



openEuler

1.0

版本发行说明

发布日期 2020-01-21

目 录

法律声明.....	iii
1 openEuler 1.0 版本发行说明.....	4
2 简介.....	5
3 基础参数说明.....	6
4 技术特性.....	8
4.1 系统安装	8
4.2 镜像制作	9
4.3 文件系统	9
4.4 内核	10
4.4.1 调测 DFX 特性增强	10
4.4.2 系统性能优化	10
4.4.3 资源管控	11
4.4.4 鲲鹏芯片驱动支持或增强	11
4.5 系统管理	11
4.6 编译器	12
4.7 系统安全	13
4.8 系统调测	13
4.9 虚拟化	14
4.10 容器	15
4.11 支持鲲鹏生态	18

法律声明

版权所有 © 2020 华为技术有限公司。

您对“本文档”的复制、使用、修改及分发受知识共享(Creative Commons)署名-相同方式共享 4.0 国际公共许可协议(以下简称“CC BY-SA 4.0”)的约束。为了方便用户理解，您可以通过访问 <https://creativecommons.org/licenses/by-sa/4.0/> 了解 CC BY-SA 4.0 的概要 (但不是替代)。CC BY-SA 4.0 的完整协议内容您可以访问如下网址获取：
<https://creativecommons.org/licenses/by-sa/4.0/legalcode>。

商标声明

openEuler 为华为技术有限公司的商标。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

免责声明

本文档仅作为使用指导，除非适用法强制规定或者双方有明确书面约定，华为技术有限公司对本文档中的所有陈述、信息和建议不做任何明示或默示的声明或保证，包括但不限于不侵权，时效性或满足特定目的的担保。

1 openEuler 1.0 版本发行说明

openEuler 是一款开源操作系统。当前 openEuler 内核源于 Linux，支持鲲鹏及其它多种处理器，能够充分释放计算芯片的潜能，是由全球开源贡献者构建的高效、稳定、安全的开源操作系统，适用于数据库、大数据、云计算、人工智能等应用场景。同时，openEuler 是一个面向全球的操作系统开源社区，通过社区合作，打造创新平台，构建支持多处理器架构、统一和开放的操作系统，推动软硬件应用生态繁荣发展。

openEuler 1.0 是社区对外发布的第一个版本，本次发行说明提供了版本相关基础能力的介绍，主要包括：

- 简介
- 基础参数说明
- 技术特性

2 简介

openEuler 1.0 介绍

openEuler 1.0 版本是 openEuler 开源社区对外发布的第一个版本，是一款安全稳定的高性能 Linux 服务器操作系统。openEuler 1.0 基于 Linux 主线内核 4.19 版本，进行了 bug 修改、漏洞修复等工作，并加入一些具备创新性的功能特性。

最小硬件要求

安装 openEuler 1.0 所需的最小硬件要求如表 2-1 所示。

表2-1 最小硬件要求

部件名称	最小硬件要求	说明
CPU	鲲鹏 920（架构为 AArch64）	-
内存	不小于 8GB	-
硬盘	不小于 10GB（为了更好的应用体验，建议不小于 120GB）	支持 IDE、SATA、SAS 等接口的硬盘。

3 基础参数说明

openEuler 基础的规格参数，如下表所示：

类别	openEuler 1.0
核心参数	Kernel 4.19
	Glibc 2.28
	GCC 7.3.0
	systemd 243
标准符合度	符合 POSIX 标准
架构支持	AArch64
核心支持能力	支持的最大内存 AArch64: 2TB
	最大文件大小 ext4: 16TB
	文件系统默认 ext4 最大支持 1EB
	支持双核及多核处理器 支持并优化 NUMA 体系架构
安装	提供文本/图形安装界面
	提供光盘、网络引导安装方式
	支持 Kickstart 自动化安装
系统工具	提供软件包升级工具，支持 dnf/yum 软件包管理和 dnf/yum 升级
	提供防火墙的配置管理工具
虚拟化支持	提供对 KVM 的虚拟化支持（KVM 版本和内核一致）

