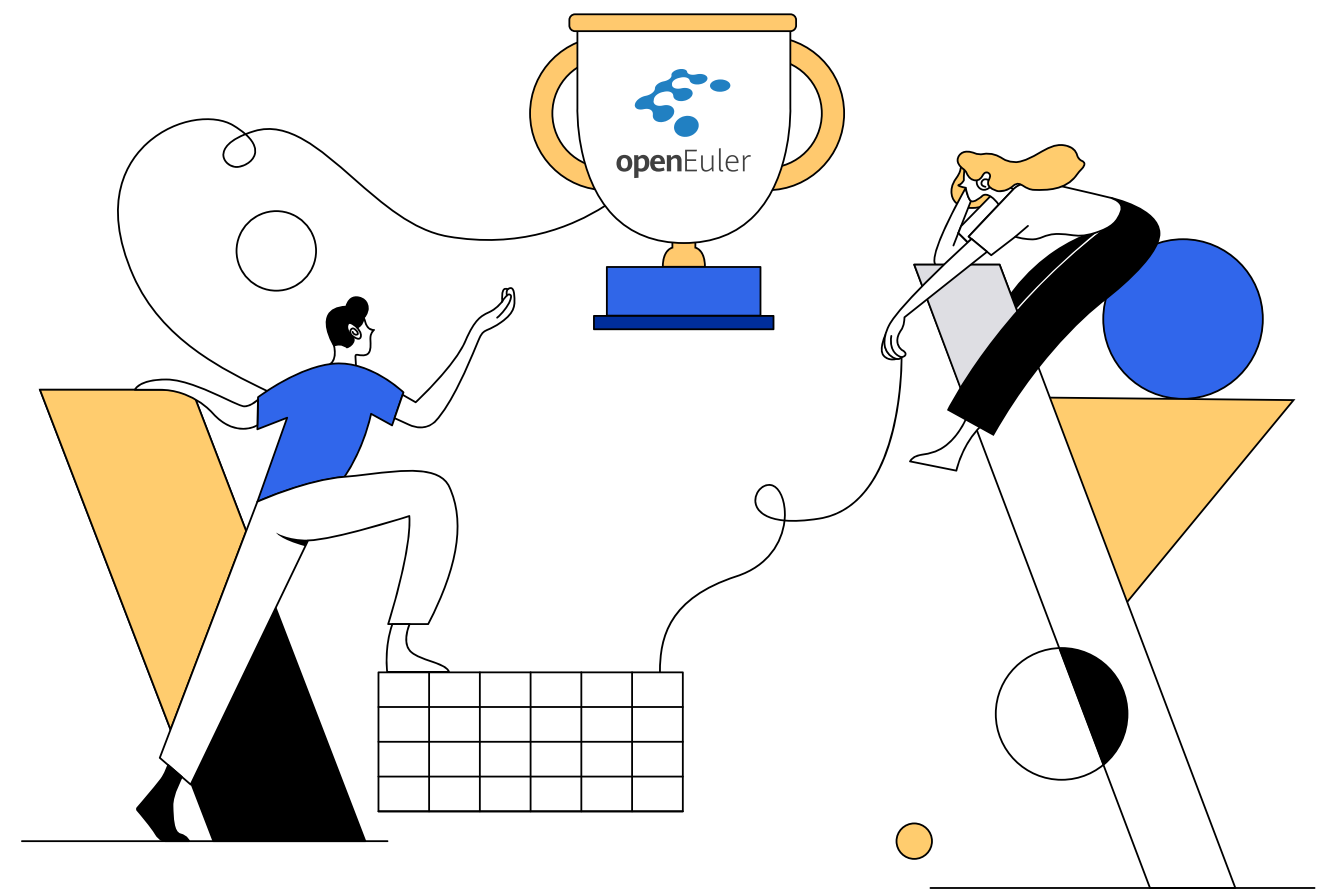


# openEuler 21.03

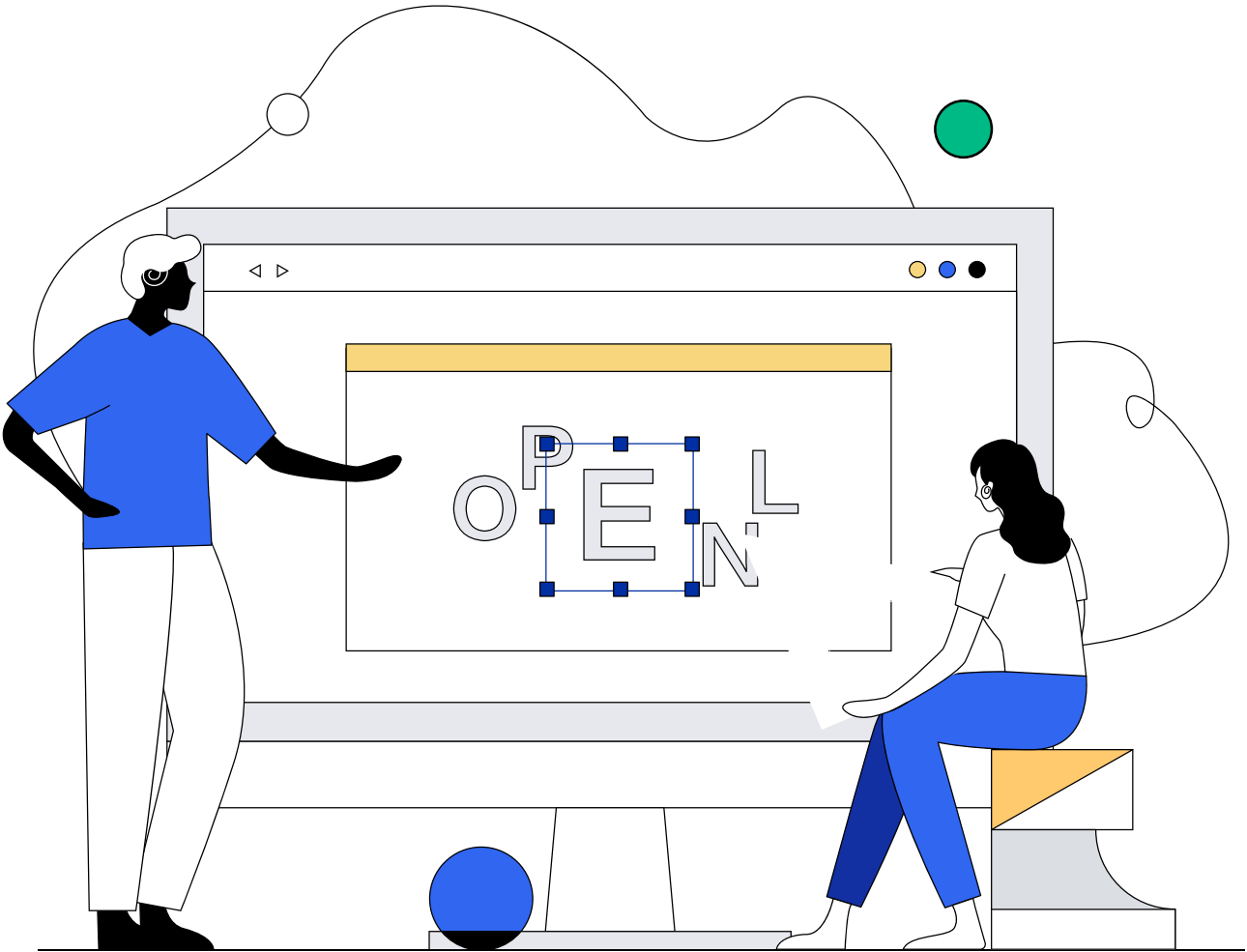
## 技术白皮书



# CONTENTS

- 01 / 概述 ..... 01
- 02 / 平台架构 ..... 04
- 03 / 运行环境 ..... 07
- 04 / 内核创新 ..... 09
- 05 / 云化基座 ..... 13
- 06 / 生态繁荣 ..... 17
- 07 / 增强特性 ..... 21
- 08 / 社区治理 ..... 25
- 09 / 著作权说明 ..... 33
- 10 / 商标 ..... 33
- 11 / 附录 ..... 33

概述



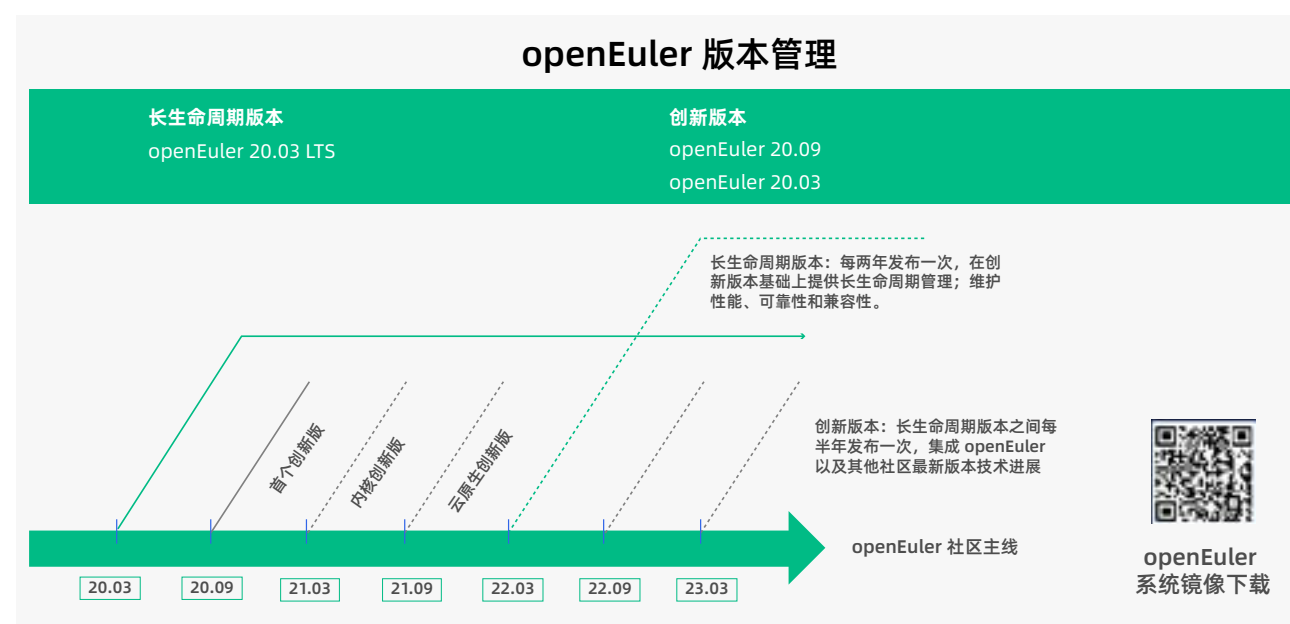
openEuler 是一个开源的 Linux 发行版平台。将通过开放的社区形式与全球的开发者共同构建一个开放、多元和架构包容的软件生态体系。openEuler 旨在通过社区合作，打造创新平台，构建支持多种处理器架构、统一和开放的操作系统社区，推动软硬件应用生态繁荣发展。

