openEuler 24.03 LTS Technical White Paper

Contents

1 Introduction

The openEuler open source community is incubated and operated by the OpenAtom Foundation.

openEuler is a digital infrastructure OS that fits into any server, cloud computing, edge computing, and embedded deployment. This secure, stable, and easy-to-use open source OS is compatible with multiple computing architectures. openEuler suits operational technology (OT) applications and enables the convergence of OT and information and communications technology (ICT).

The openEuler open source community is a portal available to global developers, with the goal of building an open, diversified, and architecture-inclusive software ecosystem for all digital infrastructure scenarios. It has a rich history of helping enterprises develop their software, hardware, and applications.

The openEuler open source community was officially established on December 31, 2019, with the original focus of innovating diversified computing architectures.

On March 30, 2020, the Long Term Support (LTS) version openEuler 20.03 was officially released, which was a new Linux distribution with independent technology evolution.

Later in 2020, on September 30, the innovative openEuler 20.09 version was released through the collaboration efforts of multiple companies, teams, and independent developers in the openEuler community. The release of openEuler 20.09 marked a milestone not only in the growth of the openEuler community, but also in the history of open sourced software in China.

On March 31, 2021, the innovative kernel version openEuler 21.03 was released. This version is enhanced in line with Linux kernel 5.10 and also incorporates multiple new features, such as live kernel upgrade and tiered memory expansion. These features improve multi-core performance and deliver the computing power of one thousand cores.

Fast forward to September 30, 2021, openEuler 21.09 was released. This premium version is designed to supercharge all scenarios, including edge and embedded devices. It enhances server and cloud computing features, and incorporates key technologies including cloud-native CPU scheduling algorithms for hybrid service deployments and KubeOS for containers.

On March 30, 2022, openEuler 22.03 LTS was released based on Linux kernel 5.10. Designed to meet all server, cloud, edge computing, and embedded workloads, openEuler 22.03 LTS is an all-scenario digital infrastructure OS that unleashes premium computing power and resource utilization.

On September 30, 2022, openEuler 22.09 was released to further enhance all-scenario innovations.

On December 30, 2022, openEuler 22.03 LTS SP1 was released, which is designed for hitless porting with best-of-breed tools.

On March 30, 2023, openEuler 23.03 was released. Running on Linux kernel 6.1, it streamlines technical readiness for Linux kernel 6.x and facilitates innovations for hardware adaptation and other technologies.

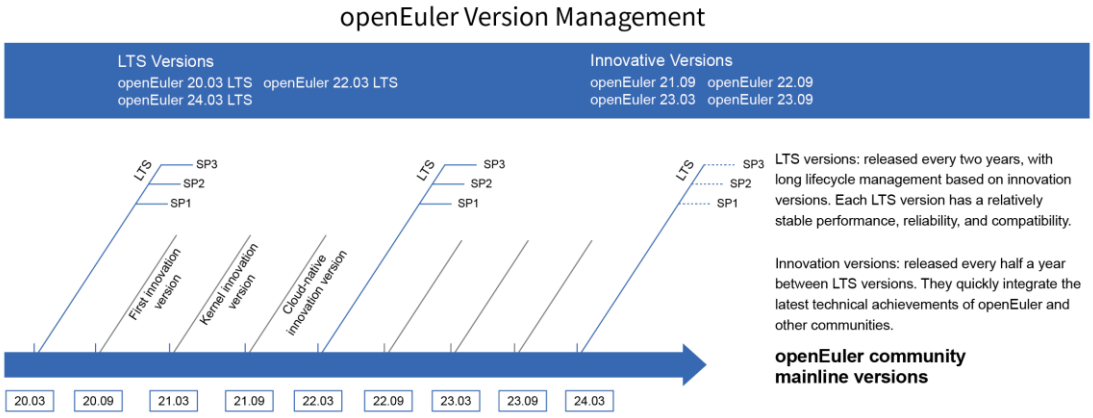
On June 30, 2023, openEuler 22.03 LTS SP2 was released, which enhances scenario-specific features and increases system performance to a higher level.

Later on September 30, 2023, the openEuler 23.09 innovation version was released based on Linux kernel 6.4. It provides a series of brand-new features to further enhance developer and user experience.

On November 30, 2023, openEuler 20.03 LTS SP4 was released, an enhanced extension of openEuler 20.03 LTS. openEuler 20.03 LTS SP4 provides excellent support for server, cloud native, and edge computing scenarios.

On December 30, 2023, openEuler 22.03 LTS SP3 was released. Designed to improve developer efficiency, it extends features for server, cloud native, edge computing, and embedded scenarios.

On May 30, 2024, openEuler 24.03 LTS was released. This version built on Linux kernel 6.6 brings new features for server, cloud, edge computing, AI, and embedded scenarios to enhance developer and user experience.

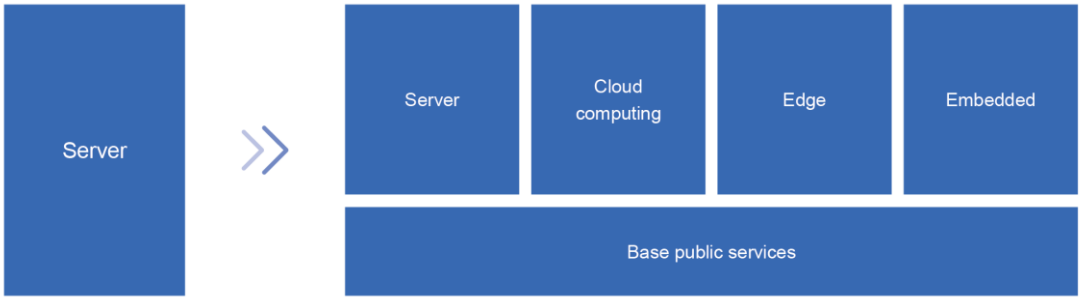


As an OS platform, openEuler releases an LTS version every two years. Each LTS version provides enhanced specifications and a secure, stable, and reliable OS for enterprise users.

openEuler is built on tried-and-tested technologies. A new openEuler innovative version is released every 6 months to quickly integrate the latest technical achievements of openEuler and other communities. The innovative tech is first verified in the openEuler open source community as a single open source project, and then these features are added to each new release, enabling community developers to obtain the source code.

Technical capabilities are first tested in the open source community, and continuously incorporated into each openEuler release. In addition, each release is built on feedback given by community users to bridge the gap between innovation and the community, as well as improve existing technologies. openEuler is both a release platform and incubator of new technologies, working in a symbiotic relationship that drives the evolution of new versions.

Innovative Platform for All Scenarios



openEuler supports multiple processor architectures (x86, Arm, SW64, RISC-V, and LoongArch) and will support other brands (such as PowerPC) in the future, as part of a focus to continuously improve the ecosystem of diversified computing power.

The openEuler community is home to an increasing number of special interest groups (SIGs), which are dedicated teams that help extend the OS features from server to cloud computing, edge computing, and embedded scenarios. openEuler is built for use in any scenario, and comprises openEuler Edge and openEuler Embedded that are designed for edge computing and embedded deployments, respectively.

The OS is a perfect choice for ecosystem partners, users, and developers who plan to enhance scenario-specific capabilities. By creating a unified OS that supports multiple devices, openEuler hopes to enable a single application development for all scenarios.

Open and Transparent: The Open Source Software Supply Chain

The process of building an open source OS relies on supply chain aggregation and optimization. To ensure reliable open source software or a large-scale commercial OS, openEuler comprises a complete lifecycle management that covers building, verification, and distribution. The brand regularly reviews its software dependencies based on user scenarios, organizes the upstream community addresses of all the software packages, and verifies its source code by comparing it to that of the upstream communities. The build, runtime dependencies, and upstream communities of the open source software form a closed loop, realizing a complete, transparent software supply chain management.

2 Platform Architecture

System Framework

openEuler is an innovative open source OS platform built on kernel innovations and a solid cloud base to cover all scenarios. It is built on the latest trends of interconnect buses and storage media, and offers a distributed, real-time acceleration engine and base services. It provides competitive advantages in edge and embedded scenarios, and is the first step to building an all-scenario digital infrastructure OS.

openEuler 24.03 LTS runs on Linux kernel 6.6 and provides POSIX-compliant APIs and OS releases for server, cloud native, edge, and embedded environments. It is a solid foundation for intelligent collaboration across hybrid and heterogeneous deployments. openEuler 24.03 LTS is equipped with a distributed soft bus and KubeEdge+ edge-cloud collaboration framework, among other premium features, making it a perfect choice for collaboration over digital infrastructure and everything connected models.

In the future, the openEuler open source community will continue to innovate, aiming to promote the ecosystem and consolidate the digital infrastructure.

Cloud base:

- **KubeOS for containers**: In cloud native scenarios, the OS is deployed and maintained in containers, allowing the OS to be managed based on Kubernetes, just as service containers.
- **Secure container solution**: Compared with the traditional Docker+QEMU solution, the iSulad+shimv2+StratoVirt secure container solution reduces the memory overhead and boot time by 40%.
- **Dual-plane deployment tool eggo**: OSs can be installed with one click for Arm and x86 hybrid clusters, while deployment of a 100-node cluster is possible within just 15 minutes.

New scenarios:

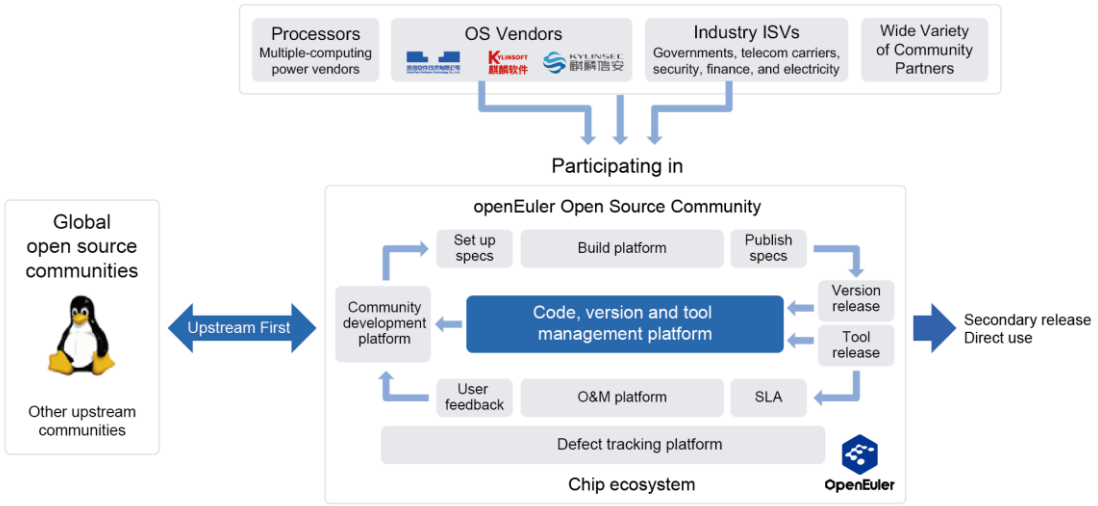
- **Edge computing**: openEuler 24.03 LTS Edge is released for edge computing scenarios. It integrates the KubeEdge+ edge-cloud collaboration framework to provide unified management, provisioning of edge and cloud applications, and other capabilities.
- **Embedded**: openEuler 24.03 LTS Embedded is released for embedded scenarios, helping compress images to under 5 MB and shorten image loading time to under 5 seconds.

Flourishing community ecosystem:

- **Desktop environments**: UKUI, DDE, Xfce, Kiran-desktop, and GNOME.
- **openEuler DevKit**: Supports OS migration, compatibility assessment, and various development tools such as secPaver which simplifies security configuration.

Platform Framework

The openEuler open source community partners with upstream and downstream communities to advance the evolution of openEuler versions.



