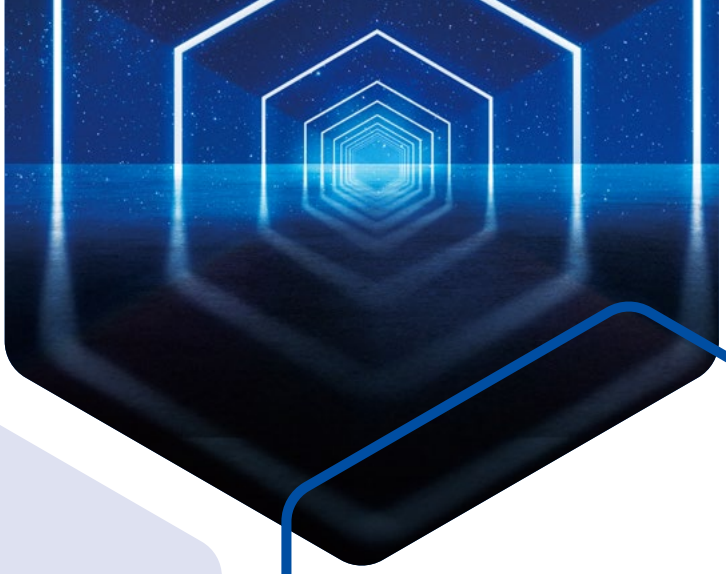
openEuler 22.03 LTS SP2

Technical White Paper



CONTENTS

01 Introduction 01

02 Platform Architecture 04

03 Operating Environments 07

04 Scenario-specific Innovations 09

05 Kernel Innovations 12

06 Cloud Base 17

07 Enhanced Features 20

08 Copyright Statement 43

09 Trademark 44

10 Appendixes 45

01 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.

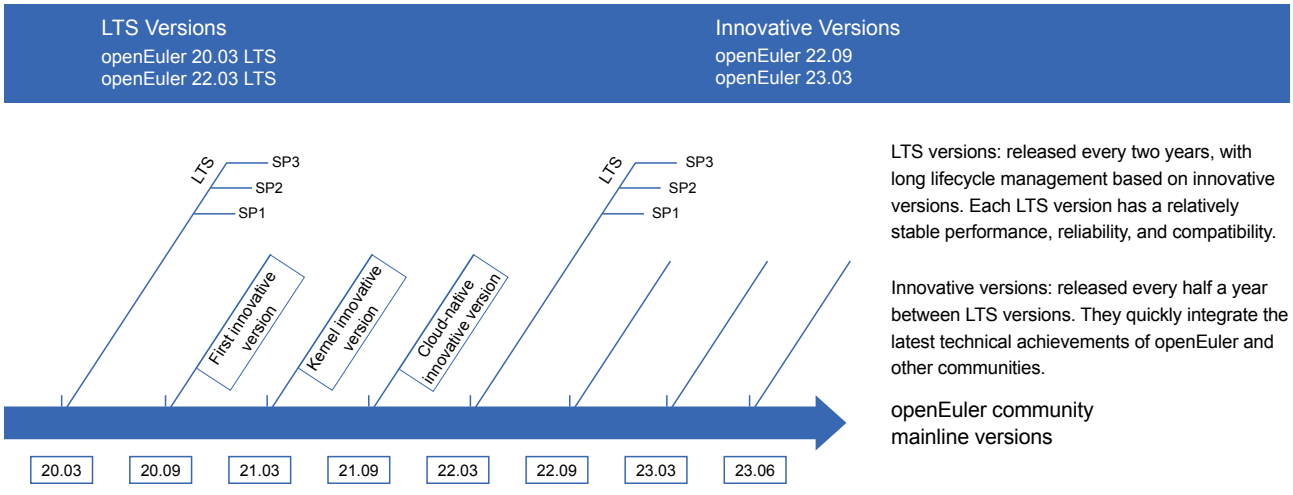
On September 30, 2022, openEuler 22.09 was released to further enhance all-scenario innovations.

