表。如果您认为长时间没有得到回应，可以向 community@openeuler.org 求助。

- 下载代码和拉分支

如果要参与代码贡献，您还需要了解如何在 Gitee 下载代码，通过 PR 合入代码等。该托管平台的使用方法类似 GitHub，如果您以前使用 GitHub，本章的内容您可以大致了解甚至跳过。

- 修改、构建和本地验证

在本地分支上完成修改后，进行构建和本地验证。

- 提交一个 Pull-Request

当你提交一个 PR 的时候，就意味您已经开始给社区贡献代码了。

- 如何新增软件包

openEuler 支持在 gitee 新增软件包的同时自动在 obs 的 openEuler:Factory 上创建同名仓库。这样在向已创建的 gitee 仓库提交代码时，会自动对代码编译进行检测。

检视代码

openEuler 是一个开放的社区，我们希望所有参与社区的人都能成为活跃的检视者。

对于贡献者，为了使您的提交更容易被接受，您需要：

- 遵循 SIG 组的编码约定，如果有的话。
- 准备完善的提交信息。
- 如果一次提交的代码量较大，建议将大型的内容分解成一系列逻辑上较小的内容，分别进行提交会更便于检视者理解您的想法。
- 使用适当的 SIG 组和监视者标签去标记 PR：社区机器人会发送给您消息，以方便您更好的完成整个 PR 的过程。

对于检视者，强烈建议本着行为准则，超越自我，相互尊重和促进协作。《补丁审核的柔和艺术》一文中提出了一系列检视的重点，说明代码检视的活动也希望能够促进新的贡献者积极参与，而不会使贡献者一开始就被细微的错误淹没，所以检视的时候，可以重点关注包括：

- 贡献背后的想法是否合理。
- 贡献的架构是否正确。
- 贡献是否完善。

注意：如果您的 PR 请求没有引起足够的关注，可以在 SIG 的邮件列表或 dev@openeuler.org 求助。

选择社区组件打包

制作 RPM 包，俗称打包，是指编译并捆绑软件与元数据例如软件全名、描述、正常运行所需的依赖列表等等的任务。这是为了让软件使用者可以使用软件包管理器舒服的安装、删除或者升级他们所使用的软件。

打包规则：openEuler 试图规范化多种多样的开源项目到一个连贯的系统。因此 openEuler 制定此打包指导来规范制作 RPM 的动作。

- openEuler 遵守一般的 Linux 基础标准 (LSB)。该标准致力于减少各个发行版间的不同。openEuler 也遵守 Linux 文件系统层级标准 (FHS)。该标准是关于如何管理 Linux 文件系统层级的参考。
- 除了遵守这些一般 Linux 发行版都会遵守的一般规则，本文档规范化了为 openEuler 社区版打包的实际细节问题。

社区安全问题披露（安全处理流程和安全披露信息请参考附录 2）

- 安全处理流程
- 安全披露信息

09/ 著作权说明

openEuler 白皮书所载的所有材料或内容受版权法的保护，所有版权由 openEuler 社区拥有，但注明引用其他方的内容除外。未经 openEuler 社区或其他方事先书面许可，任何人不得将 openEuler 白皮书上的任何内容以任何方式进行复制、经销、翻印、传播、以超 级链路连接或传送、以镜像法载入其他服务器上、存储于信息检索系统或者其他任何商业目的的使用，但对于非商业目的的、用户使用的下载或打印（条件是不得修改，且须保留该材料中的版权说明或其他所有权的说明）除外。

10/ 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为以暗示、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。

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



Open Leads to More

开 创 新 境