Hardware Support

The openEuler open source community works with multiple vendors to build a vibrant southbound ecosystem. With participation of major chip vendors including Intel and AMD, all openEuler versions support x86, Arm, ShenWei, Loongson, and RISC-V CPU architectures, and a wide range of CPU chips, such as Loongson 3 series, Zhaoxin KaiXian and KaiSheng, Intel Ice Lake and Sapphire Rapids, and AMD EPYC Milan and Genoa. openEuler can run on servers from multiple hardware vendors and is compatible with NIC, RAID, Fibre Channel, GPU & AI, DPU, SSD, and security cards.

openEuler supports the following CPU architectures:

Hardware Type	x86	Arm	LoongArch	SW64	RISC-V
CPU	Intel, AMD, Zhaoxin, Hygon	Kunpeng, Phytium	Loongson	ShenWei	Sophgo, T-Head

openEuler supports the following servers:

Hardware Type	x86	Arm
Server	• **Intel**: xFusion, Supermicro	• **Kunpeng**: TaiShan
	• **AMD**: Supermicro, H3C	• **Phytium**: QS, PowerLeader, H3C
	• **Hygon**: Sugon/Suma	
	• **Zhaoxin**: Zhaoxin	

openEuler supports the following cards:

Hardware Type	x86	Arm
NIC	Huawei, Mellanox, Intel, Broadcom, 3SNIC, NetSwift, Yunsilicon, Mucse	Huawei, Mellanox, Intel, Broadcom, 3SNIC, NetSwift, Yunsilicon, Mucse
RAID	Avago, 3SNIC, PMC, Huawei	Avago, 3SNIC, PMC, Huawei
Fibre Channel	Marvell, Qlogic, Emulex	Marvell, Qlogic, Emulex
GPU & AI	NVIDIA	NVIDIA
SSD	Huawei	Huawei

For a full hardware list, visit https://www.openeuler.org/en/compatibility/.

3 Operating Environments

Servers

To install openEuler on a physical machine, check that the physical machine meets the compatibility and hardware requirements.

For a full list, visit https://openeuler.org/en/compatibility/.

Item	Configuration Requirement
Architecture	AArch64, x86_64
Memory	At least 4 GB
Drive	At least 20 GB

VMs

Verify VM compatibility when installing openEuler.

Hosts running on openEuler 24.03 LTS support the following software package versions:

- libvirt-9.10.0-9.oe2403
- libvirt-client-9.10.0-9.oe2403
- libvirt-daemon-9.10.0-9.oe2403
- qemu-8.2.0-12.oe2403
- qemu-img-8.2.0-12.oe2403

openEuler 24.03 LTS is compatible with the following guest OSs for VMs:

Host OS	Guest OS	Architecture
openEuler 24.03 LTS	CentOS 6	x86_64
openEuler 24.03 LTS	CentOS 7	AArch64
openEuler 24.03 LTS	CentOS 7	x86_64
openEuler 24.03 LTS	CentOS 8	AArch64
openEuler 24.03 LTS	CentOS 8	x86_64
openEuler 24.03 LTS	Windows Server 2016	AArch64
openEuler 24.03 LTS	Windows Server 2016	x86_64
openEuler 24.03 LTS	Windows Server 2019	AArch64
openEuler 24.03 LTS	Windows Server 2019	x86_64

Item	Configuration Requirement
Architecture	AArch64, x86_64
CPU	≥ 2 CPUs
Memory	≥ 4 GB
Drive	≥ 20 GB

Edge Devices

To install openEuler on an edge device, check that the edge device meets the compatibility and minimum hardware requirements.

Item	Configuration Requirement
Architecture	AArch64, x86_64
Memory	≥ 4 GB
Drive	≥ 20 GB

Embedded Devices

To install openEuler on an embedded device, check that the embedded device meets the compatibility and minimum hardware requirements.

Item	Configuration Requirement
Architecture	AArch64, AArch32
Memory	≥ 512 MB
Drive	≥ 256 MB

4 Scenario-specific Innovations

AI

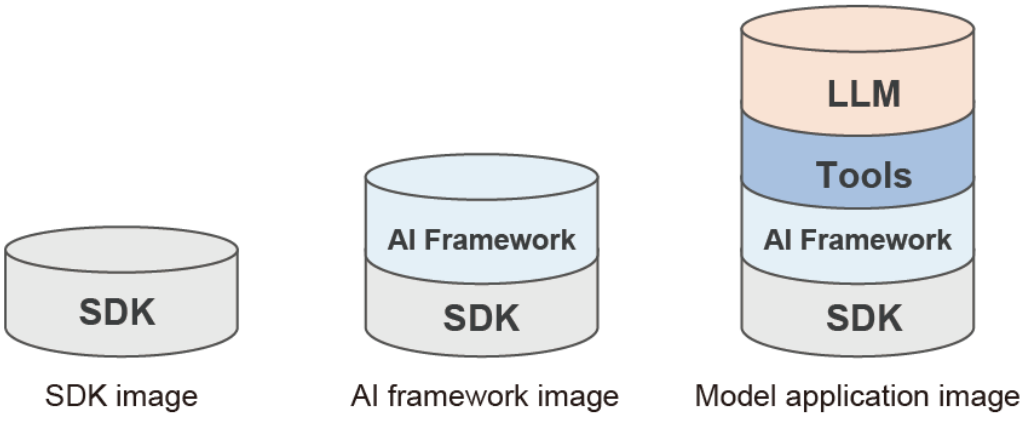
AI is redefining OSs by powering intelligent development, deployment, and O&M. openEuler supports general-purpose architectures like Arm, x86, and RISC-V, and next-gen AI processors like NVIDIA and Ascend. Further, openEuler is equipped with extensive AI capabilities that have made it a preferred choice for diversified computing power.

OS for AI

openEuler offers an efficient development and runtime environment that containerizes software stacks of AI platforms with out-of-the-box availability. It also provides various AI frameworks to facilitate AI development.

Feature Description

1. openEuler supports TensorFlow and PyTorch frameworks and software development kits (SDKs) of major computing architectures, such as Compute Architecture for Neural Networks (CANN) and Compute Unified Architecture (CUDA), to make it easy to develop and run AI applications.
2. Environment setup is further simplified by containerizing software stacks. openEuler provides three types of container images:



- **SDK images**: Use openEuler as the base image and install the SDK of a computing architecture, for example, Ascend CANN and NVIDIA CUDA.
- **AI framework images**: Use the SDK image as the base and install AI framework software, such as PyTorch and TensorFlow.
- **Model application images**: Provide a complete set of toolchains and model applications.

For details, see the openEuler AI Container Image User Guide.

Application Scenarios

openEuler uses AI container images to simplify deployment of runtime environments. You can select the container image that best suits your requirements and complete the deployment in a few simple steps.

- **SDK images**: You can develop and debug Ascend CANN or NVIDIA CUDA applications using an SDK image, which provides a compute acceleration toolkit and a development environment. These containers offer an easy way to perform high-performance computing (HPC) tasks, such as large-scale data processing and parallel computing.
- **AI framework images**: This type of containers is designed to support AI model development, training, and inference.
- **Model application images**: Such an image contains a complete AI software stack and purpose-built models for model inference and fine-tuning.

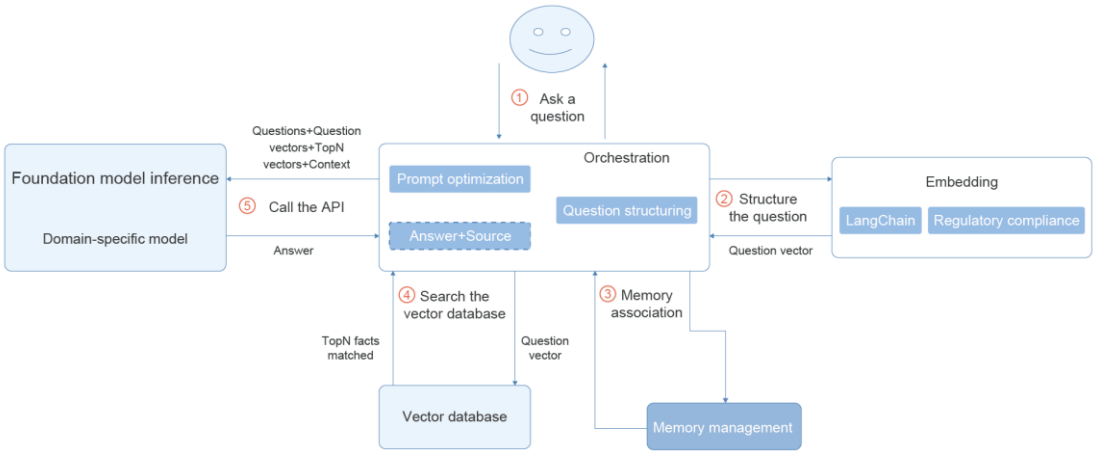
AI for OS

AI makes openEuler more intelligent. EulerCopilot, an intelligent Q&A platform, is developed using foundation models and openEuler data. It assists in code generation, problem analysis, and system O&M.

Feature Description

EulerCopilot is accessible via web or shell.

- **Web**: Provides basic OS knowledge, openEuler data, O&M methods, and project introduction and usage guidance.
- **Shell**: Delivers user-friendly experience using natural languages.



Application Scenarios

- **Common users**: Best practices for openEuler operations, such as porting applications to openEuler.
- **Developers**: Extensive knowledge of openEuler contribution processes, key features, and project development.
- **O&M personnel**: Efficient system management to quickly resolve common problems.

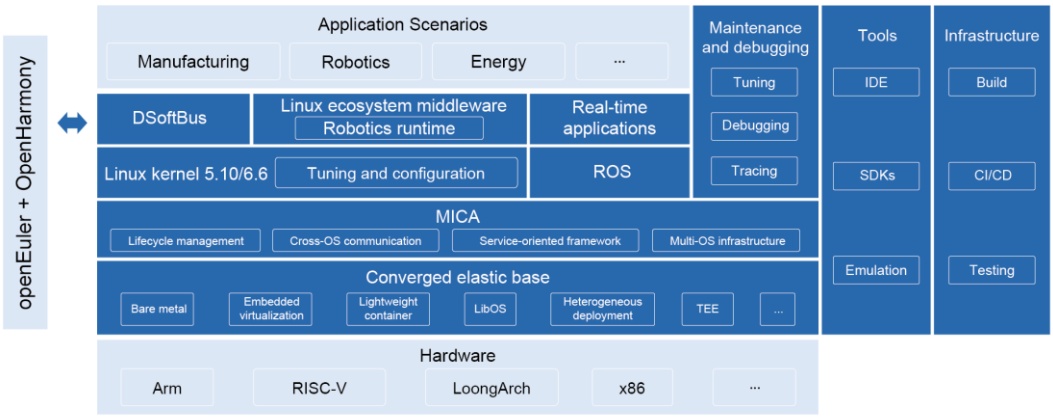
openEuler Embedded

openEuler 24.03 LTS Embedded brings new enhancements to the integrated embedded system software platform. It has made significant progress in southbound and northbound ecosystems, technical features, infrastructure, and implementation solutions.

openEuler Embedded helps build operational technology (OT) applications such as manufacturing and robotics in a closed loop design, whereby innovations help optimize its embedded system software stack and ecosystem. openEuler Embedded supports Linux kernels 5.10 and 6.6. Users can select either of the Linux kernels or customize the kernel to better fulfill their board support package (BSP) and application requirements. openEuler 24.03 LTS Embedded is equipped with an embedded virtualization base, which is available as multiple solutions, including Jailhouse partitioning-based virtualization, openAMP bare metal hybrid deployment, and Zephyr-based Virtual Machine (ZVM) & Rust-Shyper real-time virtualization. You can select the most appropriate solution to suit your services. The mixed-criticality (MICA) deployment framework is built on the embedded virtualization base to mask the differences between different bases and provides a unified set of APIs for different runtimes. The MICA deployment framework supports over 600 northbound software packages, including the ROS Humble version. ROS Humble contains ros-core, ros-base, and SLAM software packages to implement the ROS 2 runtime. SDKs with embedded features of ROS 2 are provided for ROS 2 colcon cross-compilation. openEuler 24.03 LTS Embedded is already included in the BMC ecosystem and has been adapted to openBMC. The MICA deployment framework supports a variety of southbound hardware, including Phytium, HiSilicon, Renesas, TI, and Allwinner, and also EulerPi (hardware development board designed for developers) and HiEuler Pi (native development board for openEuler Embedded). Infrastructure tools available on openEuler 24.03 LTS Embedded include the meta-tool oebuild and build tool Yocto 4.0 LTS. In addition, the LLVM toolchain is introduced to help better build images and generate SDKs. Compared with the GCC toolchain, LLVM has its advantages in performance, size, and security. openEuler Embedded has multiple commercial and enterprise editions, which have been applied in BMC, industrial controller, robot controller, and other fields.