类别	openEuler 1.0
	提供对 Qemu 4.0.0 和 Libvirt 5.5.0 的支持
	提供虚拟化管理工具实现单机环境下的虚拟机创建、配置与管理
容器支持	提供对 Docker 18.09 的支持
	提供对安全容器 Kata 1.7.0 的支持
	提供对轻量化容器引擎 iSulad 0.1.0 的支持
开发工具	支持 GCC 包含的 C、C++、Objective C、Fortran
	支持 Java、Go
	相应支持库（libstdc++、libgcj 等）
	支持 Python2/Python3、Perl、Shell 等脚本语言

4 技术特性

- 4.1 系统安装
- 4.2 镜像制作
- 4.3 文件系统
- 4.4 内核
- 4.5 系统管理
- 4.6 编译器
- 4.7 系统安全
- 4.8 系统调测
- 4.9 虚拟化
- 4.10 容器
- 4.11 支持鲲鹏生态

4.1 系统安装

安装程序

openEuler 1.0 改进了安装程序 `Anaconda`，并做了个性化定制和修改，方便用户安装。

- 提供 openEuler 主题的图形安装界面，对时区和语言进行了本地化优化。
- 支持直接格式化未分区的设备，支持在安装过程中配置临时文件存储设备 `tmpfs`，支持 `LVM` 自动精简配置。

引导程序 GRUB2

使用 `GRUB2` 作为启动引导程序，稳定方便。

- 支持 `UEFI` 规范
- 支持安全启动

4.2 镜像制作

openEuler 1.0 使用 Lorax 工具构建镜像。Lorax 包含了一系列工具，主要有以下两个：

- `livemedia-creator`：使用 Anaconda 创建 openEuler 的 live CD 镜像。
- `lorax`：使用 Anaconda 创建 openEuler 的安装镜像。

Lorax 使用指导请参见其[官方文档](#)。

4.3 文件系统

支持 ext4 文件系统

openEuler 1.0 中使用的默认文件系统是 ext4，ext4 向下兼容于 ext3/ext2 文件系统，可以将 ext3/ext2 挂载为 ext4 分区。

ext4 的主要特性：

- ext4 最大支持的文件系统是 1EB，单一最大文件值是 16TB。
- ext4 使用 extent 树代替了 ext3/ext2 的逻辑块映射，减少了元数据块的使用，提高了处理大型文件的效率。
- ext4 使用延迟块分配机制，仅当文件从高速缓存写入磁盘时才真正为数据分配磁盘块，有利于改善大文件的碎片化。
- ext4 支持日志校验和，以确认日志块是否遭到损坏，提升可靠性。
- ext4 支持惰性初始化，启用 `lazyinit` 特性后，会在后台挂载过程中逐步初始化 inode 表，这将显著加快创建 ext4 文件系统的速度。
- ext4 支持快速 `fsck`，在 ext4 中每个 inode 表将会存储未使用的 inode 列表，在 `fsck` 时将会跳过这些 inode，加速 `fsck`。
- ext4 支持三种日志模式：`journal`（可靠性最高）、`ordered`、`writeback`（性能最好）。

另外，openEuler 也对 ext4 进行了一些 bug 修复和优化，主要优化点如下：

- ext4 和 `sysmonitor` 相结合，支持 ext4 文件系统的故障告警通知。

支持 XFS 文件系统

openEuler 1.0 支持高性能的日志文件系统 XFS，并提供了专门用于 XFS 备份和恢复的工具 `xfsdump` 和 `xfstore`。XFS 需要关注的主要特性如下：

- XFS 的设计完全基于扩展，最大支持的文件系统和文件大小是 8EB - 1 字节。
- 使用 B+树的数据结构来提升 IO 灵活性，如将 B+树用于 inode 和 block 管理等。
- 将块设备分割成多个 AG（分配组），允许多个线程并行在不同的分配组上进行 IO 操作，提升并行能力。
- 支持延迟分配，在内存中记录事务，在数据真正写入磁盘之前分配磁盘块。

- 支持自描述 metadata 的元数据校验，防止在意外断电时元数据遭到损坏。

4.4 内核

openEuler kernel 基于 Linux 社区 kernel 4.19 版本，并在可靠性、安全性、虚拟化、性能等方面做了针对性的增强，更好地支持鲲鹏系列处理器。openEuler 基于该版本 kernel 的主要修改如下：