2019 年 12 月 31 日，一个面向多样性算力的操作系统开源社区 openEuler 正式成立。

2020 年 3 月 30 日，首个 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 创新版正式发布，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的事件。

2021 年 3 月 31 日，openEuler 21.03 创新版如期而至，该版本不仅将内核切换到 Linux Kernel 5.10，还在内核方向实现内核热升级、内存分层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。



openEuler 作为一个发行版平台，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。

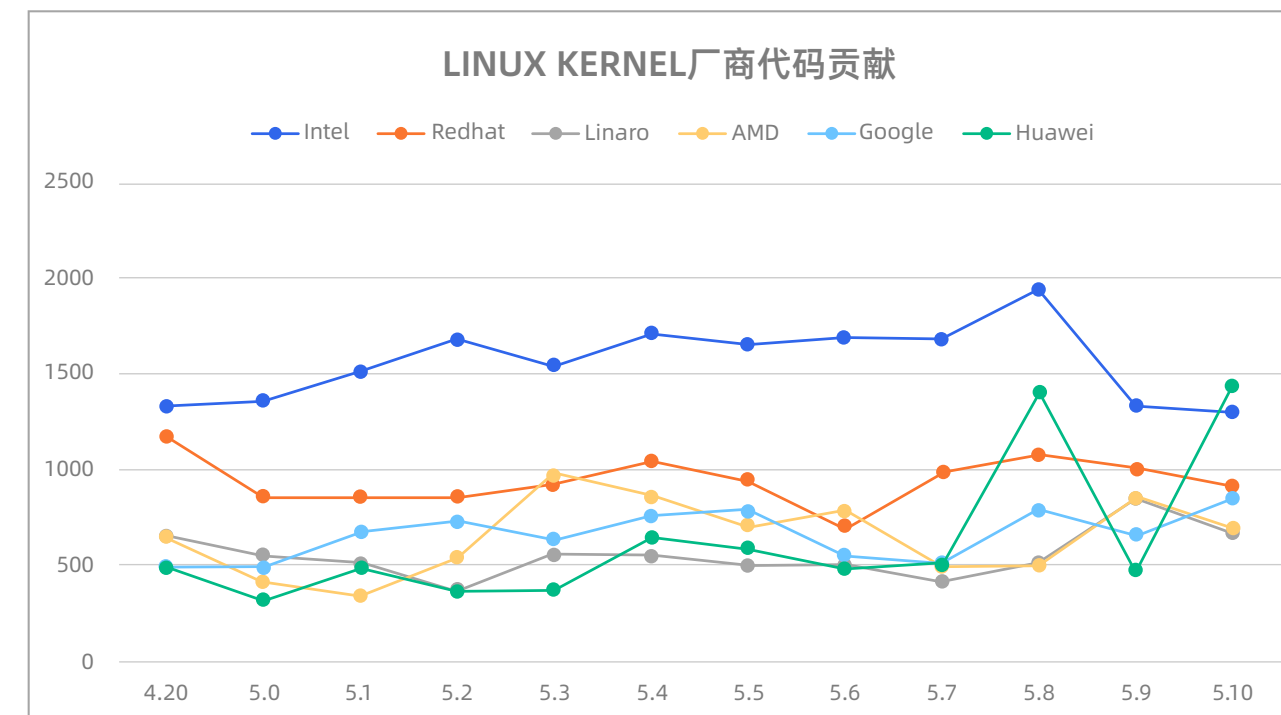
社区中的最新技术成果持续合入发行版，发行版通过用户反馈反哺技术，激发社区创新活力，从而不断孵化新技术。发行版平台和技术孵化器互相促进、互相推动、牵引版本持续演进。

## openEuler 面向多样性算力的创新

openEuler 支持 X86、Arm、RISC-V 等多处理器架构，是一个推动多样性计算架构不断创新的操作系统平台。openEuler 支持以 Intel/ 兆芯为代表的 X86 架构，以鲲鹏、飞腾为代表的 ARM 架构，同时发布了 RISC-V 的体验版本。openEuler 希望与广大用户和开发者一起加速完善多样化算力的生态和性能体验。

openEuler 希望把工作负载和最合适的算力单元进行匹配，通过软件优化提升并行处理能力，高效的释放多样性算力。

## openEuler 对 Linux Kernel 的持续贡献



华为作为 Linux Kernel 的主要贡献团队，对于内核的贡献主要集中在：ARM64 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 及代码重构的内容。十余年来总计向社区贡献 13000+ 补丁。

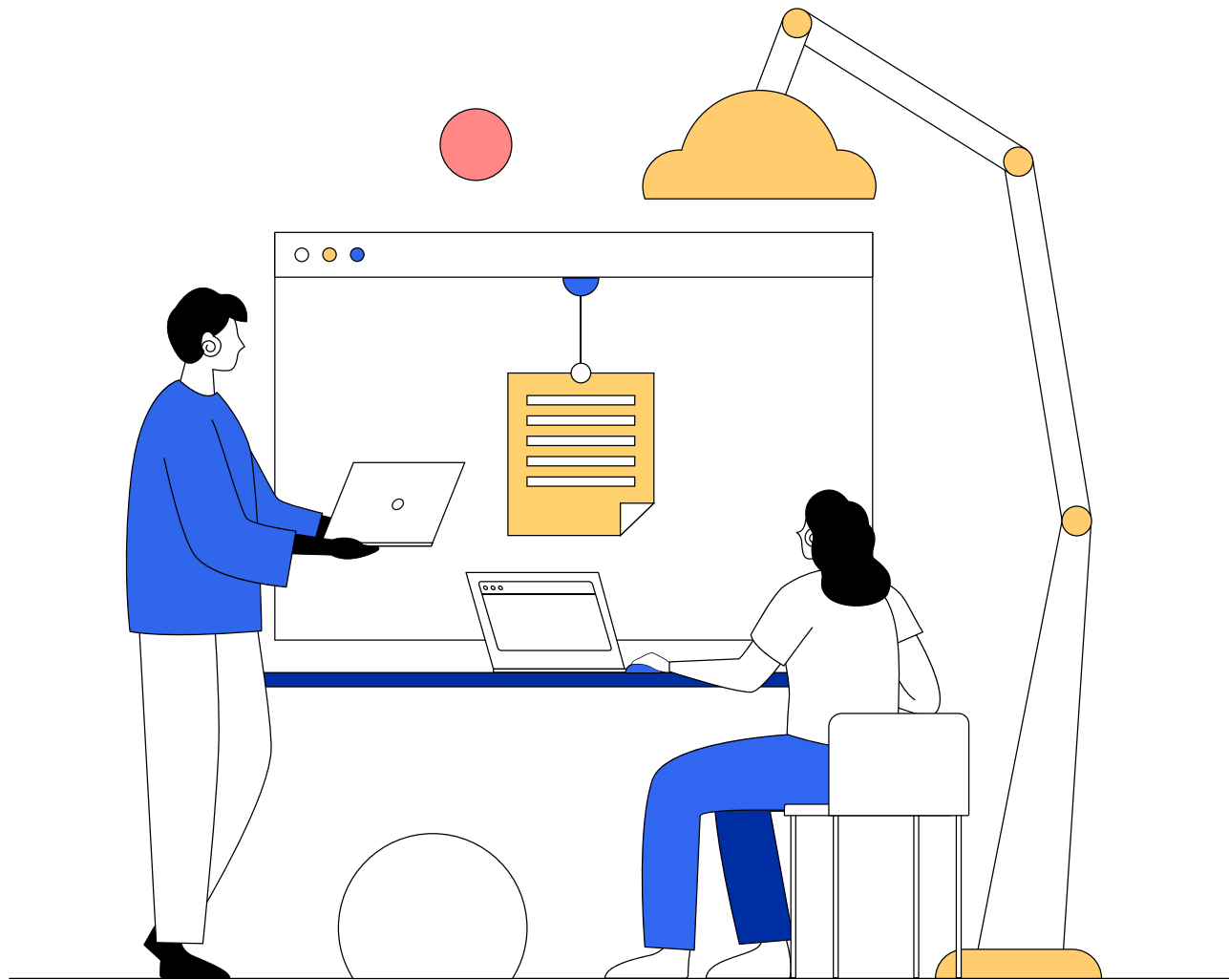
在 Linux Kernel 5.10 版本中，华为的代码贡献量排名全球第一。坚持内核创新，持续贡献上游社区。