Future versions will utilize contributions from ecosystem partners, users, and community developers, to increase support for chip architectures such as LoongArch and more southbound hardware, and optimize capabilities such as industrial middleware, embedded AI, embedded edge, and simulation systems.

System Architecture



Southbound Ecosystem

openEuler Embedded Linux currently supports AArch64, x86_64, AArch32, and RISC-V chip architectures, and will add support for LoongArch. openEuler 24.03 LTS has a vibrant southbound ecosystem and supports chips from Raspberry Pi, HiSilicon, Rockchip, Renesas, TI, Phytium, StarFive, and Allwinner.

Embedded Virtualization Base

openEuler Embedded uses an elastic virtualization base that enables multiple OSs to run on a system-on-a-chip (SoC). The base incorporates a series of technologies including bare metal, embedded virtualization, lightweight containers, LibOS, trusted execution environment (TEE), and heterogeneous deployment.

- The bare metal hybrid deployment solution runs on OpenAMP to manage peripherals by partition at a high performance level; however, it delivers poor isolation and flexibility. This solution supports the hybrid deployment of UniProton/Zephyr/RT-Thread and openEuler Embedded Linux.
- Partitioning-based virtualization is an industrial-grade hardware partition virtualization solution that runs on Jailhouse. It offers superior performance and isolation but inferior flexibility. This solution supports the hybrid deployment of UniProton/Zephyr/FreeRTOS and openEuler Embedded Linux or of OpenHarmony and openEuler Embedded Linux.
- Real-time virtualization works with the two hypervisor options, ZVM (for real-time VM monitoring) and Rust-Shyper (for Type-I embedded VM monitoring), which are incubated in the openEuler community.

MICA Deployment Framework

The MICA deployment framework is a unified environment that masks the differences between technologies that comprise the embedded elastic virtualization base. The multi-core capability of hardware combines the universal Linux OS and a dedicated real-time operating system (RTOS) to make full use of all OSs.

The MICA deployment framework covers lifecycle management, cross-OS communication, service-oriented framework, and multi-OS infrastructure.

- Lifecycle management provides operations to load, start, suspend, and stop the client OS.
- Cross-OS communication uses a set of communication mechanisms between different OSs based on shared memory.
- Service-oriented framework enables different OSs to provide their own services. For example, Linux provides common file system and network services, and the RTOS provides real-time control and computing.
- Multi-OS infrastructure integrates OSs through a series of mechanisms, covering resource expression and allocation and unified build.

The MICA deployment framework provides the following functions:

- Lifecycle management and cross-OS communication for openEuler Embedded Linux and the RTOS (Zephyr or UniProton) in bare metal mode
- Lifecycle management and cross-OS communication for openEuler Embedded Linux and the RTOS (FreeRTOS) in partitioning-based virtualization mode

Northbound Ecosystem

- **Northbound software packages**: Over 600 common embedded software packages can be built using openEuler.
- **ROS runtime**: openEuler 24.03 LTS Embedded supports the ROS 2 Humble version, which contains core software packages such as ros-core, ros-base, and SLAM. It also provides the ROS SDK that simplifies embedded ROS development.
- **Soft real-time kernel**: This capability helps respond to soft real-time interrupts within microseconds.
- **DSoftBus**: The distributed soft bus system (DSoftBus) of openEuler 24.03 LTS Embedded integrates the DSoftBus and point-to-point authentication module of OpenHarmony. It implements interconnection between openEuler-based embedded devices and OpenHarmony-based devices as well as between openEuler-based embedded devices.
- **Embedded containers and edges**: With iSula containers, openEuler and other OS containers can be deployed on embedded devices to simplify application porting and deployment. Embedded container images can be compressed to 5 MB, and can be easily deployed into the OS on another container. openEuler 24.03 LTS Embedded supports KubeEdge to streamline cloud-edge-device collaboration.

UniProton

UniProton is an RTOS that features ultra-low latency and flexible MICA deployments. It is suited for industrial control because it supports both microcontroller units and multi-core CPUs. UniProton provides the following capabilities:

- Compatible with CPU architectures like Cortex-M, AArch64, x86_64, and riscv64, and supports M4, RK3568, RK3588, x86_64, Hi3093, Raspberry Pi 4B, Kunpeng 920, Ascend 310, and Allwinner D1s.
- Connects with openEuler Embedded Linux on Raspberry Pi 4B, Hi3093, RK3588, and x86_64 devices in bare metal mode.
- Can be debugged using GDB on openEuler Embedded Linux.
- Over 890 POSIX APIs available for file systems, device management, shell consoles, and networks.

Application Scenarios

openEuler Embedded helps supercharge computing performance in a wide range of industries and fields, including industrial and power control, robotics, aerospace, automobiles, and healthcare.

5 Kernel Innovations

openEuler 24.03 LTS runs on Linux kernel 6.6 and inherits the competitive advantages of community versions and innovative features released in the openEuler community.

Inherited Upstream Features

- Folio-based memory management: The Linux memory management unit is changed from page to folio (from Latin foliō). A folio consists of one or more pages and is declared in struct folio. In folio-based memory management, operations are performed on one or more entire pages, rather than on only PAGE_SIZE bytes. In this way, compound page conversion is alleviated and tail page misuses are reduced. In terms of memory management efficiency, the folio-based mechanism can decrease the number of least recently used (LRU) linked lists and optimize memory reclamation. In addition, more continuous memory is allocated at a time to reduce the number of page faults and mitigate memory fragmentation. With respect to I/Os, this feature accelerates large I/Os and improves throughput. Large folios consisting of anonymous pages or file pages are available, and a system-level switch is provided for toggling this feature as required. In the AArch64 architecture, a contiguous bit (16 contiguous page table entries, or PTEs, cached in a single entry in a translation lookaside buffer, or TLB) is provided to reduce system TLB misses, thereby improving the overall system performance.
- EEVDF scheduling: Short for earliest eligible virtual deadline first, the EEVDF scheduler takes the scheduling latency into consideration. During task scheduling, it ensures fair allocation of CPU time to tasks, and preferentially schedules tasks with the closest deadlines from the tasks that are not allocated with enough CPU time, thereby meeting latency requirements. This solves the problem that the original Completely Fair Scheduler (CFS) can only ensure fair allocation of CPU time to tasks.
- Cgroup v2: Compared with cgroup v1, cgroup v2 has a unified hierarchy, comprehensive thread mode management, secure subtree delegation, and diverse features.
    1. Unified hierarchy

       The hierarchical management of cgroups is simplified. Users do not need to configure multiple independent cgroup trees for managing different resources, simplifying the collaboration of multiple controllers. Consistent and simplified interfaces are provided to streamline the configuration. In cgroup v2, a restriction has been implemented that prevents child cgroup controllers from being enabled when there are processes running in the parent cgroup. This brings higher security and avoids resource contention between parent and child cgroups.
    2. Comprehensive thread mode management

       Cgroup v2 introduces the thread mode to restrict the subsystems that can be threaded, that is, managed as threads. A thread can be allocated to a cgroup different from other threads in the same process for independent resource usage control.
    3. Secure subtree delegation

       The delegation mechanism allows non-privileged users to create and manage their own cgroup hierarchies. Through proper delegation, the system administrator can provide users or applications with necessary control permissions while ensuring fine-grained resource management and high system stability and security.

4. Diverse features

   Based on unified file tree management, cgroup v2 supports multiple features, including PSI, page cache writeback, enhanced allocation and isolation across resources, unified accounting for different types of memory allocation, and Memory QoS.

- Maple trees and per-VMA locks: Maple trees are used to replace the red-black trees to manage process address space. In addition, RCU-friendly design and per-VMA locking are used to reduce lock contention, improve page fault handling scalability, and accelerate performance in concurrency scenarios such as application startup.
- Per-CPU Pageset (PCP) high auto-tuning: The performance requirements for page allocation and release vary according to workloads. The PCP high auto-tuning feature can automatically adjust the pageset size of each CPU to optimize page allocation and release and improve performance in concurrency scenarios such as simultaneous kernel building.
- Multi-gen LRU: This mechanism accurately identifies hot and cold data in pages, improves system performance in high memory pressure scenarios, and reduces the OOM probability.
- Data Access MONitor (DAMON): DAMON allows lightweight memory access monitoring. It can accurately monitor accesses to virtual and physical addresses in user space online, assisting in performance improvement.
- Memory tiering: This feature aims to meet memory usage requirements in an efficient and cost-effective manner.
- HugeTLB vmemmap: The HugeTLB vmemmap feature in the AArch64 architecture is provided to reduce the footprint of memory management structures.
- Huge vmalloc: For vmalloc/vmap operations that allocate memory exceeding the minimum huge page size, huge pages instead of base pages are preferentially used to reduce TLB misses and improve vmalloc performance.
- memfd_secret system call: This interface allows userspace processes to create secret memory regions that cannot be accessed by anyone else (including the kernel). These regions can be used to, for example, store private keys, which reduces the possibility of sensitive data exposure in system memory.
- BIG TCP: This feature allows the protocol stack to send bigger TSO/GRO packets, achieving a higher network throughput and lower latency.
- XDP multi-buffer: XDP can be used to improve performance in jumbo frame scenarios.
- Thread-based NAPI polling: NIC NAPI polling can be done by kernel threads, so that the CPU scheduler can properly schedule resources to improve performance.
- BPF kfuncs: BPF can use symbols to directly call the functions provided by the kernel and KO modules. KO modules register interfaces as kfuncs to dynamically expose them to BPF.
- BPF dynamic pointers: In earlier versions, the memory used by BPF must be statically specified during the verifier check. Now, BPF can reference dynamically allocated memory.
- Perf features: obtaining Arm SPE events, directly reading PMU counters in userspace, displaying lock contention status, and shortening the average processing time by trimming PMU data.

openEuler's Key Contributions

- Large folio support for ext4:
  1. The writeback process of the iomap framework supports batch block mapping.
  2. Blocks can be requested in batches in default ext4 mode, greatly optimizing ext4 performance in various benchmarks.
  3. In ext4 buffer I/O and page cache writeback, the old buffer_head framework is replaced with the iomap framework which also adds large folio support for ext4.