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.

openEuler Version Management

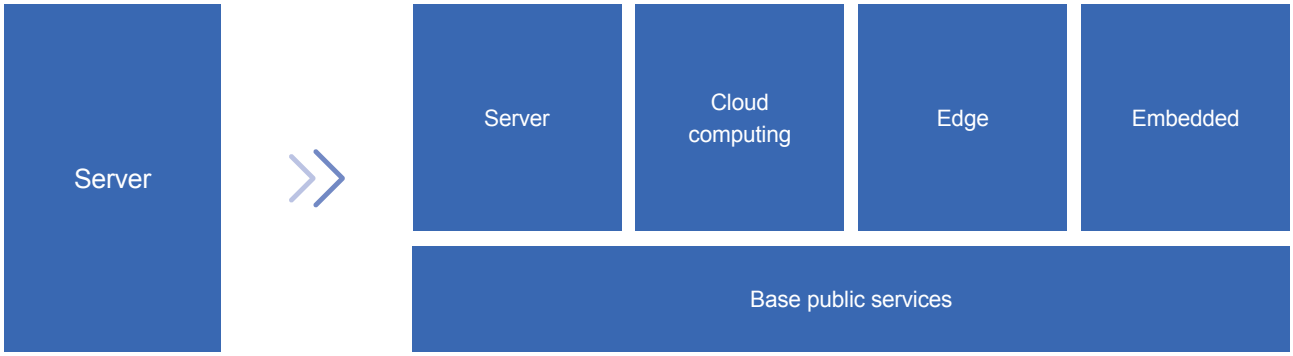


As an OS platform, openEuler releases an LTS version every two years. Each LTS version provides enhanced specifications and a secure, stable, and reliable OS for enterprise users.

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 to be used 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.

02 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 22.03 LTS SP2 runs on Linux kernel 5.10 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 22.03 LTS SP2 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:

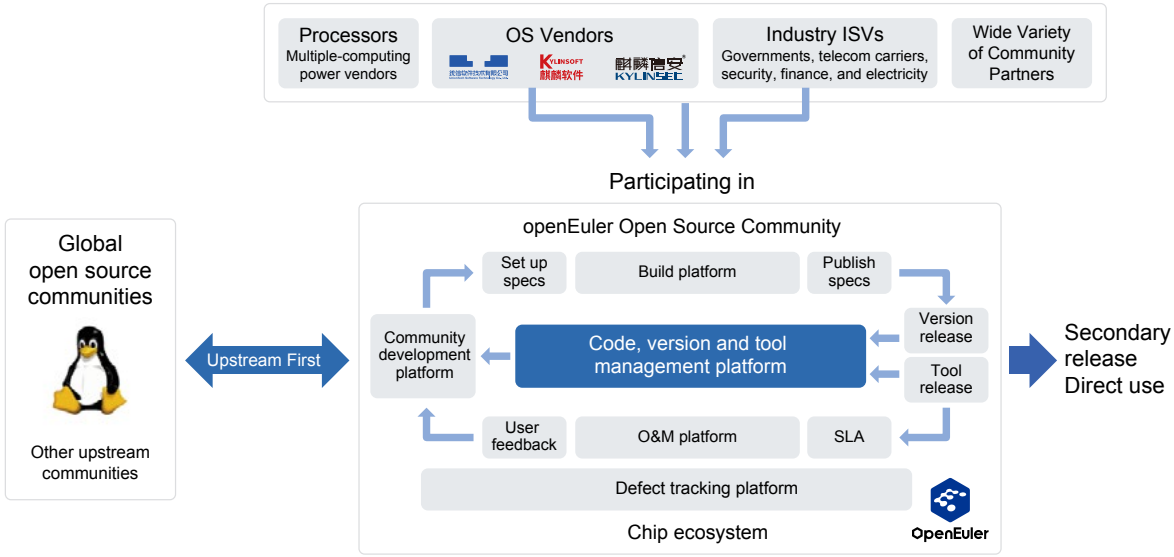
- Edge computing: openEuler 22.03 LTS SP2 Edge is released for edge computing scenarios. It integrates the KubeEdge+ edge-cloud collaboration framework to provide unified management, provisioning of edge and cloud applications, and other capabilities.
- Embedded: openEuler 22.03 LTS SP2 Embedded is released 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
CPU	Intel, AMD, Zhaoxin, Hygon	Kunpeng, Phytium

openEuler supports the following servers:

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

openEuler supports the following cards:

Hardware Type	x86	Arm
NIC	Huawei, Mellanox, Intel, NebulaMatrix, 3SNIC	Huawei, Mellanox, Intel, NebulaMatrix, 3SNIC
RAID	Avago, 3SNIC	Avago, 3SNIC
Fibre Channel	Marvell, Qlogic, Emulex	Marvell, Qlogic, Emulex
GPU & AI	NVIDIA	NVIDIA
SSD	Huawei	Huawei

For the complete compatibility list, visit https://www.openeuler.org/en/compatibility/.