- 回合社区增强特性
- 回合 Linux 4.19 stable 补丁
- 修复 CVE 和其他 Bugfix
- 鲲鹏系列处理器适配和增强
- 泰山服务器驱动支持

4.4.1 调测 DFx 特性增强

- 支持 ARM64 内核热补丁
支持 ARM64 内核热补丁，可以不用重启修复内核漏洞和 BUG。
- ARM64 kdump 增强
openEuler 内核支持预留大于 4G 的内存供 kdump 使用

说明

Linux kernel 社区原生版本只支持预留 4G 以下内存。

- ARM64 NMI Watchdog 支持
 - 支持基于 PMU (Performance Monitoring Unit) 的 NMI Watchdog
 - 支持基于 SDEI (Software Delegated Exception Interface) 的 NMI Watchdog
- ARM64 RAS 增强
 - 支持 ARM v8.2 RAS 扩展
 - 支持执行路径上内存 ECC 错误恢复
- 支持 SAS 盘暴力热插拔
- 支持 Statistical Profiling Extension 解析增强
通过解析 SPE 事件，支持精确跟踪和记录 branch miss, LL cache miss, tlb miss 等事件，增强 perf 调优功能。

4.4.2 系统性能优化

系统扩展性 (Scale Out)

- Numa Aware Qspinlock
对现有的 qspinlock 做增强，将锁的等待队列分成两个，本地 NUMA 节点等待队列和远程 NUMA 节点等待队列，本地 NUMA 节点等待队列的线程有限获取锁，减少跨 NUMA 节点的 Cache/总线冲突，从而提升性能
- MMU gather

减少 TLB flush，从而提升性能

- 关键驱动如中断控制器 ITS 和 SMMU 的内存分配 NUMA Aware

IO 性能优化

- 通过优化 IOVA 页表查找和页表释放算法，提升 IO MMU 子系统性能
- 减少关键结构体的 cache false sharing，从而提升性能

指令加速

- CRC32 和 checksum 加速
根据 ARM64 指令以及流水线特点，优化 CRC32 及 checksum 实现，大幅提升数据校验性能。

4.4.3 资源管控

- Cache 与内存带宽的分配与监控
支持 ARM v8.4 MPAM (Memory System Resource Partitioning and Monitoring) 特性。
- 内存热添加
支持动态添加虚拟内存。
- CPU 热插拔
支持动态添加虚拟机 CPU。
- 支持限制 page cache 占用内存的比例
支持限制 page cache 占用内存的比例，避免系统因 Page Cache 过多而影响业务的性能或者 OOM。

4.4.4 鲲鹏芯片驱动支持或增强

支持鲲鹏处理器相关的驱动：

- 支持 HiSilicon SAS 驱动
- 支持 HNS 和 HNS3 板载网卡
- 支持 hinic 智能网卡

4.5 系统管理

openEuler 1.0 使用 systemd 进行系统和服务的管理，systemd 与 SysV 和 Linux 标准的 init 脚本兼容。systemd 具备下列优势：

- 更快的启动速度：systemd 提供了并行启动能力，采用了 socket/D-Bus activation 等技术启动服务。
- 按需启动守护进程：systemd 提供按需启动的能力，只有在某个服务被请求的时候才启动它。当该服务结束后，systemd 可以关闭它，等待下次需要时再次启动它。

- 采用 CGroup 来跟踪和管理进程的生命周期：systemd 则利用 Linux 内核的 CGroup 特性来完成跟踪的任务。停止服务时，通过查询 CGroup，systemd 可以确保找到所有的相关进程，从而干净地停止服务。
- 能够对系统进行快照和恢复：systemd 快照提供将当前系统的运行状态保存并恢复的能力。
- 日志服务：systemd 自带日志服务 journald，相比于 syslog 服务具备一些优点，比如：采用二进制格式保存所有日志信息，更安全；无需自己编写复杂脆弱的字符串分析处理程序。用户可以使用 journalctl 命令来查看日志信息。

openEuler 1.0 修复问题：

- 修复 udev 重命名冲突，加入等待和 retry 机制，确保命名成功。

4.6 编译器

GCC

GCC 7.3.0 包括了 C、C++、Fortran 的前端，也包括了这些语言的库（如 libstdc++、libgcc 等），openEuler 1.0 配套编译器为 GCC 7.3.0，并在此基础上进行了特性增强和漏洞修复，其主要更新如下：