## 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。



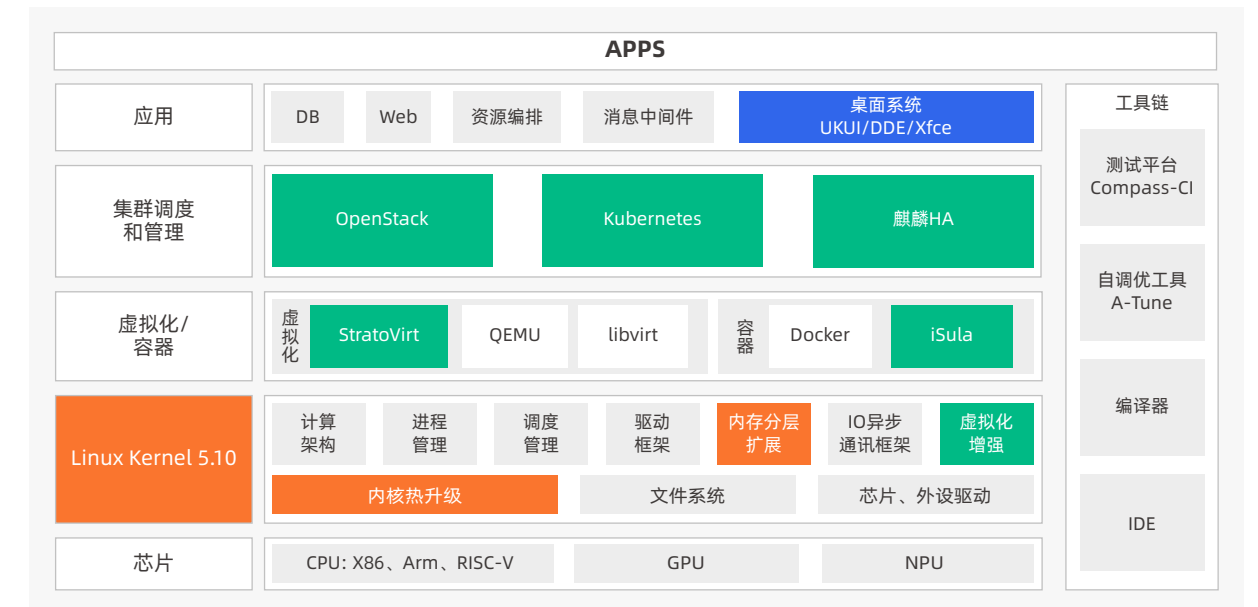
## 平台架构



### 系统框架

openEuler 操作系统使用场景主要是服务器，包括基础加速库、虚拟化、内核、驱动、编译器、系统工具、OpenJDK 等组件。

创新架构，全栈优化，打造全场景协同的 One OS，充分释放多样性算力。



#### 引领内核创新：

- **Linux Kernel 5.10**：调度、IO、内存管理深度优化。
- **内存分层扩展 etMem**：支持多种内存、存储介质统一管理，系统容量平滑扩展。
- **内核热升级**：内核漏洞快速修复，业务不感知。

#### 构筑云化基座

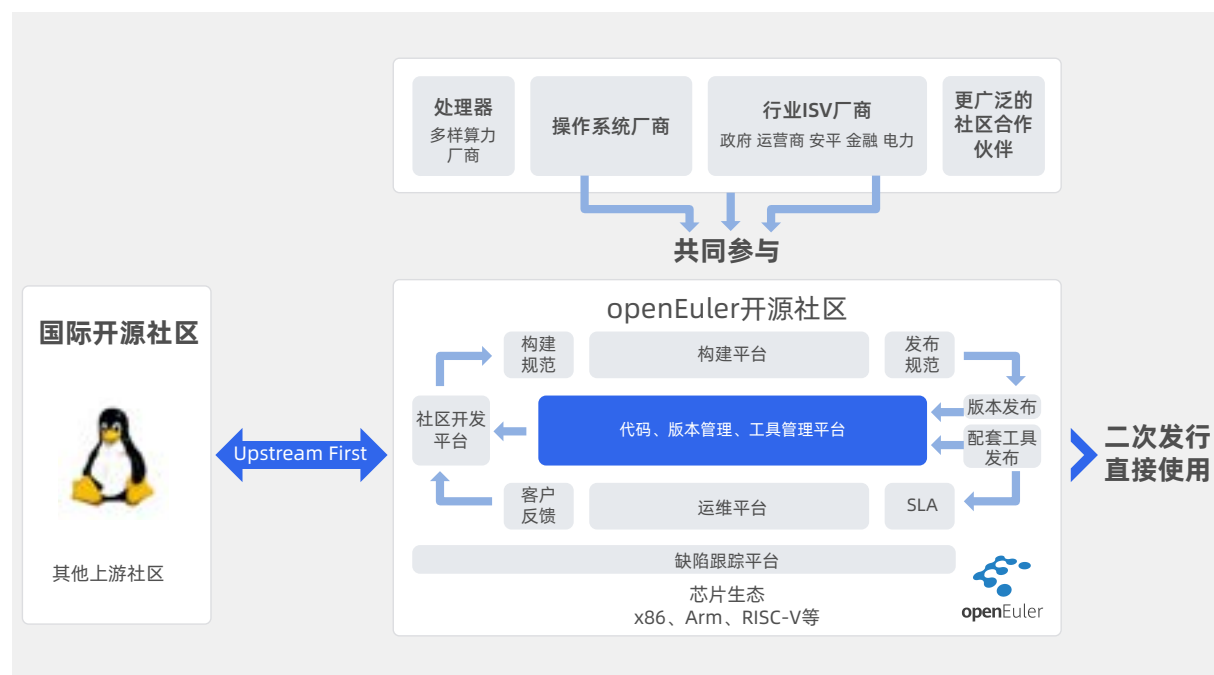
- **iSula**：iSulad 支持本地卷管理，isula-build 新增镜像拉取、推送等功能。
- **StratoVirt& 虚拟化**：支持内存弹性、大页、增强 IO 子系统、通过多通道并发提升 IO 性能。
- **OpenStack&Kubernetes**：向云而生，集成两大主流云计算调度和管理软件，构筑云化基座。
- **HA 高可用集群方案**：麒麟软件贡献的 HA 高可用集群方案，故障秒级切换。

#### 繁荣社区生态：

- **更多桌面环境支持**：UKUI、DDE、Xfce 桌面环境，丰富社区桌面环境生态。

## 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。



## 运行环境



物理机

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/>

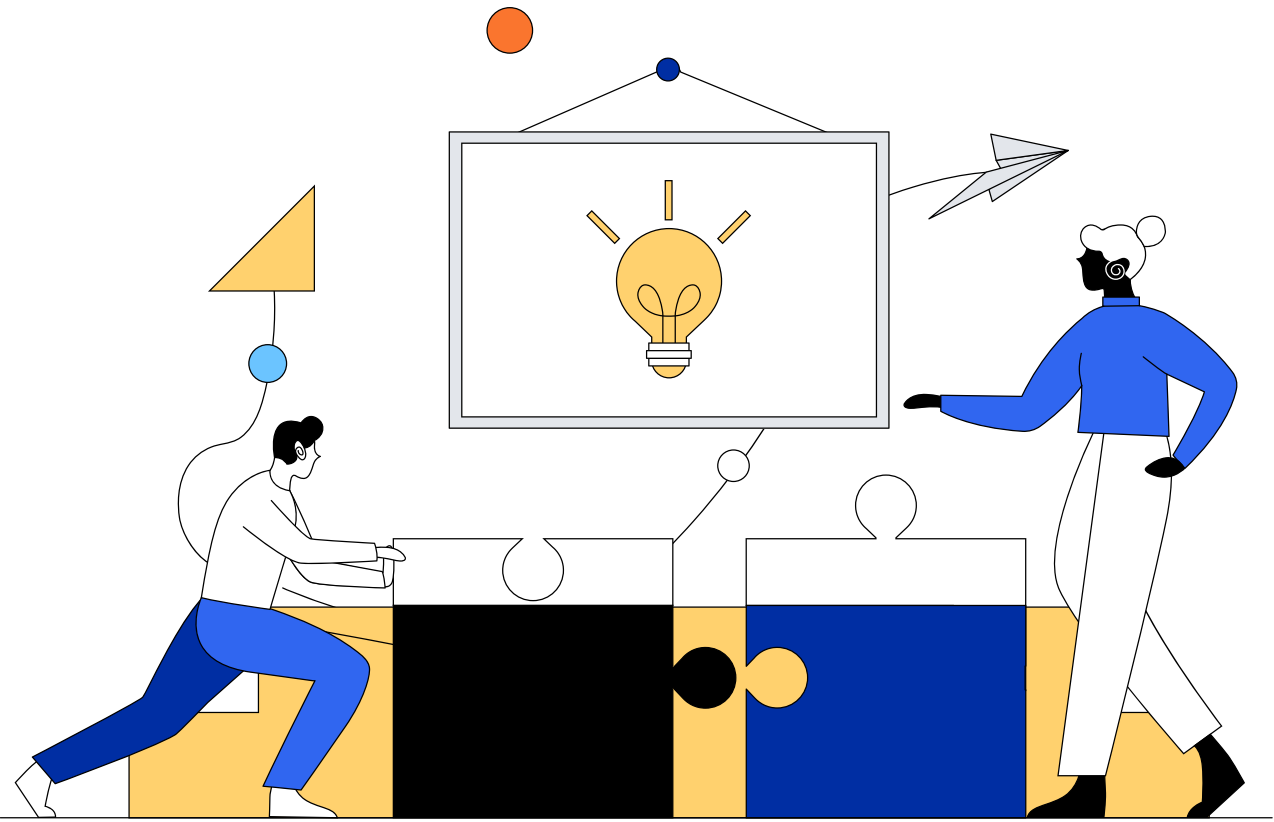
部件名称	最小硬件要求
架构	AArch64、x86_64
内存	为了获得更好的体验，建议不小于 4GB
硬盘	为了获得更好的体验，建议不小于 20 GB

虚拟机

openEuler 安装时，应注意虚拟机的兼容性问题，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	最小虚拟化空间要求
架构	AArch64、x86_64
CPU	2 个 CPU
内存	为了获得更好的体验，建议不小于 4GB
硬盘	为了获得更好的体验，建议不小于 20 GB