- Tidal affinity scheduling: An efficient memory reclamation and loading mechanism is provided for per-memcg swap device isolation. Cold data is reclaimed when the service load is light, and data is quickly loaded when the service load increases. In this way, the available memory space and service performance are increased, achieving better QoS at the same memory cost.
- The system dynamically adjusts CPU affinity based on the service load. When the service load is light, the system uses preferred CPUs to enhance resource locality. When the service load is heavy, the system schedules more CPU cores for the service to improve the QoS.
- Memory System Resource Partitioning and Monitoring (MPAM): MPAM solves system-wide or application-specific performance deterioration due to contention for shared resources in a server system that runs diverse types of services concurrently. It traces the usage of shared resources in real time by CPU or PID. The reconstructed MPAM in Linux kernel 6.6 has the following new features:
  1. L2 cache partition isolation and monitoring. The latter one provides cache usage statistics and cache bandwidth traffic monitoring.
  2. Dynamic adjustment of shared resource configurations by task priority.
  3. Shared resource limits.
- CPU QoS priority-based load balancing: CPU QoS isolation is enhanced in online and offline hybrid deployments, and QoS load balancing across CPUs is supported to further reduce QoS interference from offline services.
- Simultaneous multithreading (SMT) expeller free of priority inversion: This feature resolves the priority inversion problem in the SMT expeller feature and reduces the impact of offline tasks on the QoS of online tasks.
- Multiple priorities in a hybrid deployment: Each cgroup can have a cpu.qos_level that ranges from -2 to 2. Users can set qos_level_weight to assign a different weight to each priority and allocate CPU resources to each priority based on the CPU usage. This feature is also capable of wakeup preemption.
- Programmable scheduling: The programmable scheduling framework based on eBPF allows the kernel scheduler to dynamically expand scheduling policies to meet performance requirements of different loads.
- AArch64 vCPU hotplug: This feature enables hotplug of VM CPUs without affecting VM running and dynamic adjustment of VM computing capabilities. Secure and reliable dynamic VM scaling is also provided.
- Adaptive provisioning of computing power: To ensure responsiveness and reliability of interactive applications (such as cloud desktops) running on multi-core servers, computing power is dynamically provisioned based on load changes. When the service load is heavy, a lightweight task search algorithm triggers idle CPUs to execute tasks more quickly. This mechanism implements fast load balancing between cores and maximizes CPU utilization. In the event of resource contention, services are processed by priority and computing power is provisioned preferentially to foreground tasks with high priorities over background tasks with low priorities.
- Write control for mounted file system devices: Write control after block devices (partition devices) are mounted to a file system is provided to generate alarms for risky write operations and prevent damage to the file system.
- RISC-V BPF features: BPF capabilities are extended with new features including trampoline, Zbb, kfuncs, CPU v4 instructions, and atomic operation instructions.
- AArch64 BPF features: BPF capabilities are extended with new features including ldr and str optimizations in the BPF stack, PAC, trampoline, and CPU v4 instructions.
- Tiered-reliability memory (inherited): This feature allows users to allocate memory with corresponding reliability as required and mitigates the impact of some possible UCEs or

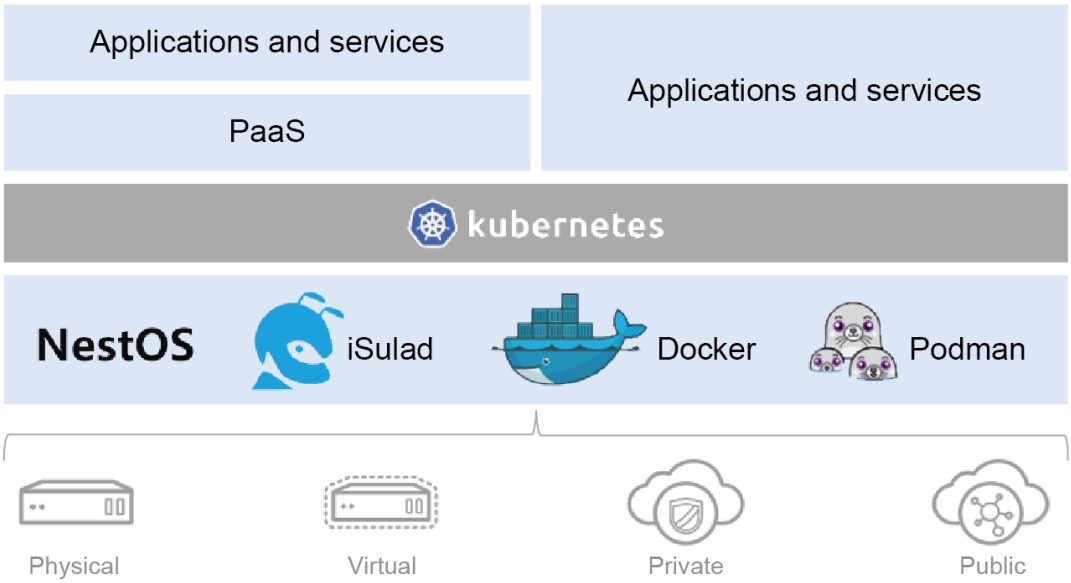
CEs. In this way, the overall service reliability does not deteriorate when partial memory mirroring (a RAS feature called address range mirroring) is used.

6 Cloud Base

NestOS

NestOS is a cloud OS incubated in the openEuler community. It runs rpm-ostree and Ignition technologies over a dual rootfs and atomic update design, and uses nestos-assembler for quick integration and build. NestOS is compatible with Kubernetes and OpenStack, and reduces container overheads and enables easy cluster setup in large-scale containerized environments.

Feature Description



- **Out-of-the-box availability**: integrates popular container engines such as iSulad, Docker, and Podman to provide lightweight and tailored OSs for the cloud.
- **Easy configuration**: uses the Ignition utility to install and configure a large number of cluster nodes with a single configuration.
- **Secure management**: runs rpm-ostree to manage software packages and works with the openEuler software package source to ensure secure and stable atomic updates.
- **Hitless node updating**: uses Zincati to provide automatic node updates and reboot without interrupting services.
- **Dual rootfs**: executes dual rootfs for active/standby switchovers, to ensure integrity and security during system running.

Application Scenarios

NestOS is developed for containerized cloud applications. openEuler 24.03 LTS introduces the NestOS-Kubernetes-Deployer to resolve problems such as inconsistent and repeated O&M operations of stacks and platforms. These problems are typically caused by decoupling of containers and underlying environments when using container and container orchestration

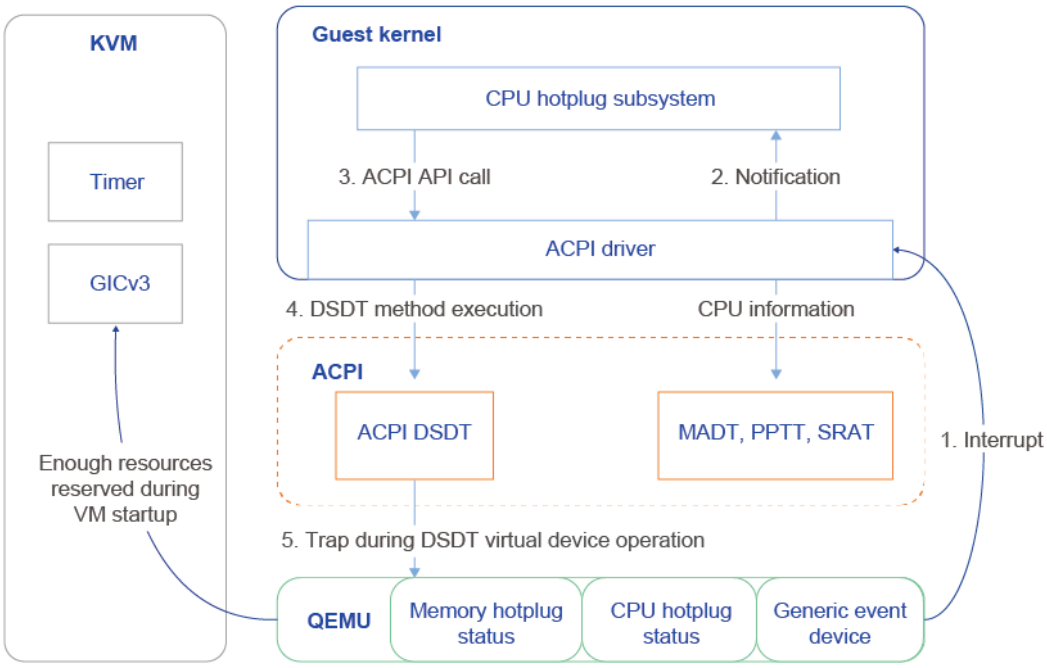
technologies for rollout and O&M. NestOS resolves these problems to ensure consistency between services and the base OS.

7 Enhanced Features

vCPU Hotplug on AArch64

vCPU hotplug allows you to increase or decrease the number of CPUs for a running VM without affecting services on it. When the internal service pressure rises to a level where existing CPUs become saturated, vCPU hotplug can dynamically boost the computing power of a VM, guaranteeing stable service throughput. vCPU hotplug also enables the removal of unused computing resources during low service load, minimizing computing costs.

Feature Description



- **Enhancement**: Building upon the existing vCPU hot adding feature of earlier openEuler versions, openEuler 24.03 LTS now includes vCPU hot removal, enhancing its adaptability for more scenarios. However, the hotplug protocol has been modified in openEuler 24.03 LTS and is no longer compatible with earlier versions. This means that to utilize the vCPU hotplug feature, the guest kernel version must match the QEMU version on the host.
- **External interface**: The **setvcpus** interface of libvirt is used to perform vCPU hotplug, which is the same as that in earlier versions.

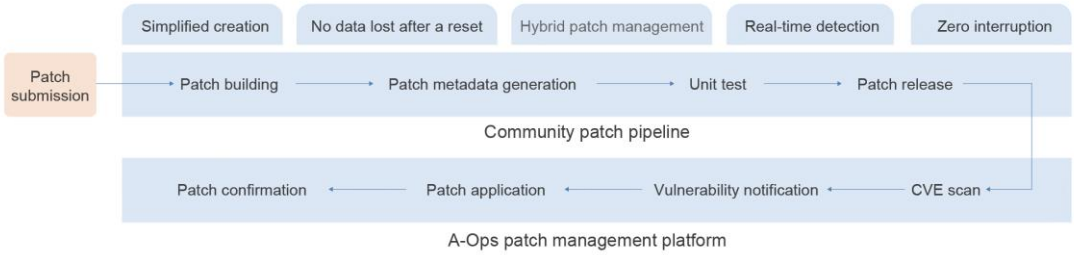
Application Scenarios

- vCPU hotplug can significantly accelerate VM startup, particularly for lightweight secure containers such as Kata containers. In this scenario, a container is initially configured with a single vCPU, with additional vCPUs being hot added after the container is started.

- vCPU hotplug enables on-demand scalability for cloud vendors, allowing them to dynamically adjust the number of vCPUs allocated to user VMs as requested, meeting users' service load requirements without disruption to running services.

A-Ops

A-Ops is an intelligent O&M platform that covers data collection, health check, fault diagnosis and rectification, and rollback of rectification. The A-Ops project includes the following sub-projects: fault detection (gala), fault locating (X-diagnosis), and vulnerability rectification (Apollo). Apollo is an intelligent patch management framework that integrates core functions such as vulnerability scanning, CVE fixing (with cold/hot patches), and hot patch rollback. Apollo periodically downloads and synchronizes security advisories and sets scheduled tasks to scan for vulnerabilities.

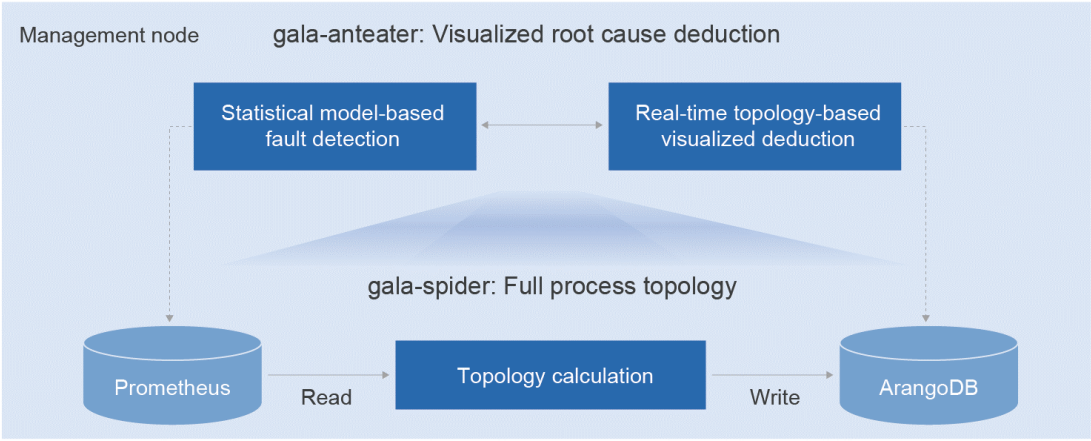


Feature Description

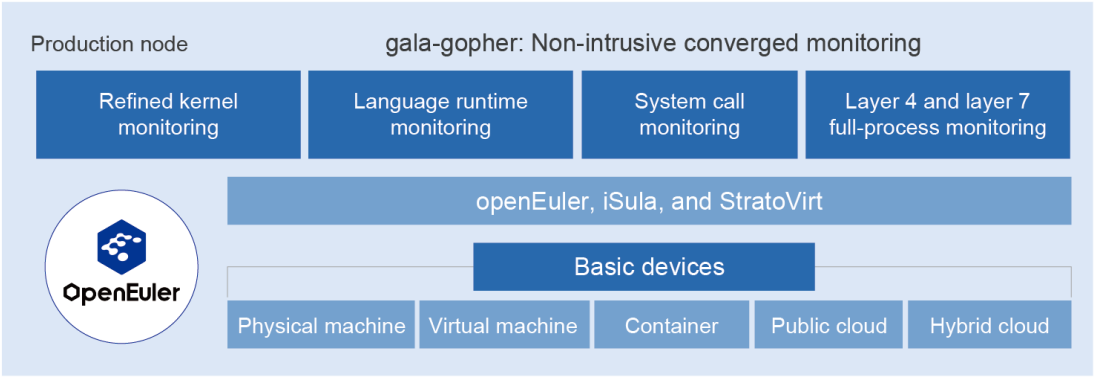
Apollo enables the intelligent management of kernel patches.