Operating Environments 03

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

openEuler supports the following virtual machines (VMs):

- centos-7.9 qemu 1.5.3-175.el7 libvirt 4.5.0-36.el7 virt-manager 1.5.0-7.el7
- centos-8 qemu 2.12.0-65.module_el8.0.0+189+f9babebb.5 libvirt 4.5.0-24.3.model_el8.0.0+189+f9babebb virt-manager 2.0.0-5.el8
- fedora 32 qemu 4.2.0-7.fc32 libvirt 6.1.0-2.fc32 virt-manager 2.2.1-3.fc32
- fedora 35 qemu 6.1.0-5.fc35 libvirt 7.6.0-3.fc35 virt-manager 3.2.0-4.fc35

Item	Configuration Requirement
Architecture	AArch64, x86_64
CPU	2 CPUs
Memory	At least 4 GB
Drive	At least 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	At least 4 GB
Drive	At least 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	At least 512 MB
Drive	At least 256 MB

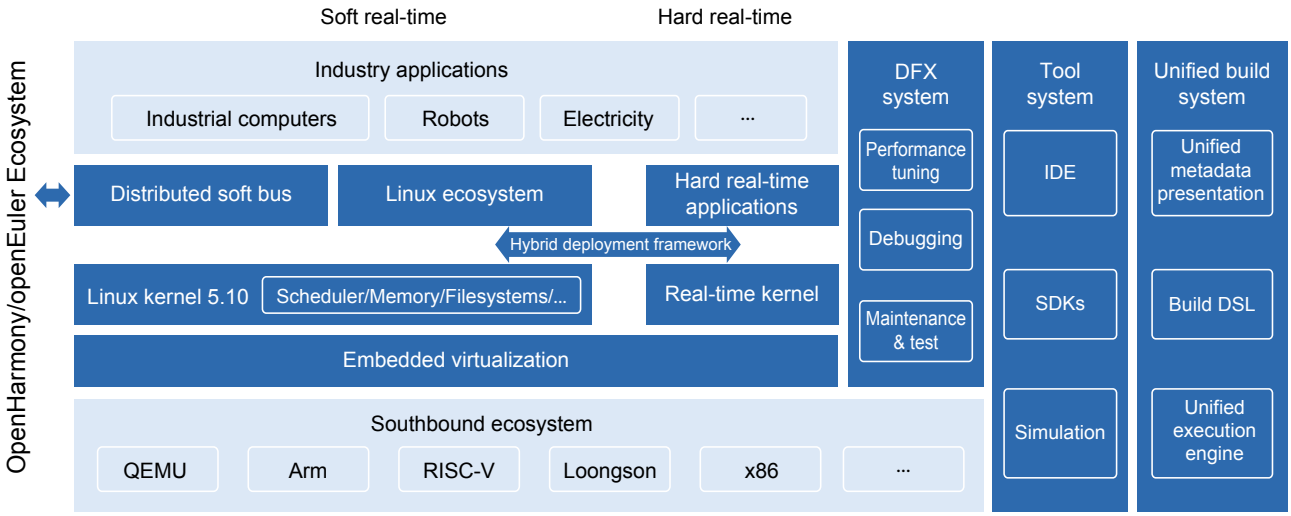
Scenario-specific Innovations 04

Embedded

openEuler 22.03 LTS SP2 Embedded offers a distributed soft bus and software package build capabilities to allow for mixed criticality deployment of real-time and non-real-time planes.

openEuler Embedded helps build applications for industrial control and robotics in a closed loop design, whereby innovations help optimize its embedded technology stack and ecosystem. openEuler 22.03 LTS SP2 Embedded is equipped with an embedded virtualization base that is available in the Jailhouse virtualization solution or the OpenAMP lightweight hybrid deployment solution. You can select the most appropriate solution to suite your services. openEuler 22.03 LTS SP2 Embedded supports the Robot Operating System (ROS) Humble version, which integrates core software packages such as ros-core, ros-base, and simultaneous localization and mapping (SLAM) to meet the ROS 2 runtime requirements. Future versions will utilize contributions from ecosystem partners, users, and community developers, to increase support for chip architectures such as RISC-V and LoongArch, and add capabilities such as industrial middleware, ROS middleware, and simulation systems.