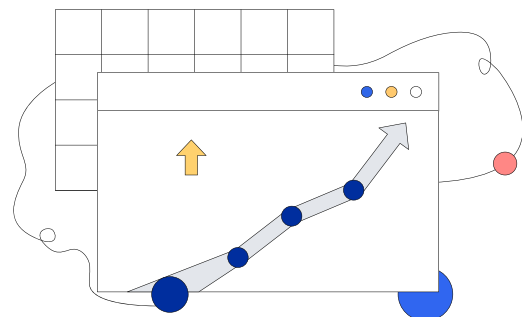




## openEuler 内核中的技术创新

openEuler 21.03 基于 Linux Kernel 5.10 内核构建，带来了超过 20 多项性能与功能提升：

- 支持调度器优化：**优化 CFS Task 的公平性，新增 NUMA-Aware 异步调用机制，在 NVDIMM 初始化方面有明显的提升；优化 SCHED\_IDLE 的调度策略，可以显著改善高优先级任务的调度延迟，降低对其他任务的干扰。优化 NUMA balancing 机制，带来更好的亲和性、更高的使用率和更少的无效迁移。
- CPU 隔离机制增强：**支持中断隔离，支持 unbound kthreads 隔离，增强 CPU 核的隔离性，可以更好的避免业务间的相互干扰。
- 进程间通信优化：**pipe\_wait、epoll\_wait 唤醒机制优化，解决唤醒多个等待线程的性能问题。
- 内存管理增强：**优化内存初始化、内存控制、统计、异构内存、热插拔等功能，并提供更有效的用户控制接口。热点锁及信号量优化，激进内存和碎片整理，优化 VMAP、vmalloc 机制，显著提升内存申请效率。KASAN、kmemleak、slub\_debug、OOM 等内存维测特性增强，提升定位和解决内存问题的效率。
- cgroup 优化单线程迁移性能：**消除对 Thread Group 读写信号量的依赖；引入 Time Namespace 方便容器迁移。
- 系统容器支持对容器内使用文件句柄数进行限制：**文件句柄包括普通文件句柄和网络套接字。启动容器时，可以通过指定 --files-limit 参数限制容器内打开的最大句柄数。
- 支持 PSI：**提供了一种评估系统资源 CPU、内存、数据读写压力的方法。准确的检测方法可以帮资源使用者确定合适的工作量，帮助系统制定高效的资源调度策略，最大化利用系统资源，改善用户体验。
- TCP 发包切换到了 Early Departure Time 模型：**解决原来 TCP 框架的限制，根据调度策略给数据包设置 Early Departure Time 时间戳，避免大的队列缓存带来的时延，同时大幅提升 TCP 性能。
- 支持 MultiPath TCP 可在移动与数据场景提升性能和可靠性：**支持在负载均衡场景多条子流并行传输。



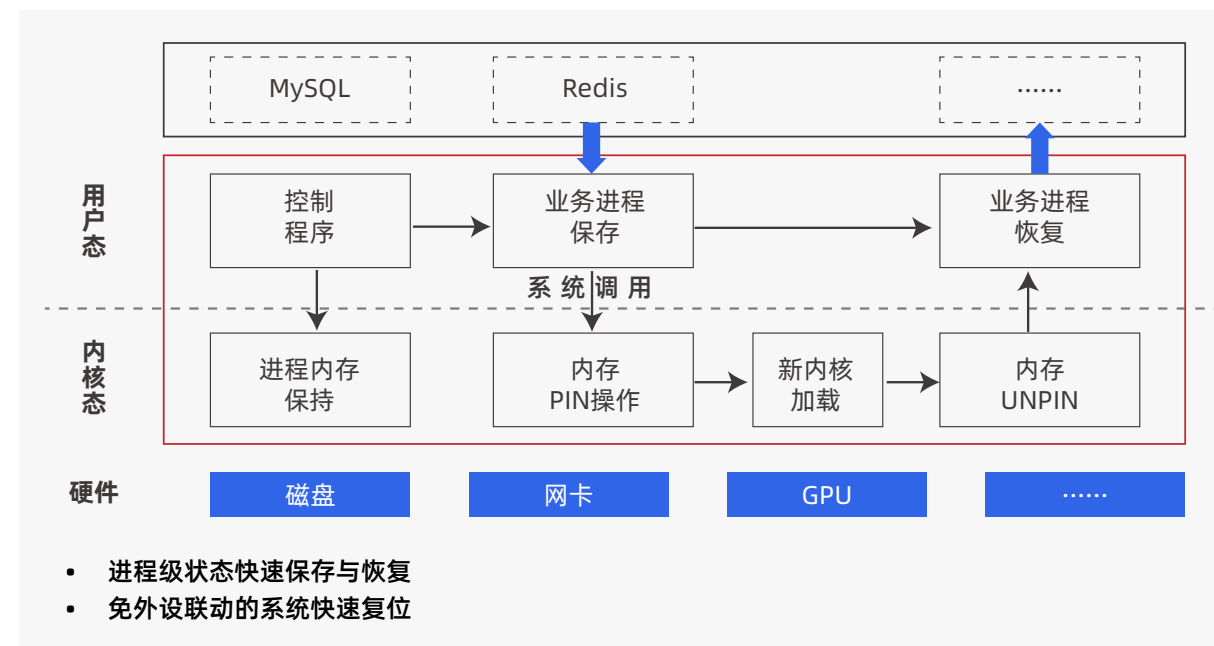
- Ext4 引入一种新的、更轻量级的日志方法：**- fast commit，可以大大减少 fsync 等耗时操作，带来更好的性能。
- dm-writecache 特性：**提升 SSD 大块顺序写性能，提高 DDR 持久性内存的性能。
- 支持 io\_uring：**io\_uring 是一个新的异步 IO 框架，支持 polling 模式，在 polling 模式下，性能提升显著，与 SPDK 接近，在队列深度较高时性能更好。
- IMA 商用增强：**在开源 IMA 方案基础上，增强安全性、提升性能、提高易用性，助力商用落地。
- 支持 per-task 栈检查：**增强对 ROP 攻击的防护能力。
- 支持 ILP32：**在鲲鹏 920 ARM64 环境上支持 32 位的应用程序。
- MPAM 资源管控：**支持 ARM64 架构 Cache QoS 以及内存带宽控制技术。
- 支持基于 SEDI 和 PMU 的 NMI 机制：**使能 hard lockup 检测。使能 perf nmi，能更精确的进行性能分析。
- 支持虚拟机热插拔：**ARM64 支持虚拟机 CPU 热插拔，提高资源配置的灵活性。
- ARM64 kdump 增强：**支持对 4G 以上地址的内存预留，支持更大内存的机器。
- 支持 Raspberry PI：**openEuler 21.03 内核原生支持 Raspberry PI，支持在 Raspberry PI 上使用 openEuler 21.03 内核进行源码调试。
- RISC-V：**RISC-V 平台支持 KVM。
- 硬件：**支持 Hi1822 智能网卡。

## 内核热升级

修复内核高危安全 CVE 漏洞，升级加载新的安全内核，内核热升级可以保证关键核心业务不中断。内核在升级前检测业务使用资源状态，利用快速冻结技术对业务使用资源的状态进行冻结，利用快速加载技术对新内核进行加载，然后对冰冻的资源状态进行恢复。

为了在修复内核问题的过程中保证业务不中断，热补丁技术应运而生。但热补丁技术有很多局限性，例如：不能改变数据结构、不能修复 inline 函数、不能修复复杂逻辑问题等等，导致使用热补丁技术仅能够修复 20% 左右的问题。在数据中心还会带来运维问题，例如补丁叠加导致运维基线无法确定，最终导致运维难度加大。内核热升级技术应运而生。

### 功能描述



- 控制程序：**对整个内核热升级流程进行指挥串联，交换内核态补丁与用户态守护程序信息，对指定业务进行内核热升级，并进行质量可靠性保证，在升级失败后保证回滚到旧内核上。
- 业务进程保存：**利用系统 Checkpoint 保存业务进程和资源状态，保证系统业务状态与资源的一致性。
- 新内核加载：**利用系统 Kexec 机制实现新内核的快速加载，保证端到端业务秒级恢复。
- 业务进程恢复：**利用系统 Restore 技术对已保存的业务状态与资源进行恢复。

### 应用场景

#### 应用场景 1：内核 CVE 修复

典型应用程序（如 Nginx、Redis、Mysql 等）运行在物理机或者虚拟机上，保持有大量的连接及占用大量的内存。当该内核出现了严重 CVE，需要进行修复的时候，该业务需进行三个选择：

- 重启主机：**连接的用户会感知到业务中断，且由于重启时间长，导致业务中断时间较长，用户不可以接受。
- 业务迁移：**典型业务占用大量的内存，需要对整个内存进行迁移，这样会导致长时间的业务性能低下问题，不能满足客户性能指标。
- 热补丁：**无法修复所有的问题，部分问题热补丁无法打入。