- **Hot patch source management**: When openEuler vulnerabilities are released through a security advisory, the software package used for fixing the vulnerabilities is also released in the update repository. By default, after openEuler is installed, the cold patch update repository of the corresponding OS version is provided. Users can also configure the update repository of cold or hot patches.
- **Vulnerability scanning**: Manual or periodic cluster scans can be performed to check the impact of CVEs on a cluster and cold or hot patches are provided for repair.
- **Hybrid patch management**: Cold and hot patches can be applied independently or together to implement silent incorporation of hot patches on the live network and reduce hot patch maintenance costs.
- **Hot patch lifecycle management**: hot patch removal, rollback, and query.

gala utilizes a non-intrusive observation technology based on the eBPF + Java agent and provides intelligent assistance to diagnose sub-health faults, such as performance jitter, increased error rate, and slow system response. The figure shows the architecture of gala.



OpenTelemetry ecosystem interface



gala has the following features:

- Online application performance jitter diagnosis: Online performance diagnosis for database applications, to identify issues including network issues (packet loss, retransmission, latency, and TCP zero window), I/O issues (slow drives and I/O performance deterioration), scheduling issues (high system CPU usage and deadlock), and memory issues (out of memory and leakage).
- System performance diagnosis: TCP and I/O performance jitter diagnosis in common scenarios.
- System risk inspection: Second-level inspection on kernel protocol stack packet loss, virtualization network packet loss, TCP exceptions, I/O latency, system call exceptions, resource leakage, JVM exceptions, application RPC exceptions (including error rates and latency of eight common protocols), and hardware faults (uncorrectable errors and drive media errors).
- Full-stack I/O monitoring: Full-stack I/O monitoring for the SDS scenario, covering the process I/O and block layer I/O of the guest OS, virtual storage layer frontend I/O, and SDS backend I/O.
- Refined performance profiling: Online real-time continuous collection of performance statistics, including CPU performance, memory usage, resource usage, and system calls in high precision (collected every 10 ms) and multiple dimensions (covering system, process, container, and pod) to generate flame and timeline graphs.

- Full-stack monitoring and diagnosis of Kubernetes pods: Real-time topology of pod cluster service flow, pod performance monitoring, DNS monitoring, and SQL monitoring from the perspective of Kubernetes.

Application Scenarios

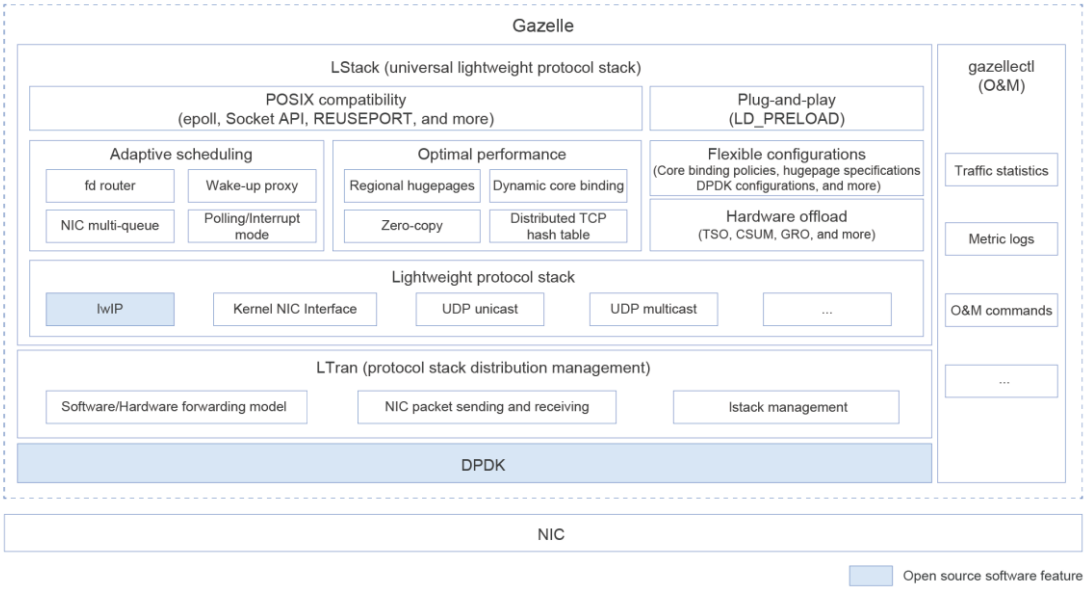
A-Ops is applicable to openEuler and other Linux distributions in database, SDS, virtualization, and cloud-native scenarios. A-Ops provides full-stack monitoring capabilities for users in industries such as finance, telecom, and Internet to diagnose sub-health faults, and promptly checks configuration errors caused by manual operations in cluster scenarios. In addition, A-Ops manages cold and hot patches in a unified manner to simplify patch management and directly provides hot patches for high-severity kernel CVEs to prevent the system from being restarted when dealing with the emergent kernel issues.

Gazelle

Gazelle is a high-performance user-mode protocol stack. It directly reads and writes NIC packets in user mode based on the Data Plane Development Kit (DPDK), transmits the packets through shared hugepage memory, and uses the LwIP protocol stack, thereby greatly improving the network I/O throughput of applications and accelerating the network for databases. With Gazelle, high performance and universality can be achieved at the same time. In openEuler 24.03 LTS, support for the UDP protocol and related interfaces is added for Gazelle to enrich the user-mode protocol stack.

Feature Description

Gazelle architecture



- **High performance (ultra-lightweight)**: High-performance lightweight protocol stack capabilities are implemented based on DPDK and LwIP.
- **Ultimate performance**: A highly linearizable concurrent protocol stack is implemented based on technologies such as regional hugepage splitting, dynamic core binding, and full-path zero-copy.

- **Hardware acceleration**: TCP Segmentation Offload (TSO), checksum (CSUM) offload, Generic Receive Offload (GRO), and other offload technologies streamline the vertical acceleration of software and hardware.
- **Universality (POSIX compatibility)**: Full compatibility with POSIX APIs eliminates the need to modify applications. The recvfrom and sendto interfaces of UDP are supported.
- **General networking model**: Adaptive scheduling of the networking model is implemented based on mechanisms such as fd router and wake-up proxy. The UDP multi-node multicast model meets the requirements of any network application scenario.
- **Usability (plug-and-play)**: LD_PRELOAD enables zero-cost deployment by removing the requirement for service adaptation.
- **Easy O&M (O&M tool)**: Complete O&M methods, such as traffic statistics, metric logs, and CLI commands, are provided.

New features

- Available in single VLAN, bond4, and bond6 modes, and supports NIC self-healing after network cables are reinstalled.
- A single-instance Redis application on Kunpeng 920 VMs supports over 5,000 connections, improving performance by more than 30%.
- The TCP_STREAM and TCP_RR tests of netperf (packet length less than 1,463 bytes) are supported.
- Logs of the LStack, lwIP, and gazellectl modules of Gazelle are refined for more accurate fault locating.
- User-mode UDP stack achieves 50% higher performance over the kernel protocol stack.

Application Scenarios

Gazelle is designed to address network I/O bottlenecks for applications such as Redis and MySQL.

iSulad

iSulad is a lightweight container engine developed using C/C++. It is not restricted by hardware architecture or specifications, has low memory overhead, and can be used in a wider range of fields. iSulad features a unified architecture design that is suited for cloud, edge, and device scenarios. It controls performance and memory overheads in different scenarios to meet specific containerization requirements. In the openEuler 24.03 LTS version, iSulad adds support for Container Runtime Interface (CRI) v1.29, control group (cgroup) v2, and Container Device Interface (CDI).

Feature Description

- **CRI v1.29**: CRI is a common protocol used by kublet to communicate with container engines. CRI v1.29 has the following API changes over CRI v1.25:

API	Change Description
RuntimeConfig	With this new API, iSulad supports systemd-cgroup and cgroupfs drivers, and this API can be called to obtain the drive type.
GetContainerEvents	This new API can be called to obtain pod lifecycle events in stream mode.
ContainerStats	ContainerStats has a new field, **SwapUsage**, which can be used to

API	Change Description
	obtain information about virtual memory usage.
ContainerStatus	ContainerStatus has a new value, **OOMKilled**, for the **reason** field. The **reason** field is used to check whether an out-of-memory (OOM) killed event has occurred in the container.

- **cgroup v2**: As a mechanism that controls resource usage of process groups in Linux, cgroup has the two versions, v1 and v2. Compared with cgroup v1, cgroup v2 features unified hierarchy, accurate resource control, and more efficient resource allocation. With cgroup v2, Kubernetes is able to delegate more secure cgroup subtrees to containers and support enhanced resource allocation and isolation.
- **CDI**: It is an API specification for container runtimes to connect to third-party devices. Device vendors can write device description files for their devices based on the CDI specification. Container engines can then load devices based on their description files.

Constraints

- CRI v1.29 supports only the runc runtime.
- cgroup out-of-memory (OOM) triggers the deletion of a container's cgroup path. If iSulad processes an OOM event after the cgroup path is deleted, iSulad cannot capture the OOM event. As a result, the **reason** field of the ContainerStatus API may be incorrect.
- iSulad does not allow using different cgroup drivers together when managing containers. After a container is started, the cgroup driver configuration in iSulad must not be changed.
- iSulad identifies only cgroups mounted to the **/sys/fs/cgroup** directory.
- iSulad does not allow the mixed use of cgroup v1 and cgroup v2. The cgroup version used in iSulad is determined by the cgroup version in the **/sys/fs/cgroup** directory.
- iSulad supports the CDI feature through CRI only.

Application Scenarios

CRI v1.29 can be used for Kubernetes 1.29 and iSulad interconnection. It can be enabled using the **enable-cri-v1** option of iSulad.

To enable cgroup v2 support in iSulad, the cgroup v2 feature must be enabled in the kernel first.

The CDI feature of iSulad supports CDI-compliant devices. When creating a container with CRI, users can specify a device name that complies with the CDI specification. During the creation, iSulad mounts the right device designated in the CDI description file to the container.

Secure Boot

Secure boot uses public and private key pairs to sign and verify components in the boot process. A typical secure boot process uses the previous component to verify the digital signature of the next component, and only once it is verified, the next component runs; otherwise, the boot stops. Secure boot prevents unverified components from being loaded and executed, to mitigate security threats to the system and ensure integrity of each component.