Feature Description



openEuler 22.03 LTS SP2 Embedded provides the following features for embedded scenarios:

- Lightweight deployment: The open source Yocto is a small-scale and lightweight framework that allows you to customize OS images. It can compress OS images to under 5 MB and shorten OS startup time to under 5s.
- Support for diverse hardware types: Raspberry Pi, x86, Hi3093, and RK3568 devices can serve as the universal hardware for embedded deployments.
- Soft real-time kernel: This capability is inherited from Linux kernel 5.10, and helps respond to soft real-time interrupts within microseconds.
- Embedded virtualization base: Multiple virtualization solutions are available to cater for different hardware and service scenarios.
  - 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 FreeRTOS and openEuler Embedded Linux.
  - » Real-time virtualization is a solution originating from the openEuler community that balances performance, isolation, and flexibility. This solution supports the hybrid deployment of Zephyr and openEuler Embedded Linux.
- Embedded software packages: Over 350 common embedded software packages can be built using openEuler.
- Hard real-time kernel: The open source Real-Time Operating System (RTOS) kernel, UniProton, supports 103 POSIX APIs. It ensures a context switchover latency of 3 µs and an interrupt latency of 2 µs.

The following features will be available in future versions:

- Southbound ecosystem: RISC-V and LoongArch.
- Mixed-criticality deployment framework: A mixed deployment framework will be available for critical services, delivering lifecycle management, cross-OS communication and collaboration, and service orientation, allowing more soft and hard real-time OSs to join the openEuler ecosystem.
- Embedded elastic base: Virtualization capabilities, including Jailhouse and ZVM, will be optimized with a shorter latency to support the southbound ecosystem.
- UniProton: The hard real-time middleware will provide various POSIX APIs and common middleware to facilitate application development and porting.
- Pan-industrial, -embedded APIs: High-performance northbound APIs will be defined for performance-intensive scenarios in the RTOS field.
- Industry certifications: openEuler 22.03 LTS SP2 Embedded is currently under testing to be verified for different standards and certifications, including IEC61508 and CC EAL.

Application Scenarios

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

05 Kernel Innovations

What's New in the openEuler Kernel

openEuler 22.03 LTS SP2 runs on Linux kernel 5.10 and inherits the competitive advantages of community versions and innovative features released in the openEuler community.

- 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 quality of service (QoS) of online tasks.
- 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.
- Tidal affinity scheduling: 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 adds new CPU cores to improve the QoS.
- Kernel Same-page Merging (KSM) at the process or container level: Before this feature was introduced, user-mode programs needed to explicitly call the madvise function to specify the memory address range involved in memory deduplication. However, some programs written in non-C languages cannot call madvise. openEuler 22.03 LTS SP2 supports the following functions, which enable KSM for programs without explicitly calling madvise.
  - Full-range deduplication at the process granularity: A prctl system call interface is added to enable KSM for a process. Calling this interface enables all memory addresses (private anonymous pages) in a process to participate in KSM deduplication. Subprocesses forked from the process also inherit this deduplication mode. Rather than calling the madvise function multiple times, the process only needs to call the prctl interface once to enable full-range KSM deduplication.
  - Full-range deduplication at the container granularity: The container argument **memory.ksm** is added to the memory cgroup v1 directory. After **1** is written, full-range memory deduplication is enabled for all processes in the container.
- Enhanced Data Access MONitoring (DAMON): This feature enables online, proactive, and lightweight monitoring and reclamation of memory resources when the memory load is light. You can customize a policy to initiate the most appropriate operation on the memory areas based on the monitoring result.
- Enhanced uswap: Memory pages can be swapped out to the back-end storage in user mode, which saves memory resources.
- Intel Emerald Rapids (EMR): It is Intel's next-generation CPU platform built on the Intel 7 process. With Intel EMR, openEuler boosts hardware performance and delivers new hardware features such as Trust Domain Extensions (TDX). The support for Intel EMR is critical to fully unleashing the performance of users' mission-critical applications and computing platforms on openEuler.
- ACPI for AArch64 MPAM 2.0: Memory System Resource Partitioning and Monitoring (MPAM) is an extension feature of Armv8.4. It resolves system-wide or application-specific performance deterioration due to contention for shared resources (cache, DMC, and interconnects) in server systems that run diverse types of services concurrently.