利用内核热升级，可以保持业务无中断与性能无损的情况下，修复内核问题。

## 应用场景 2：内核小版本升级

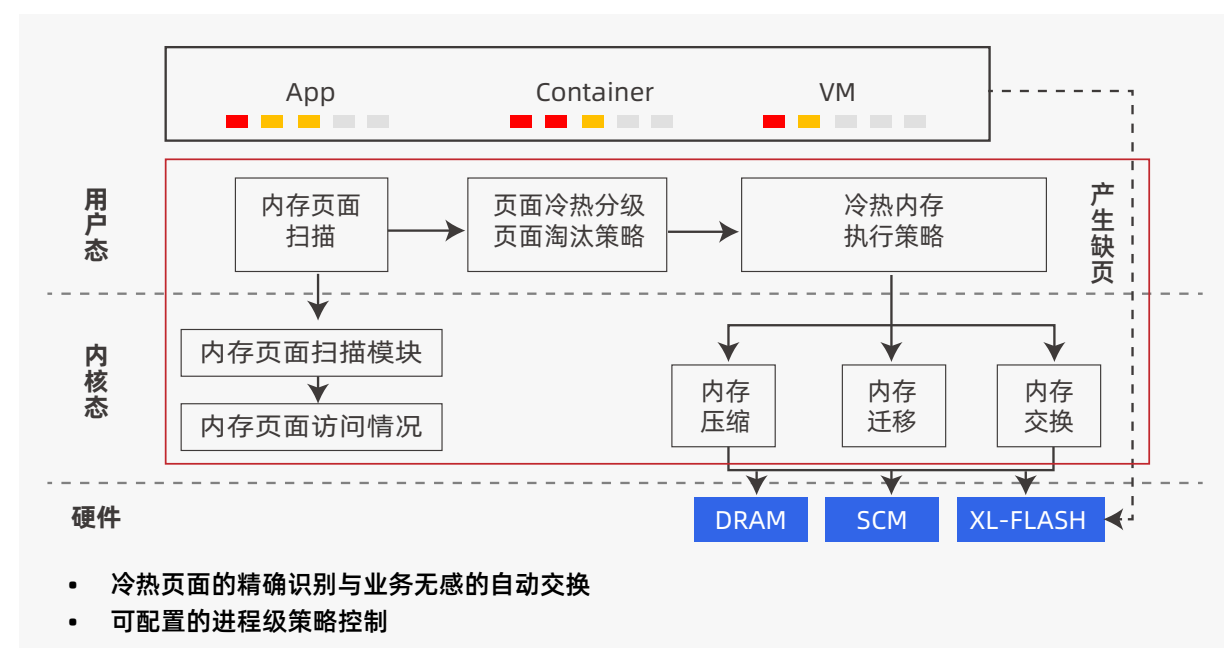
操作系统整个生命周期中，需要不断引入新特性来满足客户需要，这些内核特性对业务有非常大的作用（性能、安全与调测等）。如果引入该特性，需要重启业务与重启主机，这样将导致业务客户端感知到中断，对用户不友好，利用内核热升级，可以让客户无感知的情况下对引入新的内核特性。

## 内存分层扩展 etMem

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。

内存分层扩展通过 DRAM 和低速内存介质，如 SCM、AEP 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较好，收益较大。

### 功能描述



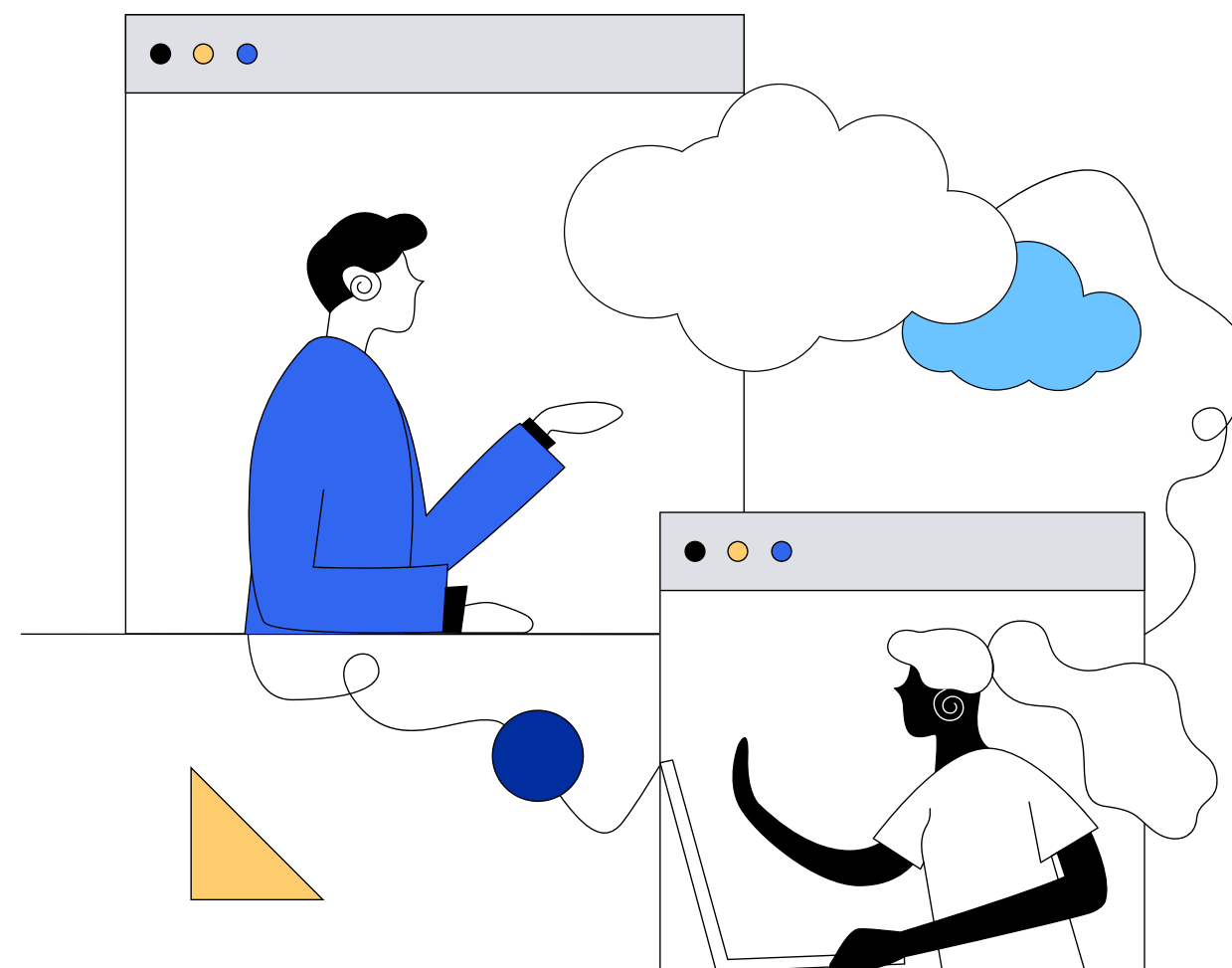
1. **进程级控制：**etMem 支持通过配置文件来进行内存扩展的进程，相比于操作系统原生的基于 LRU 淘汰的 kswap 机制，更加灵活和精准。
2. **内存扫描：**新增内核功能，由用户态 etMem 进程触发，对指定进程进行内存访问扫描，并返回扫描结果。
3. **冷热分级：**由用户选择冷热分级策略配置文件，对获取到的内存访问结果进行分级，区分出热内存和冷内存。
4. **淘汰策略：**根据 etMem 配置文件和系统环境配置，对冷内存进行淘汰，淘汰流程使用内核原生能力，安全可靠，用户无感知。

### 应用场景

**应用场景：**节点内业务进程内存分层扩展

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。

## 云化基座





## 容器基础设施 iSula

iSula 是 openEuler 的容器技术项目，包括容器全栈生态中的多个软件。其中通用容器引擎 iSulad 是一种新的容器解决方案，提供统一的架构设计来满足 CT 和 IT 领域的不同需求。相比 Golang 编写的 Docker，iSulad 使用 C/C++ 实现，具有轻、灵、巧、快的特点，不受硬件规格和架构的限制，开销更小，应用领域更为广泛。

### 新增功能

openEuler 21.03 相比 openEuler 20.09 版本有以下特性更新：

- **iSulad 支持本地卷管理：**用以支持在创建或运行容器时创建本地卷用于数据的持久化，允许用户将需要持久化的数据写入卷内进行保存，卷内的数据在容器删除后不会消失。可以通过 iSula 的卷管理命令对卷进行管理。
- **isula-build 新增功能：**
  - ① 支持 pull 和 push 命令：通过命令 pull 拉取镜像到本地磁盘，通过 push 镜像推送至远端仓库。
  - ② Manifest 列表管理：在不同的架构中使用相同的 Manifest 获取对应架构的镜像，Manifest 包含 create、annotate、inspect 和 push 子命令。
  - ③ 支持 OCI 镜像格式：支持 OCI 格式镜像的构建、保存、导入导出、推送和拉取。
  - ④ 支持在特定用户组下运行：isula-build 的组件支持在特定用户组下运行，在用户组内的非特权用户可以操作 isula-build，提高安全系数。
  - ⑤ 支持导入和导出多个镜像：添加支持保存多个镜像到一个 tarball、或导入包含多个镜像的 tarball。

更多详细内容请参考 openEuler 20.09 技术白皮书：<https://openeuler.org/openEuler-whitepaper.pdf>

## 虚拟化平台 StratoVirt

StratoVirt 是 openEuler 开源平台上实现面向云数据中心的企业级虚拟化平台，Strato 意指大气层中的平流层，寓意为保护 openEuler 平台上业务平稳运行的轻薄的保护层。

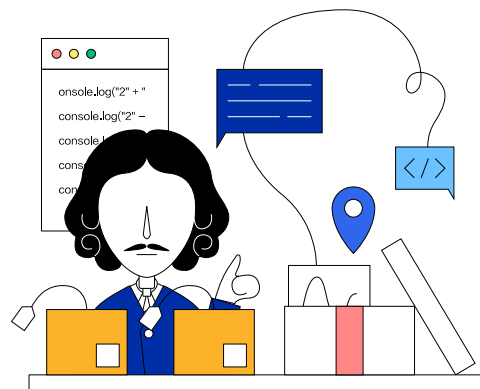
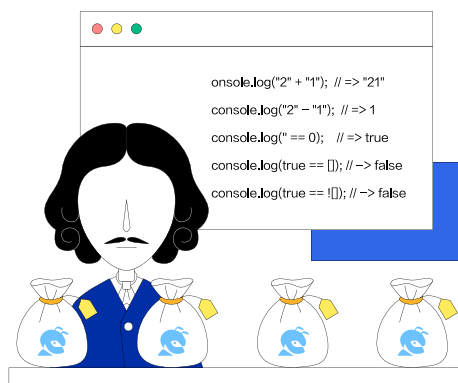
QEMU 代码量庞大、CVE 安全漏洞频出，业界逐步演进出以 Rust 语言实现的 CrosVM、FireCracker 和 Rust-VMM 等新型虚拟化架构或组件。具备安全、轻量、高性能、低损耗、组件灵活拆分的特点。