In addition to secure boot, openEuler extends digital signature protection to kernel modules and application files (secure boot in a broad sense) and supports IMA-based file integrity. Verification comprises four phases:

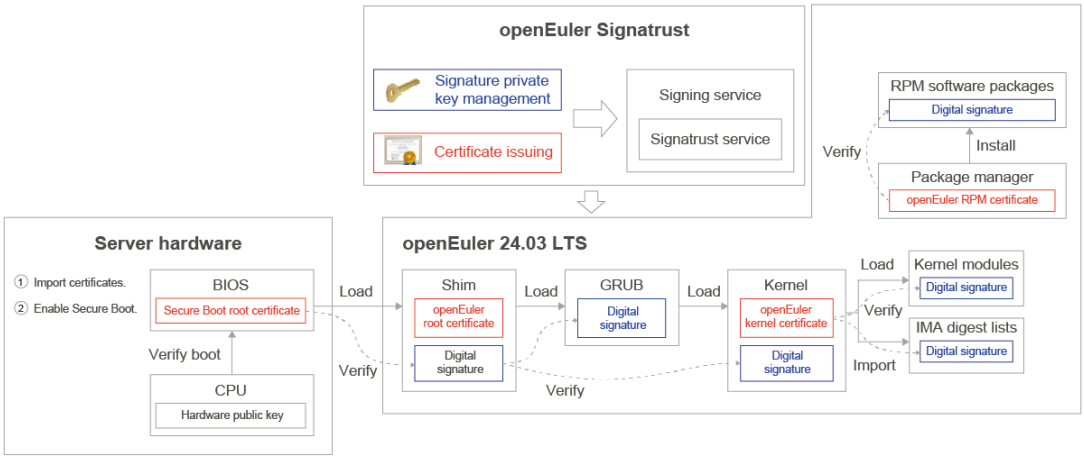
- **Boot phase**: BIOS > Shim > GRUB > Kernel (signature verification is performed before EFI loading).
- **Running phase (module loading)**: Kernel > Kernel module (signature verification is performed when a module is inserted).
- **Running phase (file access)**: Kernel > File (signature verification is performed upon application execution or common file access).
- **Running phase (software package installation)**: Package manager > RPM software package (signature verification is performed during software package installation).

Feature Description

Secure boot is available out-of-box thanks to a PKI-based software build signing system in the openEuler community. In the software build phase, digital signatures are automatically added to target files and public key certificates are preset for critical components. After installing the openEuler image, you can enable the signature verification to improve system security.

openEuler Signatrust is a reliable signing service developed by the infrastructure SIG. Designed for typical OS signing operations, it supports key management for OpenPGP and X.509 systems and can seamlessly integrate with various software package forms, including EFI, RPM, KO, and ISO. It enables the signing process for countless software packages, boosting efficiency and enhancing community key management.

The following is an overview of build signing in openEuler 24.03 LTS:



1. openEuler Signatrust generates and manages public and private key pairs and certificates, and provides signing services through Signatrust.
2. The EulerMaker build system calls signing interface of Signatrust to add a digital signature to the target file during software build.
3. Components with the signature verification function, like shim and kernel, are preconfigured with signature verification certificates in the build phase.
4. After the openEuler image is installed, functions are enabled such as secure boot, kernel module verification, IMA, and RPM verification. Signature verification is performed during system boot and running to ensure the authenticity and integrity of system components.

Obtain openEuler signature's root certificate from the certificate center: https://www.openeuler.org/en/security/certificate-center/.

Constraints

- Signatrust can only sign components from the openEuler community, and not those developed by external projects or custom user files.
- Signatrust supports only the RSA algorithm and keys of 4,096 bits.

openEuler 24.03 LTS integrates the following default signatures:

File Type	File Format	Signature Format
EFI files	EFI image	Authenticode
Kernel module files	Kernel module	CMS
IMA digest lists	Binary	CMS
RPM software packages	RPM package	OpenPGP

Application Scenarios

Deployments that require secure boot, kernel module verification, IMA, and other security functions as well as signature verification.

GreatSQL

GreatSQL is a free open source database designed to deliver high availability, performance, compatibility, and security for financial-grade operations. It is an ideal replacement for MySQL or Percona Server for MySQL.

Feature Description

- **High availability**

  GreatSQL has optimized the underlying working mechanisms of MySQL Group Replication (MGR) to deliver high availability and performance stability for MGR.

  - Supports geographical labels to improve data reliability across equipment rooms.
  - Supports arbitrator nodes to achieve higher availability at lower server costs.
  - Read/write nodes now support dynamic virtual IP addresses (VIPs) for streamlined HA switchover.
  - Supports the single-primary mode for faster running and higher performance.
  - Supports intelligent primary node election to optimize HA switchover.
  - Adopts a new flow control algorithm to ensure stable transactions and avoid severe jitters.
  - Resolved fluctuation issues when a node is added or removed.
  - Optimized the algorithm for clearing the transaction authentication queue to prevent typical fluctuations that occur every minute in heavy-traffic loads.
  - Resolved the issue that the MGR cluster is blocked when the drive space on some nodes is full.
  - Resolved the issue of primary node election due to long transactions.
  - Resolved the issue that the system waits for a long time during recovery.

- **High performance**

  Compared with MySQL and Percona Server for MySQL, GreatSQL delivers higher, more stable performance, recording 30% faster performance than MySQL in TPC-C tests. In the TPC-H test, performance is increased ten- or even hundred-fold.

  - Supports the large-scale parallel in-memory query accelerator AP engine, which is similar to MySQL HeatWave, to improve data analysis by several orders of magnitude.
  - Supports InnoDB parallel query in lightweight OLAP deployments, averaging 15 to 40 times improvement in TPC-H tests.
  - Optimized the InnoDB transaction system with optimization schemes, such as lock splitting and lock-free transactions, to boost performance in OLTP scenarios by ~20%.
  - Supports parallel **LOAD DATA** for scenarios where a large amount of data is frequently imported, with a 20-fold improvement in performance.
  - Supports the thread pool to reduce the cost of creating and destroying threads and ensure stable performance in high concurrency scenarios.

- **High compatibility**

  Supports common Oracle applications, including data types, functions, SQL syntax, and stored procedures.

- **High security**

  Supports multiple security features, such as logical backup encryption, cloned backup encryption, audit log storage to tables, and tablespace encryption using SM algorithms, to further ensure service data security in financial-grade application scenarios.

Application Scenarios

GreatSQL are suited to the following scenarios:

- **Financial-grade high availability**: Optimized underlying mechanisms of MGR to further improve availability and performance stability.
- **High-concurrency transaction system**: In TPC-C and TPC-H tests, GreatSQL delivers 30% and >100% higher performance that of MySQL, respectively.
- **Oracle compatibility**: GreatSQL is compatible with most Oracle data types, functions, SQL syntax, and stored procedures.
- **Scenarios with high security requirements**: Security features are used to ensure data security in financial-grade deployments.

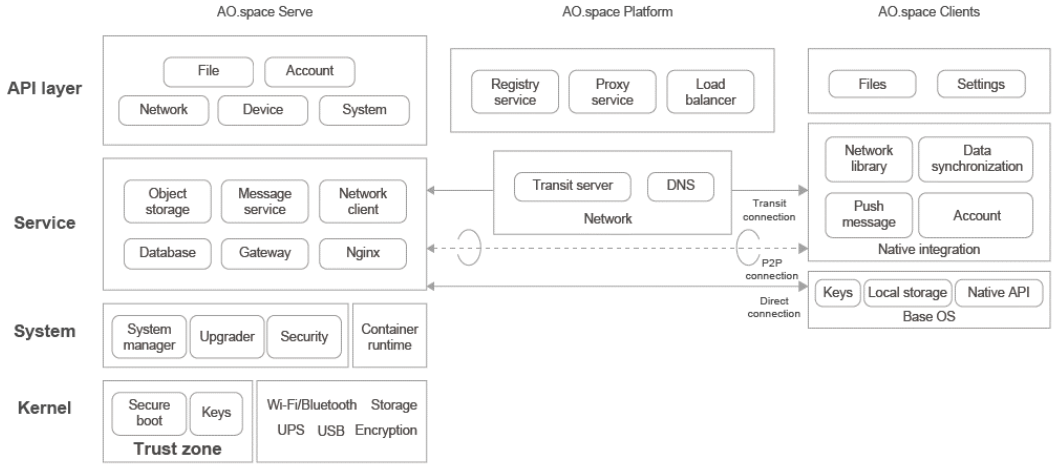
Repository

https://gitee.com/src-openeuler/greatsql

AO.space

AO.space is a solution that focuses on the protection of personal data and privacy. By utilizing technologies such as end-to-end encryption and device-based authentication, it empowers users with complete control over their accounts and personal data. Furthermore, AO.space enables anytime, anywhere data access with technologies like transparent platform forwarding, point-to-point (P2P) acceleration, and direct LAN connection. In addition, AO.space leverages progressive web application (PWA) and cloud native technologies to build a seamlessly integrated frontend and backend application ecosystem.

Feature Description



The AO.space system consists of the server, client, and platform. The server, deployed on a continuously running network device like a personal server or computer, is the core of the personal space. Clients consist of regularly used personal devices like smartphones, tablets, and computers. AO.space offers web, iOS, and Android client applications. The platform provides essential network access, security protection, and other services when personal data cannot be parsed.

- **Server**: The server is the core of AO.space and is generally deployed on a personal device. It consists of space software, space services, container runtime, base OS (such as openEuler), and hardware. The following services and applications of AO.space are deployed on the base OS as containers:
  - Web service (Nginx): entry service of the server.
  - Agent: manages basic space services and bridges the server, clients, and platform. It supports multiple OSs.
  - Gateway: routes, forwards, encrypts, decrypts, and authenticates API traffic, and authorizes requests to the AO.space application layer.
  - AOFS: virtual file system that combines object storage and file storage to store and manage files in the space.
  - Preview: generates previews for files in the space.
  - Container manager (ContainerMgr): communicates with underlying container services.
  - SQL database instance (PostgreSQL): stores and manages data in relational databases.
  - NoSQL database instance (Redis): stores and manages data in non-relational databases and provides the messaging function.
  - Network client: establishes a secure communication channel with the network forwarding service of the platform to ensure stable communication between the server and clients on different networks, and sets up P2P connections with clients.
  - Space applications: expand the functionality of the AO.space system. Space applications include frontend applications, backend applications, and hybrid applications. These built-in or third-party applications (such as CardDAV and CalDAV) can be accessed using the domain name of the space.
- **Client**: Clients, as the frontend of the AO.space system, are responsible for user interaction on different personal devices, providing users with seamless access to all functions of

AO.space anytime and anywhere. Currently, AO.space offers web, iOS, and Android client applications that incorporate the following key modules:

- P2P encrypted channel
- Space binding
- File
- Device
- Home
- Space application
- Security

- **Platform**: The platform provides basic network resources and related management capabilities. It contains the following components:
  - Ingress gateway (Endpoint): processes and allocates the overall traffic in the AO.space system.
  - Base service (BaseService): registers devices within the AO.space system, and coordinates and manages platform network resources such as domain names and forwarding proxies.
  - Network forwarding service (Transit server): forwards network traffic so that a user can securely access the server through the Internet, whether at home or in the office.

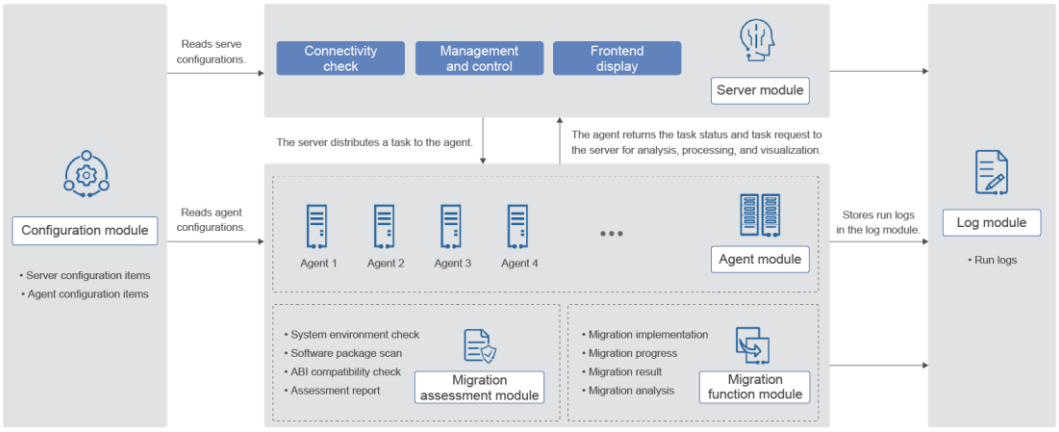
Application Scenarios

In the rapid evolution of the digital landscape, AO.space delivers innovative design and practices in areas such as personal digital identity, edge computing, data privacy protection, trusted data storage, and personal AI assistant, contributing to the sustainable development of the personal digital ecosystem.

migration-tools

migration-tools, developed by UnionTech Software Technology Co., Ltd., is positioned to meet demand for smooth, stable, and secure migration to the openEuler OS. Currently, it supports OS migration to openEuler on the WebUI.