SMT Expeller Free of Priority Inversion

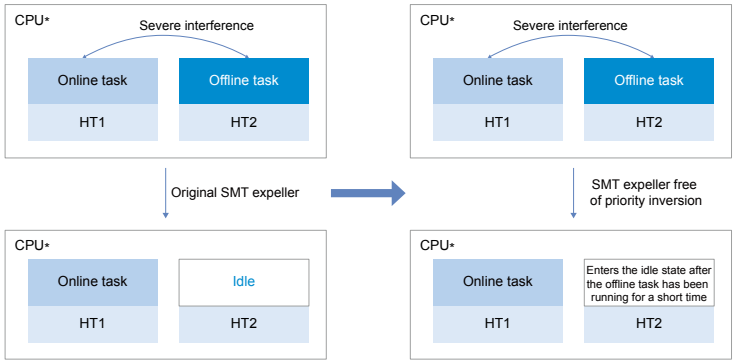
In cloud scenarios where online and offline services are deployed together to improve resource utilization, prioritizing the QoS of online services is a difficult challenge. When SMT is enabled, offline and online tasks running on the same CPU may interfere with each other. The SMT expeller for a hybrid deployment can prevent offline tasks from causing instructions per cycle (IPC) interference to online tasks. Any change made by the SMT expeller to the Completely Fair Scheduler (CFS) task policy may cause priority inversion. The SMT expeller free of priority inversion feature prevents critical resources from being occupied by expelled offline tasks.

Feature Description

Assume that CPU A and CPU B are associated for SMT. Assign an online task to CPU A and an offline task to CPU B. The online task on CPU A occupies 100% of CPU A's resources for a long time. The offline task on CPU B is expelled and is not executed, and critical resources cannot be released. In this case, if a high-priority task waits for the critical resources occupied by the offline task, the task priority is reversed. The system checks how long the offline task has been throttled to check whether the system is at risk of priority inversion. If such a risk exists, the system stops throttling the offline task until the critical resources in the kernel are released.

Two parameters are available for configuring this feature:

SMT expeller free of priority inversion



- **/proc/sys/kernel/qos_overload_detect_period_ms**

  Description: a period of time (in milliseconds). If the online task has been occupying CPU A's resources for a period longer than the value specified by this parameter, the process of resolving priority inversion is triggered.

  Value range: 100–100000

  Default value: 5000

  Configuration suggestion:

  » If the value is too small, the process of resolving priority inversion is triggered frequently, which has a great impact on the online task and may cause misreports.

  » If the value is too large, the system may be suspended for a long time due to priority inversion.

- **/proc/sys/kernel/qos_offline_wait_interval_ms**

  Description: sleep time (in milliseconds) of the offline task before entering the user mode in the event of overload.

  Value range: 100–1000

  Default value: 100

  Configuration suggestion:

  » If the value is too large, the offline task enters the sleep state and the CPU enters the idle state for a period of time after the online task is stopped. This causes the CPU to be underutilized.

  » If the value is too small, the offline task is frequently awakened and interferes with the online task.

Application Scenarios

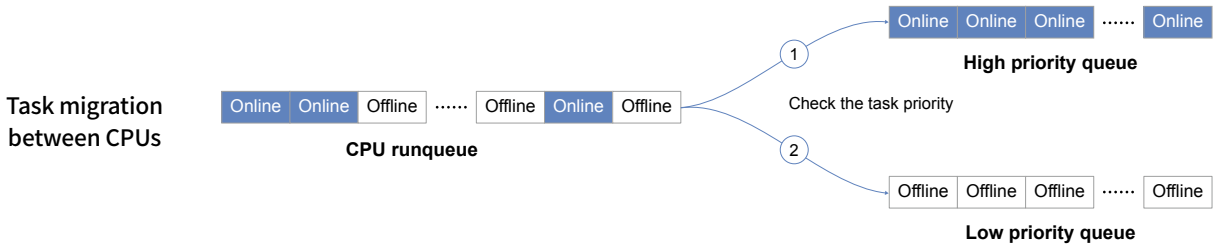
To enable SMT expeller free of priority inversion in a hybrid deployment scenario, enable the **CONFIG_QOS_SCHED_SMT_EXPELLER** argument.