### StratoVirt 主要优势如下：

- **强安全性：**基于 Rust 实现语言级安全，模型设计上最小化攻击面，实现多租户物理隔离。
- **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，支持 Serverless 负载。
- **软硬协同：**StratoVirt 支持 x86 的 VT，支持鲲鹏的 Kunpeng-V。
- **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。

### 新增功能

- **支持内存弹性：**快速按照工作负载的内存需求来实时分配和回收内存，通过 virtio-balloon 内存回收速度最高可达 3GB/s。



- **支持大页：**在轻量级虚拟机下支持大页，可为轻量级虚拟机提供连续的物理内存页面，提高虚拟机内存访问效率。
- **IO 子系统增强：**支持多通道并发 IO 能力，提高 IO 性能。支持 IO-QOS 能力，提升虚拟机 IO 流量管理的灵活性和稳定性。
- **系统调用过滤：**通过极简设备模型设计和 SECOMP 过滤系统调用，最简配置下仅需使用 35 个系统调用，有效减小系统攻击面。

更多详细内容请参考 openEuler 20.09 技术白皮书：<https://openeuler.org/openEuler-whitepaper.pdf>

## OpenStack

OpenStack 是一个由 NASA 和 Rackspace 合作研发并发起的，以 Apache 许可证授权的开源项目。OpenStack 支持几乎所有类型的云环境，提供实施简单、可大规模扩展、生态繁荣、标准统一的云计算管理平台。OpenStack 通过各种互补的服务提供各种 IaaS 服务的解决方案，每个服务通过 API 进行集成。

OpenStack Victoria 是 2020 年 OpenStack 社区最新稳定发布版本，包含计算、存储、网络、PaaS、安全、集群管理等多个模块。已经完成 Nova、Keystone、Neutron、Glance、Ironic、Horizon 几个模块对 openEuler 的适配。

OpenStack 的迁移适配工作由中国联通主导的 OpenStack SIG 负责完成。

### 功能描述

#### • Keystone

提供了 OpenStack 其余组件的认证信息和令牌的管理、创建、修改等功能，使用 MySQL 等数据库存储认证信息。

#### • Nova

提供虚拟机的创建、运行、迁移、快照等服务，提供 API 与控制节点对接，由控制节点下发任务，使用 nova-api 进行通信。

#### • Neutron

网络管理服务，提供了对网络节点的网络拓扑管理，负责管理私有网络与公有网络的通信、虚拟机网络之间通信 / 拓扑、管理虚拟机之上的防火墙等等，同时提供 Neutron 在 Horizon 的管理界面。

#### • Glance

用于管理虚拟机部署时所能提供的镜像，包含镜像的导入、格式以及模板，是一套虚拟机镜像发现、注册、检索系统。

#### • Cinder

提供相应的块存储，虚拟出一块磁盘，可以挂载到相应的虚拟机之上，不受文件系统等因素影响，对虚拟机来说，相当于新加了一块硬盘，可以完成对磁盘的任何操作，包括挂载、卸载、格式化、转换文件系统等等操作。

#### • Ironic

提供裸金属服务器纳管、部署、生命周期维护功能。

#### • Horizon

提供界面化的操作控制台。通过调用 API 实现 OpenStack 内部资源进行管理和展示。

### 应用场景

#### 应用场景 1：云计算平台部署

用户可通过 openEuler 软件安装源进行 OpenStack 平台部署，支持基本云平台功能。

#### 应用场景 2：商业云计算平台开发

以 OpenStack 为基础架构的云计算服务商，可直接使用 openEuler 作为服务器操作系统，并以此为基础平台进行定制开发。

## HA 高可用集群方案

HA 高可用集群软件是一个通用的高可用集群方案，致力于为用户提供业务连续性保障、数据持续保护、灾难恢复的高可用环境。该方案利用健康检测、秒级切换等功能，可以解决软硬件或人为原因造成的单点集群故障而引起的业务中断问题。有效确保单点系统或集群上关键任务应用程序和数据的稳定性和可靠性，保护业务系统的高可用。

HA 高可用集群方案迁移适配工作由麒麟软件主导的 HA SIG 负责完成，并且在 openEuler 开源。

### 功能描述

- 多种灵活的集群援备模式

支持双机热备、双机互备、多机备份 (N+M) 等多种保护模式，支持物理机 - 物理机、物理机 - 虚拟机等多种模式；支持多种文件系统及主流存储设备，满足业务应用各种保护需求。

- WEB 的集群管理界面

HA 高可用集群方案提供图形化 Web 方式实现高可用集群管理，HA-API 是麒麟软件依托多年对 HA 技术的深入研究和应用，自行研发的一套后端接口。HA SIG 在此基础上用 Go 语言进行了代码重构。目前已经发布了后端接口 V1.0 版。相对众多前端管理平台，其具有友好、直观、易操作等特点。有关 HA 高可用集群方案的通用资源和保护配置都可以在管理界面中完成。

- 心跳方式

支持网络心跳，包括单心跳和双心跳模式，全方位监控共享数据资源。

- 秒级切换能力及故障恢复后自动回迁机制

完善的保护机制及丰富的应用程序代理，使服务器或资源出现故障时能够秒级检测并自动完成切换操作。当故障排除后，服务自动回迁，支持 7×24 小时永不停机的企业级应用系统。

- 极端情况下对数据完整性的保护

通过监控共享数据资源，在双机出现极端故障的情况下，保护服务器数据的完整性。

- 极低的系统资源占用

系统资源占用极低，基本不会与被保护应用争抢系统资源。

- 完整的日志

HA 高可用集群方案提供完整的日志及相关调试信息，便于系统管理员进行监控、管理。

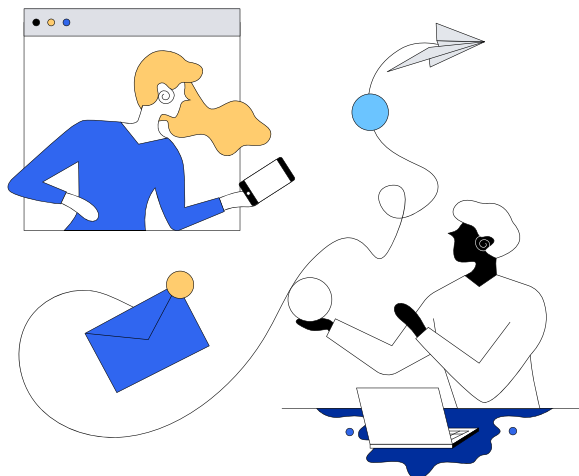
### 应用场景

#### 应用场景 1：业务连续性保障

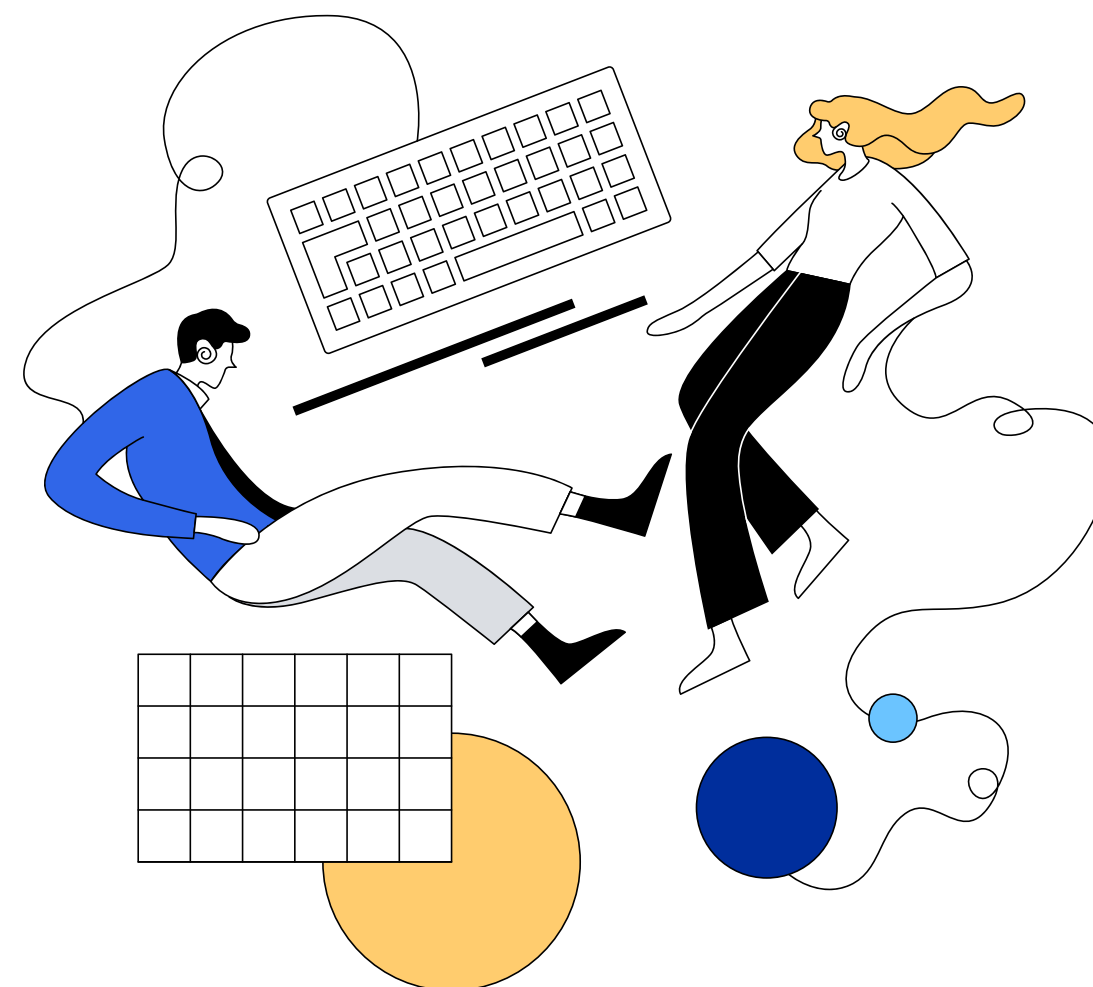
随着企业信息系统的广泛应用和深入发展，用户的核心应用数量越来越多，特别一些关键的业务系统，更是要求提供 7×24 小时不间断服务，对业务连续性的要求很高。HA 高可用集群软件可以做到双机热备、双机互备和 N+M 模式，将出现故障的服务器中的应用切换到其他服务器中继续运行。错误检测时间小于 10 秒，可在 1 秒内执行业务切换操作。