特性增强

- 针对鲲鹏 920 的 TaishanV110 核，支持该 TaiShan 核微架构下的流水线优化
- 支持 AArch64 后端下的长跳转（Long Call）功能
- 支持循环优化 loop unroll and jam，将多层循环中的外层循环展开，并将展开后的多个内层循环合并
- 支持循环优化 loop interchange，将多层循环中的内外层循环进行交换

漏洞修复

修复如下 GCC 安全漏洞

- 修复 ARM 后端下栈保护功能在特殊场景中能够被黑客绕过保护的安全漏洞（CVE-2018-12886）
- 修复 POWER9 后端下多次调用 __builtin_darn 函数的返回值相同的安全漏洞（CVE-2019-15847）

OpenJDK

JDK（Java Development Kit）是 Java 开发者进行 Java 开发所必须的软件包，包含 JRE（Java Runtime Environment）和编译、调测工具。openEuler 1.0 配套 JDK 为 OpenJDK 8（兼容 Java SE 8 & JDK 8），并在此基础上进行了 GC 优化、并发稳定性增强、安全性增强等，提高了 Java 应用程序在 ARM 上的性能和稳定性。

4.7 系统安全

openEuler 1.0 提供多重安全手段，包括身份识别与认证、安全协议、强制访问控制、完整性保护、安全审计等安全机制，保障操作系统的安全性，为各类上层应用提供安全基础。

- 提供 openssh/pam/sss/krb5 等支持本地及远程集中身份识别与认证机制
- 提供 openssl-1.1.1 版本支持 TLSv1.3
- 采用主流 SELinux 安全框架，支持安全策略配置
- 提供基于 TPM 2.0 的 TPM 驱动及 TPM-TSS 协议栈，支持使能可信启动和安全启动
- 支持 AUDIT 审计框架，对系统进行安全审计

4.8 系统调测

支持 kdump

Kdump 是 Linux 内核态异常后的一种转储内存数据的机制。Kdump 服务启动时预先加载的 kdump 内核，通过这个小内核，重新挂载系统分区，通过 kdump 工具，将崩溃内核的内存以转储 vmcore 文件的方式保存到系统盘上，开发和维护工程师可以利用 crash 工具打开 vmcore，分析系统崩溃原因。

能够触发 kdump 包含：内核地址访问异常 (oops)、系统内核严重异常 (panic)、内核态死循环 (软件狗狗叫)、长时间关中断 (NMI 狗狗叫)、系统内存耗尽 (oom)。

特性增强

- 支持 AArch64 架构下同时使用高低端内存启动 kdump 服务
- 支持传递更长的启动参数给 kdump 内核 (由 512 字符增加到 1536 字符)
- 支持更大的物理内存地址空间 (100TB)

支持性能调测工具 (Perf)

Perf 是一个功能强大的性能分析工具，可以实现对程序硬件事件、软件事件、内核 tracepoint 及硬件断点等性能指标的监视，这些性能指标可以帮助程序员定位性能瓶颈，方便进行性能调优。

通过 Perf 工具，应用程序可以利用 PMU，tracepoint 和内核中的特殊计数器来进行性能统计。Perf 工具不但可以分析指定应用程序的性能问题，也可以用来分析内核的性能问题，可以同时分析应用代码和内核，从而全面理解应用程序中的性能瓶颈。

支持 blktrace 工具

blktrace 是一个针对 Linux 内核中块设备 I/O 层的跟踪工具，用来收集磁盘 IO 信息中当 IO 进行到块设备层时的详细信息 (如 IO 请求提交、入队、合并、完成等等一些列的信息)。

通过使用这个工具，使用者可以获取 I/O 请求队列的各种详细的情况，包括进行读写的进程名称、进程号、执行时间、读写的物理块号、块大小等等，是一个 Linux 下分析 I/O 相关内容的很好的工具。