Feature Description



migration-tools comprises modules for the server, agent, configurations, logs, migration assessment, and migration function.

- **Server module**: the core of migration-tools. This module is developed on the Python Flask Web framework. It receives task requests, processes execution instructions, and distributes the instructions to each Agent.
- **Agent module**: installed in the OS to be migrated to receive task requests from the Server module and perform migration.
- **Configuration module**: reads configuration files for the Server and Agent modules.
- **Log module**: records logs during migration.
- **Migration assessment module**: provides assessment reports such as basic environment check, software package comparison and analysis, and pre-migration compatibility checks.
- **Migration function module**: provides quick migration, displays the migration progress, and checks the migration result.

Application Scenarios

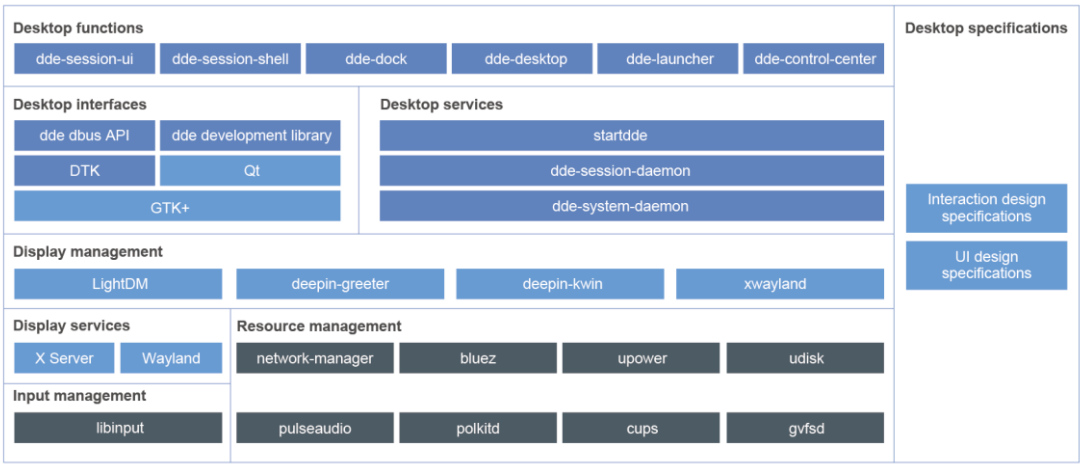
Finance, telecom, energy, and other industries often need to replace the OSs on existing hardware, such as the x86_64 architecture. migration-tools is a good choice for migrating applications and system components from other OSs to openEuler.

DDE for Servers

Deepin Desktop Environment (DDE) was originally developed for Uniontech OS and has been used in the desktop, server, and dedicated device versions of Uniontech OS.

Feature Description

DDE focuses on delivering high quality user interactions and visual design. DDE is powered by independently developed core technologies for desktop environments and provides login, screen locking, desktop, file manager, launcher, dock, window manager, control center, and additional functions. Due to its user-friendly interface, excellent interactivity, high reliability, and strong privacy protection, it is one of the most popular desktop environments among users. DDE helps you boost your creativity and efficiency at work, keep in touch with friends, effortlessly browse the Internet, and enjoy music and videos.



DDE adopts the DTK framework with unified UI elements and excellent UX design, and integrated third-party graphics libraries such as Qt and GTK+. Display services, input management, and resource management are at the bottom layer and are generally backend services written in Go. They provide interfaces required by upper-layer GUI programs to implement desktop functions such as user creation, screen brightness setting, device volume setting, and network connection management. Display management, desktop interfaces, and desktop services are at the shell layer and communicate with backend services through the D-Bus protocol. They provide support for UI definition and interactions, such as the login screen, window appearance, and GUI application controls.

Application Scenarios

Desktop functions are at the application layer and are generally functional interfaces that can be operated by users, such as the launcher and dock.

Kiran-desktop 2.6

Kiran-desktop is a desktop environment developed by Kylinsec. Its login screen, screen locking, start menu, control center, and other various modules comprise a friendly and intuitive user interface. Now, you can use Kiran-desktop 2.6 in openEuler 24.03 LTS.

Feature Description

Kiran-desktop provides a graphical user interface for the OS, including the login screen, desktop icons, control panel, system panel, file manager, and desktop applications.

- **Login screen**: Kiran-desktop allows you to log in using your user name and password. Type your user name, press **Enter**, type your password, and then press **Enter** again to log in. Kiran-desktop also provides multi-factor login authentication. In the **Authentication Management** section in **Control Center**, you can register your biometric data like fingerprints, finger veins, iris, and face or utilize a Ukey for authentication. You can also choose to enable the authentication methods you prefer.
- **Desktop icons**: The following desktop icons are displayed after you log in to the system:
  - **Computer**: View both local and accessible remote drives and folders from this computer.
  - **Home folder**: Browse the content of your home directory.
  - **Trash**: Review deleted files.
- **System panel**: The system panel at the lower part of the screen consists of the taskbar, tray, and the date and time area. The taskbar displays pinned and running applications. By default, the start menu, file manager, Firefox browser, and workspace are displayed. You can also pin other applications to the taskbar and open, close, maximize, and minimize application windows through the taskbar. The tray allows you to change input methods, adjust the volume, and set the network. The date and time area is displayed on the right of the tray.
- **Control Center**: **Control Center** offers a comprehensive graphical environment for configuring your desktop preferences. It provides various tools for desktop customization, system configuration management, and network service configuration. From **Control Center**, you can:
  - Configure and manage the system.
  - Configure network services.
  - Personalize the desktop environment.

- **File Manager**: **File Manager** provides efficient management of files and directories by helping you create, edit, copy, move, delete, open, cut, and rename files and directories.

Application Scenarios

Kiran-desktop provides common desktop applications. You can click the start menu to see all installed applications.

- **Firefox**: Firefox is a free and open source web browser that utilizes the Gecko rendering engine to support multiple OSs. It is small in size, fast in speed, and has advanced features like tabbed browsing, faster loading, a pop-up blocker, customizable toolbar, extension management, better search features, and a convenient sidebar.
- **Terminal**: **Terminal** is a medium for users to use commands. You can enter commands in the terminal window to interact with the OS.
- **Calculator**: **Calculator** is a quick and simple tool for performing basic mathematical operations such as addition, subtraction, multiplication, and division. It also provides scientific computing and programmer modes.
- **Text Editor**: **Text Editor** is a convenient tool for quickly writing down notes and viewing or editing plain text files.
- **Disks**: **Disks** allows you to view, modify, and configuring drives and media. You can use this tool to create and restore drive images, partition drives, and format drives.
- **Help Manual**: Kiran-desktop provides a help manual that introduces the features of the desktop environment and common applications in the system.
- **Screenshot Tool**: Ported from the Kylinsec Server OS, this lightweight and versatile screenshot software offers a simple user interface and easy usage.

UKUI

UKUI is a lightweight desktop environment developed by the KylinSoft software team for Linux distributions. Compatible with mainstream architectures such as x86 and AArch64, it features an intuitive UI and user-friendly and consistent interaction.

Feature Description

- **Control panel**: Allows you to perform basic system settings, such as date and time, preferences, and device management.
- **Start menu**: Manages all applications installed in the system. You can switch between the default window and full-screen modes, and search for required content using Chinese, English, Pinyin, and initial letters.
- **Taskbar**: Supports dark and light themes and other special effects (such as frosted glass), and allows real-time preview of files, folders, terminals, web pages, and images.
- **Sidebar**: Divided into two parts. The upper part is used to manage notifications, and the lower part provides shortcuts such as the screenshot and system settings.
- **File manager**: Combines the search bar and address bar, and provides multi-tab display and convenient file search.

Application Scenarios

UKUI provides a streamlined and efficient desktop GUI.

OpenStack Wallaby and Antelope

OpenStack is an open source project that provides a cloud computing management platform. It aims to deliver scalable and flexible cloud computing services to support private and public cloud environments.

Feature Description

OpenStack offers a series of services and tools to help build and manage public, private, and hybrid clouds. The service types include:

- **Compute service**: creates, manages, and monitors VMs. It empowers users to quickly create, deploy, and destroy VMs and container instances, enabling flexible management and optimal utilization of computing resources.
- **Storage service**: provides object storage, block storage, file storage, and other storage. Block storage services, such as Cinder, allow users to dynamically allocate and manage persistent block storage devices, such as VM drives. Object storage services, such as Swift, provide a scalable and distributed object storage solution, facilitating storage of large amounts of unstructured data.
- **Network service**: empowers users to create, manage, and monitor virtual networks, and provides capabilities for topology planning, subnet management, and security group configuration. These features enable building of complex network structures while ensuring security and reliability.
- **Identity authentication service**: provides comprehensive identity management and access control capabilities, including user, role, and permissions management. It ensures secure access and management of cloud resources while safeguarding data confidentiality and integrity.
- **Image service**: enables image creation, management, and sharing through image uploading, downloading, and deletion. Users can perform management operations on images with ease and quickly deploy VM instances.
- **Orchestration service**: automates application deployment and management, and facilitates service collaboration and integration. Orchestration services like Heat help streamline application deployment and management by automatically perform related tasks based on user-defined templates.

Application Scenarios

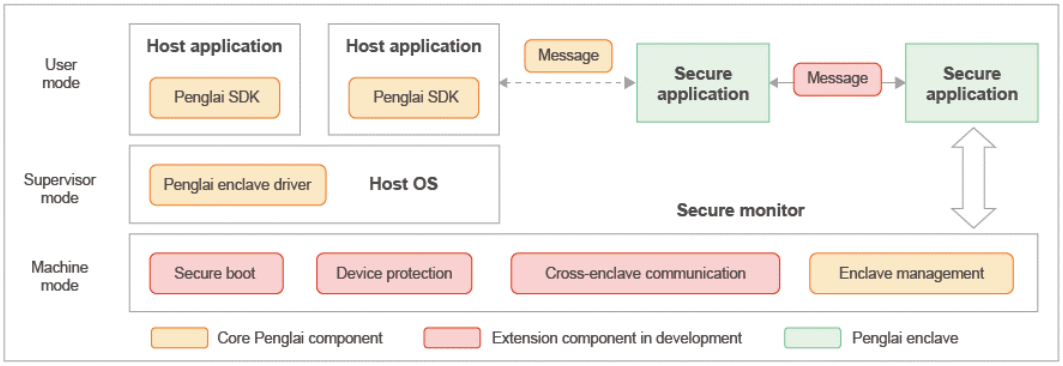
OpenStack is applicable to the following scenarios:

- **Private cloud**: Enterprises can deploy OpenStack in their own data centers or private cloud environments based on their requirements and IT resource status. In this way, OpenStack provides centralized resource management, automated deployment, auto scaling, and robust security measures such as access control, data encryption, and log audit.
- **Public cloud**: OpenStack implements resource pools for public clouds to meet the performance and user isolation requirements in multi-tenant cloud environments. In this scenario, OpenStack empowers public cloud providers by offering the infrastructure for elastic computing, containers, networking, and storage, facilitating centralized resource management and ensuring high availability.
- **Hybrid cloud**: OpenStack offers a comprehensive solution for hybrid clouds. A hybrid cloud integrates private and public clouds, facilitating seamless data and application migration, backup, and restoration for enhanced flexibility, efficiency, and security.
- **Large-scale VM management**: OpenStack helps enterprises and service providers plan and manage a large number of VMs for on-demand provisioning of computing resources.