#### 应用场景 2：数据库和系统服务保护

HA 高可用集群方案支持主流系统和应用软件，如 Nginx、httpd、MariaDB 等，支持二次开发。HA 高可用集群软件能够保护这些数据库和系统服务，当集群中节点上的软件出现故障，会自动切换到其他节点。



## 生态繁荣



软件包信息和依赖管理工具 pkgship

pkgship 是一个 RPM 软件包依赖关系的管理工具，为开发者提供软件包依赖查询服务，支撑开发者引入、升级和删除软件包，同时为应用兼容性评估提供数据支撑，使开发者能够快速了解跨操作系统的多版本软件包依赖差异。

pkgship 在 openEuler 社区提供前端服务，可供开发者查询详细的依赖图谱关系以及依赖层级关系，并支持下载软件包依赖关系文件。

功能描述

- 包信息导入：pkgship 根据配置不同的软件库服务器，可导入对应版本的 OS 内所有软件包的信息并对其依赖进行整合分析。
- 包信息查询：pkgship 支持查询导入的 RPM 源码包、二进制包信息，包含 version、license、description 信息，支持查询软件包中的文件。
- 包依赖数据查询：pkgship 获取软件包的安装依赖、编译依赖、自依赖和被依赖的依赖列表、图谱以及详细依赖关系文件。

应用场景

应用场景 1：软件包兼容性评估

不同版本的操作系统，同一软件包的差异评估，是社区开发者评估软件包兼容性的重要参考信息。pkgship 基于不同的 Linux 发行版和同一个 Linux 发行版的不同版本，给开发者提供同一软件包的版本及包含文件的信息，能帮助开发者判断软件包之间的异同。

应用场景 2：引入新软件包

在开源社区中引入新软件包时，软件包在编译、安装时的经常遇到问题。不同软件包的编译依赖、安装依赖不尽相同，所依赖的软件包又会引入其他的依赖包。pkgship 能帮助开发者快速的识别依赖包范围，提升引入效率。

应用场景 3：升级、删除软件包

在某版本中升级或删除软件包时，需要评估该软件包影响范围。pkgship 可以识别该软件包被哪些软件包依赖，帮助开发者快速圈定影响范围。

桌面环境

UKUI

UKUI 是由麒麟团队开发的基于 Linux 发行版的轻量级桌面环境。UKUI 3.0 使用 Qt 开发，以用户体验为根本，减繁取易，将视觉和交互舒适自然的结合在一起。

UKUI 更注重易用性和敏捷度，各元件相依性小，可以不依赖其它套件而独自运行，提供亲切可靠的交互功能，为 openEuler 21.03 版本提供可靠、高效的桌面环境。

功能描述

- 控制面板：进行系统的基本设置如日期与时间、个性化设置、设备管理等。
- 开始菜单：管理系统中已安装的所有应用，并提供智能记忆常用应用列表功能。可在默认和全屏尺寸两者之间切换，支持汉字、英文、拼音或首字母等多种检索方式。
- 任务栏：支持深浅双色主题和毛玻璃效果，支持文件、文件夹、终端、网页、图片等窗口的实时预览。
- 侧边栏：侧边栏一分为二，上半部分专管通知，下半部分为剪贴板和小插件操作，提供了诸如闹钟、便签本等小插件。

- 文件管理器：支持多标签页显示，将文件管理器的搜索栏和地址栏合二为一，重构的搜索功能，可以更加快速精准的搜索所需要的文件。

应用场景

应用场景 1：个人 PC 领域

为用户提供了操作简单，更加高效的桌面环境。

应用场景 2：服务器领域

在服务器领域，UKUI 资源占用平稳，功能稳定，细节严谨，简化用户操作流程，为用户提供良好的使用体验。

DDE

统信桌面环境简称 DDE，是一款安全、稳定且易用的 Linux 桌面环境，采用 Qt 作为前端，Go 为后端，Gala 作为 WM，LightDM 作为 DM。

功能描述

- 控制中心：UOS 通过控制中心来管理系统的基本设置，包括帐户管理、网络设置、日期和时间、个性化设置、屏幕显示设置、系统和应用更新升级等更多设置。
- 窗口管理器：窗口管理器可以在不同的工作区内展示不同的窗口内容，通过窗口管理器您可以同时使用多个桌面，对桌面窗口进行分组管理。
- 桌面组件：桌面是用户登录后看到的主屏幕区域，在桌面上用户可以根据自己的习惯设置壁纸、屏保等，还可以增加常用的应用快捷方式。同时包含通知消息、剪贴板等常用组件。
- 文件管理器：图形化分类管理文件，提供文件操作、检索等；同时提供基础的光盘刻录、外部设备与网络挂载。
- 任务栏：任务栏是指位于桌面底部的长条，主要由启动器、应用程序图标、托盘区、系统插件等组成。在任务栏中，您可以对其上的应用程序进行打开、新建、关闭、退出等操作，还可以设置音量，连接 Wi-Fi，进入关机页面等。
- 启动器：启动器帮助您管理系统中已安装的所有应用，在启动器中使用分类导航或搜索功能可以快速找到需要的应用程序。
- DTK：全称是统信工具套件，是基于 Qt5 开发，提供大量界面美观且实用的 UI 图形库，同时也包含工具库，为开发者提供开发及调试的工具。

Xfce

Xfce 是一个轻量级的类 Unix 的桌面系统，其特点是快速、轻量，界面美观和对用户友好，它被设计用来提高您的工作效率，在节省系统资源的同时，能够快速加载和执行应用程序。

Xfce 由独立的软件组件构成，可根据需要单独或者组合在一起使用。

Xfce 由 C 语言写成，该版本已将所有核心组件移植到 Gtk3 和 GDBus。大多数组件还支持 GObject Introspection。

本项工作旨在为 openEuler 21.03 版本提供桌面环境支持，扩充 openEuler 在桌面办公领域的能力。

Xfce 的迁移适配工作由北京拓林思主导的 Xfce SIG 负责完成。

## 功能描述

- Application Finder (xfce4-appfinder) : 快速运行程序和命令的程序。
- Configuration Storage System (xfconf) : 基于 D-Bus 的配置存储系统。
- Desktop Manager (xfdesktop) : 配置桌面背景、图标、启动器及目录。
- Development Tools (xfce4-dev-tools) : Xfce4 的开发工具集。
- Helper Applications (exo) : 管理首选应用和编译 .desktop 文件的程序。
- File Manager (thunar) : Xfce 文件管理器。
- Menu Library (garcon) : 菜单相关库。
- Panel (xfce4-panel) : 应用启动器，窗口按钮，程序菜单，工作区切换器放置面板。
- Power Manager (xfce4-power-manager) : 设备电源管理程序。
- Session Manager (xfce4-session) : 保存桌面状态及下次启动时恢复的程序。
- Settings Manager (xfce4-settings) : Xfce4 配置程序。
- Thumbnail Service (tumbler) : D-Bus 服务，为应用程序请求各种 URI 方案和 MIME 类型的缩略图。
- Utility Sharing Library (libxfce4util) : 用于在 Xfce 应用程序之间共享常用的非 GTK+ 实用程序的库。
- Widget Sharing Library (libxfce4ui) : 用于在 Xfce 应用程序之间共享常用的 Xfce 小部件。
- Window Manager (xfwm4) : 窗口管理器。

## 增强特性





毕昇 JDK

毕昇 JDK 是华为基于 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 运行在华为内部 500 多个产品上，积累了大量使用场景，并针对 Arm 进行性能优化。毕昇 JDK 提供了对 OpenJDK8 和 OpenJDK 11 的支持，其中毕昇 JDK 8 与 Java SE 标准兼容，同时持续贡献 OpenJDK 上游社区。

毕昇 JDK 特征优势如下：

- 1. 稳定、高效：在 SpecJbb 等基准测试中性能大幅优于 AdoptOpenJDK 产品，稳定、高效。
- 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：毕昇 JDK 免费提供发行包和开源代码。

功能描述

- 支持 **Java Flight Recorder**：低负载的在线性能诊断工具，性能损耗小，对于长时间运行的应用程序，默认设置性能损耗小于 1%。可以在生产环境中开启，通过 ``-XX:+FlightRecorder`` 命令启用，并生成一个 dump 文件用于分析，配合 JMC 工具可以获得更好的性能分析可视化效果。
- 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。
- 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。
- 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。
- **G1GC 支持 NUMA-Aware 特性（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。
- 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。

应用场景

应用场景 1：大数据应用

毕昇 JDK 针对大数据应用在运行时、GC 和 JIT 都有优化，例如优化 GC 中内存的分配和回收，消除 JIT 代码中冗余内存屏障等。在大数据应用的基准测试中，使用毕昇 JDK 和使用 AdoptOpenJDK 相比可以获得 5%~20% 的性能收益。辅以使用毕昇 JDK 中的优化特性，如 AppCDS 和 ZGC 还能获得更高的性能收益。