4.9 虚拟化

Qemu

openEuler 1.0 中配套 Qemu 为 4.0.0 版本，相比早期版本修复了诸多缺陷并进行了特性和设备支持能力的增强，重要更新如下：

特性增强

- 热迁移 postcopy 支持 hugpages 和 hugetlbf
- 支持 GICv3 KVM 的热迁移
- 热迁移 postcopy 支持限速，支持 migrate_pause，支持 migrate_recover
- 虚拟机内呈现 CPU frequency（GuestOS 需要 4.19 以及更高版本内核）
- 虚拟机内部呈现 CPU topology 和 cache 信息

设备支持

- PCI/PCIe: 支持新的 pcie-root-port 设备，新的 virt 类型主板默认 disable 了 PCIe-to-PCI-bridge 上的 SHPC
- 支持 SMMUv3 模拟 (soft-iommu)
- virt 类型主板开始支持 512 虚拟 CPU（注意不支持单个虚拟机的 CPU 数量超过所在的物理机的 CPU 数量）
- 将 PCIe ECAM Region 扩大到 256M，支持更多的 PCI 设备
- virt4.0 guest machine type 支持超过 1TB RAM

性能优化

- I/O 线程性能优化，提升多磁盘并发读写场景下的性能

KVM

基于 4.19 版本内核的 KVM 进行 bug 修复、社区新特性回合和特性增强，重点能力如下：

虚拟机动态 IPA

openEuler 1.0 通过修改 Stage2 页表的分级、布局，支持用户态配置动态的 IPA 位数，可以支持最多 48 位 IPA

Perf 性能统计支持 ARM 虚拟化

支持命令“perf kvm stat record/report”，在虚拟机或 vCPU 粒度统计 exit、trap 的类型、次数和时间

IRQfd 路径注入中断优化

优化通过 `IRQfd` 路径注入的中断，降低 `vhost-user`、`vhost-net`、`GICv3+` 直通设备（网卡、磁盘）等场景下 CPU 资源消耗。

Open vSwitch

openEuler 1.0 中配套的 Open vSwitch 为开源社区的 2.11.1 版本，仅进行了 openEuler 的内核版本适配。

4.10 容器

openEuler 1.0 中提供容器运行的基础平台 `iSula`。`iSula` 基础容器平台同时提供 `Docker engine` 与轻量化容器引擎 `iSulad`，用户可根据需要自主选择。

同时，`iSula` 容器平台还支持运行 `kata-container`。`Docker`、`containerd`、`runc`、`kata-container` 均在社区版本的基础上进行了稳定性增强与 `bugfix`。

iSulad

`iSula` 在居住于中南美洲亚马逊丛林的巴西原住民眼里，它是一种非常强大的蚂蚁，学术上称为“子弹蚁”，因为被它咬一口，犹如被子弹打到那般疼痛，它是世界上最强大的昆虫之一。`iSulad` 作为轻量化的容器底座，可以为多种场景提供最灵活、最稳定、最安全的底层支撑，与子弹蚂蚁“小个头、大能量”的形象不谋而合。

`iSula` 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。轻量级容器具有轻、灵、巧、快的特点，不受硬件规格和架构的限制，底噪开销更小，可应用领域更为广泛。

iSulad 重点能力如下：

容器镜像管理

容器镜像是容器运行的基础，在主机上以分层的形式存放，运行前通过联合挂载的方式挂载成统一视图的 `rootfs` 供容器使用。容器的镜像管理包括镜像的拉取、导入、删除以及查询。

容器管理

容器管理功能提供用户创建、运行、停止、删除、重启等操作容器的能力。

CNI 网络

实现 `CRI` 接口对接 `CNI` 网络的能力，包括 `CNI` 网络配置文件的解析、`CNI` 网络的加入和退出。

CRI 接口

`CRI API` 接口是由 `kubernetes` 推出的容器运行时接口，`CRI` 定义了容器和镜像的服务接口。`iSulad` 支持 `CRI` 接口，实现和 `kubernetes` 的对接。

容器资源管理

对容器使用资源进行动态管理：

- namespace 共享
- 限制运行时的 CPU 资源
- 限制运行时的内存资源
- 限制运行时的 IO 资源
- 限制容器 rootfs 存储空间
- 限制容器内文件句柄数
- 限制容器内可以创建的进程/线程数
- 配置容器内的 ulimit 值