Penglai TEE Support on RISC-V

The Penglai TEE patch enhances openEuler on RISC-V with Trusted Execution Environment (TEE) support, leveraging the hardware security mechanisms of RISC-V like physical memory protection (PMP) for high-security application scenarios, including secure communication, key protection, and code authentication.

Feature Description



The main components of the Penglai TEE system include a software development kit (SDK), a secure monitor (which is the core of Penglai TEE) responsible for lifecycle management and resource allocation for enclave instances, user-mode enclave instances with PMP-protected memory, and PMP hardware extension support.

Constraints:

- Currently, only C or C++ applications can run in Penglai TEE.
- The POSIX interface forwarding function of secGear is required for system call support.
- The RISC-V hardware must support the PMP mechanism.
- When resources are limited, Penglai TEE currently supports up to 14 enclave instances, with potential limitations due to hardware resources. Future updates will expand this capacity.
- Currently, a Penglai enclave uses 4 MB secure memory by default, which can be modified in the Penglai enclave driver.
- The Penglai TEE SDK introduced to openEuler does not contain the code of secGear. You can download and compile the SDK by referring to the secGear development framework.

Application Scenarios

- **Secure communication**: For scenarios requiring encrypted communication and data protection, such as financial services, Penglai TEE delivers secure cryptographic capabilities including encryption, decryption, signature verification, and hash algorithms.
- **Secure boot and authentication**: For device boot and remote access control, Penglai TEE provides a secure boot procedure and authentication mechanism to prevent unauthorized access and tampering.
- **Encrypted processing of data**: Penglai TEE provides a secure execution environment for data that needs to be processed in an encrypted environment, such as personal privacy information and sensitive enterprise data.

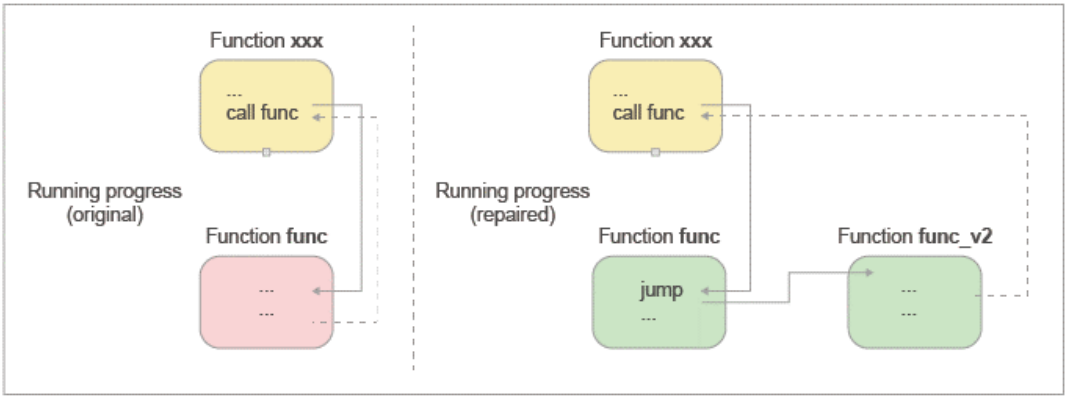
- **Internet-of-Things (IoT)**: The lightweight design of Penglai TEE enables deployment on resource-constrained IoT devices, providing essential security protection.

Kernel Hot Patches on RISC-V

openEuler's kernel hot patch mechanism (including the code and patch creation tools) for x86_64 and AArch64 has been successfully ported to RISC-V, maintaining equivalent functionality, application scenarios, and constraints. Hot patches offer a valuable solution for addressing vulnerabilities without the need for kernel or application restarts, ensuring OS performance and security on servers.

Feature Description

Created based on functions as basic repair units, a hot patch module replaces instructions of defective functions with those of patch functions, allowing for hot fixes without machine or application restarts. openEuler provides tools for kernel-mode hot patch creation (kpatch), user-mode hot patch creation (libcareplus), and hot patch service management (SysCare).



Constraints: The kernel hot patch mechanism on RISC-V has the same constraints as SysCare and kpatch.

- Only 64-bit OSs are supported.
- Only files of the Executable and Linkable Format (ELF) are supported.
- C programs are supported.
- Global and static variables can be modified with limitations.
- Multiple files or symbols cannot have the same name.

Application Scenarios

- Non-disruptive hot patching of kernel vulnerabilities on servers
- Non-disruptive hot patching of application vulnerabilities
- Hot patch service offered by operating system vendors (OSVs)

HAOC

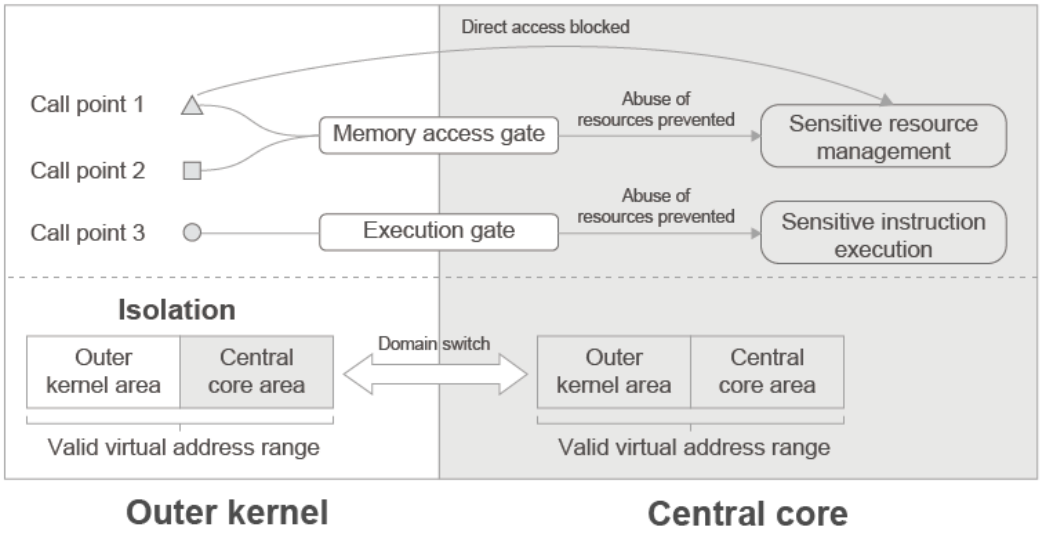
Hardware-Assisted OS Compartmentalization (HAOC) enhances the security of the internal OS structure to prevent kernel hacks. At its core, HAOC aims to reconstruct the flat Linux kernel

architecture based on the least privilege principle to form an intra-kernel layered architecture, that is, kernel compartmentalization. This design achieves multi-layer isolation by dividing a monolithic kernel into multiple compartments on the legacy processor, a practice that prevents horizontal movement and privilege escalation of kernel attacks.

At the innermost layer of the compartmentalized kernel is the central core, a trusted computing base (TCB) that provides a secure and efficient isolated execution environment without trusting the Linux kernel. This layer maintains the last line of defense for the most sensitive data in the Linux kernel, while protecting other layers.

HAOC is released with openEuler 24.03 LTS, and though the kernel compartmentalization for Arm processors can protect the page table structure and process credentials and effectively mitigate kernel privilege escalation attacks. In the future, HAOC will continue to integrate new features, including an improved central core layer to protect sensitive resources, a high-risk layer in the compartmentalized kernel for security of device drivers and kernel extensions, and compartmentalized design of the x86 architecture.

Feature Description



The central core layer of HAOC realizes a strict and isolated execution environment inside the address space in the Linux kernel. Based on the PAN and HPD mechanisms of Arm processors, memory access and execution gates filter direct access to memory and arbitrary execution of sensitive/privileged instructions, to ensure the central core layer can be accessed only through the security gates. The security gates do not depend on the protection mechanism of the outer kernel, and can independently ensure the atomicity and certainty of the execution.

The central core layer provides an independent running environment comprising independent stacks and secure interrupt handling capabilities. Currently, the central core layer provides security protection for kernel page tables and credentials, meaning attackers cannot bypass the security gates to tamper with sensitive resources. To further prevent potential code obfuscation, in which attackers abuse the security gate interfaces to access sensitive data, the central core layer also performs security checks on all access interfaces.

Application Scenarios

Structural improvements to the Linux kernel. Even if the kernel has unknown vulnerabilities, HAOC can still prevent attackers from implementing kernel privilege escalation.

GCC for openEuler

The baseline version of GCC for openEuler has been upgraded from open source GCC 10.3 to GCC 12.3 and supports features such as automatic feedback-directed optimization (AutoFDO), software and hardware collaboration, memory optimization, Scalable Vector Extension (SVE), and vectorized math libraries.

- The default language of GCC for openEuler has been upgraded from C14/C++14 to C17/C++17, enabling GCC for openEuler to support more hardware features like Armv9-A and x86 AVX512-FP16.

Item	GCC 10.3.0	GCC 11.3.0	GCC 12.3.0
Release date	2021-04-08	2022-04-21	2023-05-08
C standard	C17 by default C2x supported	C17 by default C2x supported	C17 by default C2x supported
C++ standard	C++ 14 by default C++ 17 supported C++ 2a experimental optimization C++ 20 partially supported	C++ 17 supported C++ 2a experimental optimization C++ 20 partially supported	C++ 17 supported C++ 2a experimental optimization C++ 20 partially supported
New architecture features	Armv8.6-a (bfloat16 Extension/Matrix Multiply Extension) SVE2 Cortex-A77 Cortex-A76AE Cortex-A65 Cortex-A65E Cortex-A34	Armv8.6-a, +bf16, +i8mm Armv8.6-r Cortex-A78 Cortex-A78AE Cortex-A78C Cortex-X1	Armv8.6-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

- GCC for openEuler supports structure optimization and instruction selection optimization, leveraging Arm hardware features to improve system running efficiency. In the benchmark tests such as SPEC CPU 2017, GCC for openEuler has proven to deliver higher performance than GCC 10.3 of the upstream community.
- Further, it fuels AutoFDO to improve the performance of the MySQL database at the application layer.

Feature Description

- **SVE vectorization**: Significantly improves program running performance for Arm-based machines that support SVE instructions.
- **Memory layout**: Rearranges the positions of structure members so that frequently accessed structure members are placed in continuous memory space, increasing the cache hit ratio and improving program performance.
- **Redundant member elimination**: Eliminates structure members that are never read and deletes redundant write statements, which in turn reduces the memory occupied by the structure and alleviates subsequent bandwidth pressure, while improving performance.

- **Array comparison**: Implements parallel comparison of array elements to improve execution efficiency.
- **Arm instructions**: Simplifies the pipeline of ccmp instructions for a wide range of deployments.
- **AutoFDO**: Uses perf to collect and parse program information and optimizes feedback across the compilation and binary phases, boosting mainstream applications such as MySQL databases.

Application Scenarios

In terms of general-purpose computing, GCC for openEuler reported 20% gains over GCC 10.3 in the SPEC CPU 2017 test.

With AutoFDO enabled, MySQL performance is 15% higher, while with kernel-mode PGO enabled, the UnixBench performance boosted by over 3%.

8 Copyright Statement

9 Trademark

10 Appendixes

Appendix 1: Setting Up the Development Environment

Environment Setup	URL
Downloading and installing openEuler	https://openeuler.org/en/download/
Preparing the development environment	https://gitee.com/openeuler/community/blob/master/en/contributors/prepare-environment.md
Building a software package	https://gitee.com/openeuler/community/blob/master/en/contributors/package-install.md

Appendix 2: Security Handling Process and Disclosure

Security Issue Disclosure	URL
Security handling process	https://gitee.com/openeuler/security-committee/blob/master/security-process-en.md
Security disclosure	https://gitee.com/openeuler/security-committee/blob/master/security-disclosure-en.md