应用场景 2：云计算

G1GC 内存伸缩可以有效节约资源，测试发现在低流量场景中内存可以节约 60% 以上。同时毕昇 JDK 通过动态的释放策略保证内存使用平滑变化。

应用场景 3：鲲鹏服务器的 Java 应用

毕昇 JDK 针对鲲鹏服务器的弱内存模型进行优化，无效内存屏障。充分利用硬件性能，在 G1GC 引入 NUMA-Aware 特性，提高应用访问内存的效率。提供和增强 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。

GCC

在鲲鹏通过 SVE、pipeline 等实现了软硬件深度协同优化提升 GCC 性能，主要优势如下：

- 1. 性能优异：充分发挥鲲鹏的硬件特性，运行效率更高，在 SPEC CPU 2017 等基准测试中性能大幅优于上游社区的 GCC 9.3 版本。
- 2. 开源：提供发行包和开源代码，坚持创新并持续贡献上游社区。

功能描述

- 支持 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。
- 支持 ARM 架构下 128 位四精度浮点数据库，能够有效提升浮点数运算的精度。
- 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大幅度提升程序运行的性能。
- 支持 ARM 架构下 BF16 和矩阵乘法指令，能够为有需要的开发者提供这部分指令和 intrinsic 的支持。
- 支持新的循环优化，启用此选项，可以有效减少冗余的循环，并且合理的拆分、合并多个循环，提升程序运行的性能。
- 支持结构体优化，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，能够大幅度提升程序运行的性能。

应用场景

应用场景 1：高性能计算领域

在高性能计算领域的测试中，运行 WRF 应用，相比于上游社区的 GCC 9.3 版本能够获得 10% 左右的性能收益。

应用场景 2：鲲鹏上的应用程序

在鲲鹏的 SPEC CPU 2017 基准测试中，相比于上游社区的 GCC 9.3 版本能够获得 10% 以上的性能收益。

机密计算框架 secGear

在云上或数据中心中，为了保护敏感数据的安全性，通过一个安全隔离区来保护使用中的数据称为机密计算技术。硬件会提供一个可信的隔离执行环境供软件使用，让处理敏感数据的软件运行在可信执行环境中，在普通环境里无法访问到可信执行环境李的内容。

Intel x86 处理器提供 SGX 特性，划分一部分主存给可信执行环境，对可信执行环境内存进行加密，CPU 加载内存的时候进行解密，外部攻击者无法访问到明文数据。

Arm 处理器提供 TrustZone 特性，在内存控制器上实现访问控制，划分给安全区的内存和设备普通模式下的 CPU 无法访问，外部攻击者除非攻破安全区软件，否则无法访问到安全区里的内容。

secGear 是基于硬件机密计算技术为开发者提供的一个应用开发框架，开发者基于 secGear 框架可以简化编写安全应用的复杂度，提升开发效率。

secGear 支持 x86、Arm 等多种芯片架构，不同的体系架构下硬件机密计算技术的实现和编程接口是不一样的，通过 secGear 开发框架可以让开发者用在编写代码时获得一致的编程体验。

功能描述

- 提供 API 支持应用对机密计算生命周期的管理和对机密计算安全函数的调用。
- 为安全侧应用开发提供类 POSIX 接口，实现非安全侧一致的编程体验。

- 支持可信执行环境状态下的数据加密和持久化，支持本地和远程证明。
- 提供必要的代码辅助生成工具及机密计算二进制签名工具。
- 提供 C 编程语言支持。
- 支持 Intel SGX 和 ARM TrustZone。

## 应用场景

### 应用场景 1：多方计算

多方计算：组织或实体想使用多方拥有的数据做运算而互相不泄露各自的数据给第三方。比如有三个公司，每个公司有各自的数据，而且每个公司的数据只能自己公司可以访问，有一种情景，三家公司的数据进行合并通过 AI 训练进行运算，如何避免数据的泄露？

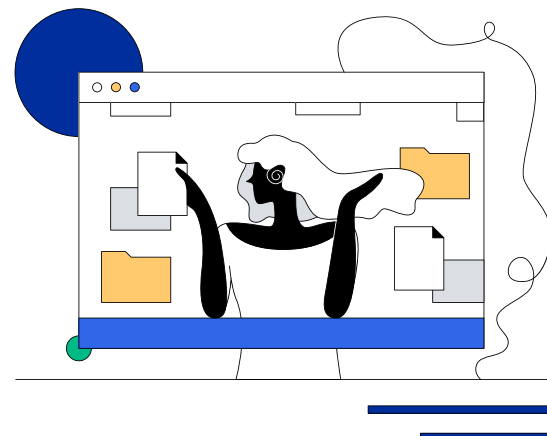
这三家公司可以通过机密计算技术，将数据通过加密的通道传输给对方的可信执行环境，通过远程证明的方式保证可信执行环境及运行的代码可信，保障各家公司共享数据做 AI 训练而不泄露各自的数据。

### 应用场景 2：密钥管理服务

目前公有云的密钥管理服务，一般会基于硬件 HSM 来实现密钥的安全管理。这些服务可以基于 secGear 来开发基于机密计算的安全模块替代硬件加密模块来保护密钥。

### 应用场景 3：安全数据库

安全数据库，在某些应用场景下，为了保护数据库里的内容，数据库所有者只希望应用能通过 SQL 访问返回的结果，对数据库访问的过程需要严格保护，这种情况下，可以基于 secGear 开发安全数据库应用，保障数据库的处理过程在可信的执行环境中运行，保证数据库中的数据的安全。



## 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。
- ② 输入订阅邮箱并点击“订阅”按钮。
- ③ 登录邮箱并回复从 [openeuler.org](https://openeuler.org) 发来的确认邮件。

如果接收到从 [@openeuler.org](mailto:@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

- ① 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 + “-join” 后缀组成）。
- ② 回复从 [openeuler.org](https://openeuler.org) 发送的确认邮件。

以订阅 Dev([dev@openeuler.org](mailto:dev@openeuler.org)) 作为例子，邮件如下：

receivers: [dev-join@openeuler.org](mailto:dev-join@openeuler.org)

subject: subscribe

content: NA

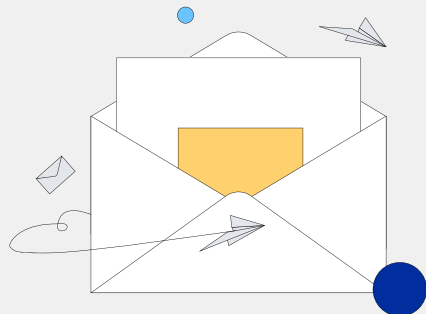
当前有 Announce 和 Discussion 两种列表类型，对于 Discussion 列表的发送方式，跟发送一般的邮箱到私有地址没有区别，按照平常的方式发送即可，用一对中括号加一个主题作为前缀添加到邮件主题里会是一种很好的实践，但是不是必须的，Announce 类型的邮件列表只是用于宣布消息或者注意事项，不接受邮件发布。

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 + “-leave” 为后缀组成）。
- 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

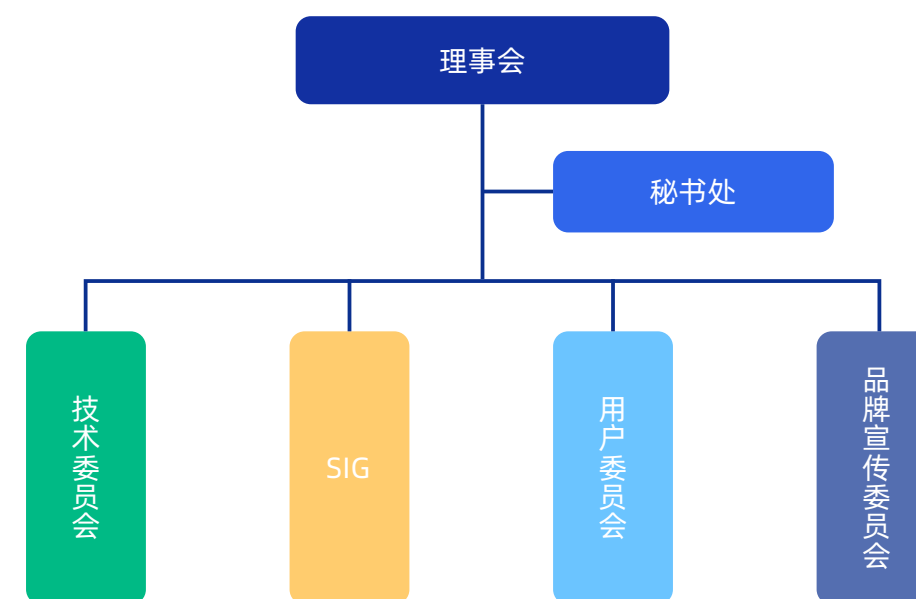
如果过程中遇到问题，请联系基础设施团队：

- 邮件：[infra@openeuler.org](mailto: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

秘书处是在 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. 引导社区在体系架构、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. 引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](mailto: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。

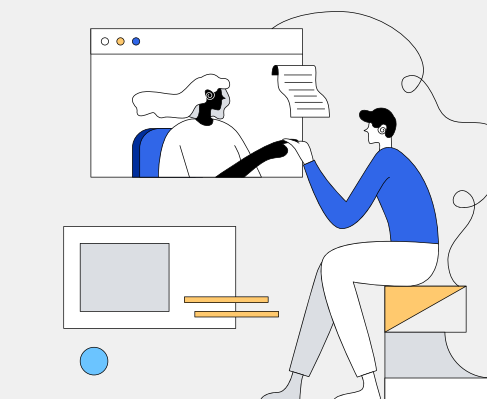
#### 我们的准则

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](mailto: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](mailto: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

1. 安装 openEuler
2. 开发环境准备
3. 下载和构建软件包

具体详情参见附录 1。

### 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](mailto: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
- 安全披露信息

安全处理流程和安全批露信息请参考附录 2。

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。



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。

附录

附录 1：搭建开发环境

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

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	<a href="https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-process.md">https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-process.md</a>
安全披露信息	<a href="https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-disclosure.md">https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-disclosure.md</a>