Docker

Docker 的英文本意是码头工人，码头工人的工作就是将商品打包到 Container（集装箱）并且搬运 Container、装载 Container。从 Docker 字面上的解释就可以看出 Docker 是干什么的，对应到 Linux 中，Docker 就是将 App 打包到 Container，通过 Container 实现 App 在各种平台上的部署，运行。Docker 通过 Linux Container 技术将 App 变成一个标准化的、可移植的、自管理的组件，实现了应用的 build once、run everywhere。Docker 技术特点是：应用快速发布、应用部署和扩容简单、更高的应用密度、应用管理更简单。

openEuler 1.0 集成 moby 18.09 版本，并在版本基础上进行了 bugfix 和稳定性增强。

安全容器

安全容器是虚拟化技术和容器技术的有机结合，相比普通 Linux 容器，安全容器具有更好的隔离性。

普通 Linux 容器利用 namespace 进行进程间运行环境的隔离，并使用 CGroup 进行资源限制；因此普通 Linux 容器本质上还是共用同一个内核，单个容器有意或无意影响到内核都会影响到整台宿主机上的容器。

安全容器是使用虚拟化层进行容器间的隔离，同一个主机上不同的容器间运行互相不受影响。

iSula 容器平台支持创建安全容器，使用 kata-container 作为容器的运行时。

openEuler 1.0 集成了 kata-container 1.7 版本，该版本主要特性如下：

- 在社区版本的基础上，进行了稳定性和可靠性加固
- 重构了社区 Host Cgroup 资源限制方案，支持 CPU、Memory 和 Blkio 资源限制，更适用于生产环境
- 更灵活的网络配置方案，支持通过 kata-network 接口创建和更新网卡和路由表
- 支持 DPDK/SPDK 高性能网络
- 支持容器网络状态信息监控

系统容器

系统容器主要应对在重计算、高性能、大并发的场景下，重型应用和业务云化的问题，相比较虚拟化技术，系统容器可直接继承物理机特性，同时具备性能更优良，较

少底噪的优点。从系统资源分配来看，系统容器在有限资源上相比虚拟机可分配更多计算单元，降低成本，通过系统容器可以构建产品的差异化竞争力，提供计算密度更高，价格更便宜，性能更优良的的计算单元实例。

iSula 容器平台支持创建系统容器，并能支持在系统容器内动态调整设备、运行资源，且提供更优秀的 user namespace 隔离。

系统容器相比于普通容器，提供了如下增强特性：

本地文件系统启动

普通容器只能通过远端 pull 镜像到本地启动，pull 镜像过程可能会因网络问题导致耗时较长，而系统容器支持本地文件系统启动，可实现快速部署

容器内运行 systemd

系统容器与普通容器最大的差异就在于容器的 init 进程启动，普通容器无法实现容器启动时自动拉起 systemd，而系统容器具备这个能力

增强隔离性

- 支持使用 lxcfs 插件实现容器内资源视图隔离，同时支持 lxcfs 插件热插拔和高可用
- 支持多对应 user namespace，实现容器和主机 uid/gid 隔离
- 支持配置容器内 namespace 化内核参数可读写，在容器内可动态配置 namespace 化内核参数值

控制容器访问权限

支持通过 TLS 认证方式来对用户身份进行验证，并通过对接 authz 插件服务控制用户的操作权限

容器资源动态管理

- 动态添加/删除/更新容器内的设备
- 动态添加/删除/更新容器内的网卡
- 动态添加/删除容器内的路由
- 动态添加/删除容器内的卷

其它特性列表

- 环境变量持久化
- 限制容器内句柄数
- 执行 reboot/shutdown 命令
- 动态加载内核模块
- 共享内存通信

4.11 支持鲲鹏生态

openEuler 1.0 与鲲鹏处理器协同设计开发，提供 OS、虚拟化、容器、工具链等基础软件栈能力，打造高安全、高性能、鲲鹏原生等核心 OS 基础设施竞争力。

- 多核/重核加速技术：提升鲲鹏芯片多核计算性能。
- 芯片级加速技术：HPRE 和 SEC 加速器的软件全栈支持，大幅提升加解密运算、压缩解压缩运算的性能。