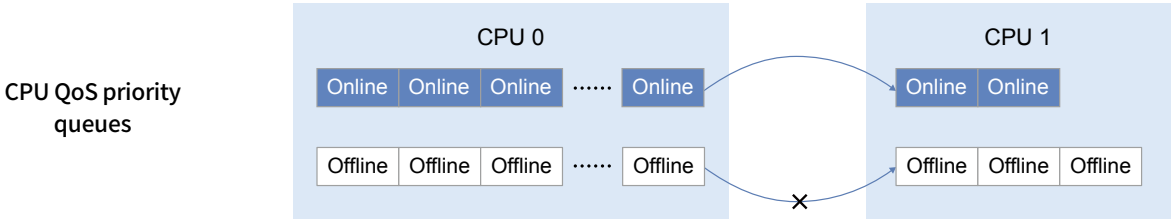
CPU QoS Priority-based Load Balancing

Task priorities are not differentiated in first-in first-out (FIFO) task migration queues between CPUs. Therefore, the QoS of high-priority tasks is not ensured in cross-CPU migration preemption. For example, CPU-sensitive tasks may not be scheduled preferentially. In scenarios where online and offline containers are deployed together, CFS load balancing requires a priority-based queuing model to implement QoS load balancing between high and low priorities. Such a model must schedule and execute online services more quickly than offline services, minimize the QoS interference from offline tasks, and make full use of CPUs.

Feature Description

Online and offline tasks managed by the CFS are assigned to task queues of different priorities. During multi-CPU load balancing, high-priority tasks are preferentially selected from the task queue to ensure that they are preferentially scheduled. Low-priority tasks are throttled during migration to reduce QoS interference and performance overhead caused by context switches and wakeup preemption of low-priority tasks.





Configurable parameter: /proc/sys/kernel/sched_prio_load_balance_enabled

Description: indicates whether to enable CPU QoS priority-based load balancing.

Value range: 0 and 1

Default value: 0

Application Scenarios

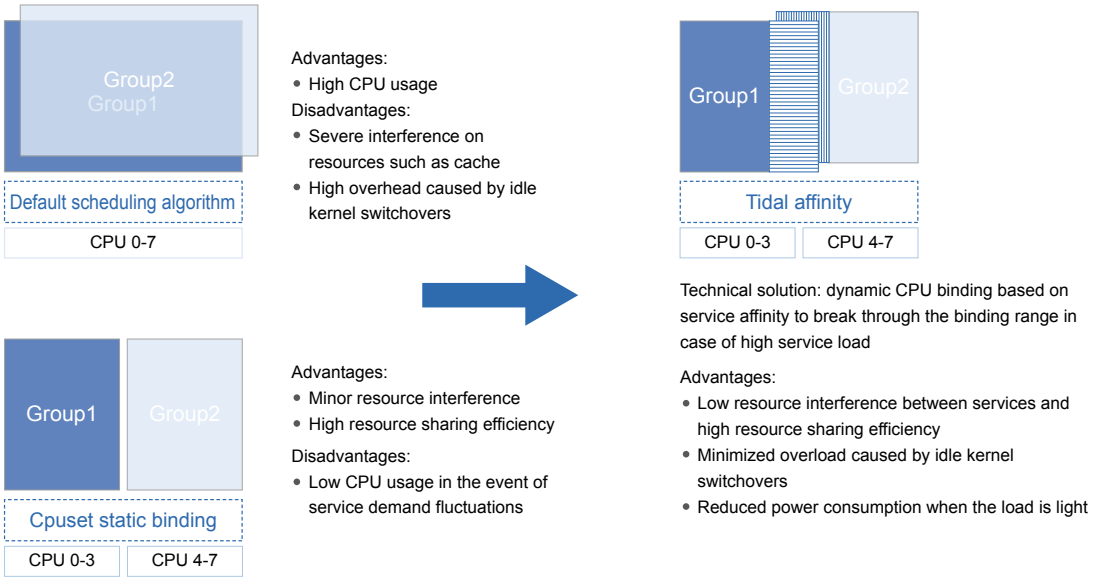
To enable CPU QoS priority-based load balancing in a hybrid deployment scenario, enable the **CONFIG_QOS_SCHED_PRIO_LB** argument.

Tidal Affinity

Multi-core servers allow you to deploy multiple types of services on the same server. A hybrid deployment improves CPU utilization but also aggravates conflicts of resources such as cache. Each service can use more CPU resources to meet the QoS. However, CPU utilization deteriorates due to problems such as more frequent migration between CPUs, idle switchovers, and cross-NUMA memory access.

Feature Description

Tidal affinity technology detects service load changes and dynamically allocates CPU resources to services. Specifically, when the service load is light, fewer CPU resources are allocated to the services, so as to reduce CPU migration, idle switchovers, and cache misses while meeting QoS requirements, improving CPU utilization, and increasing the energy efficiency ratio. When the service load is high, more CPU resources are allocated to improve the QoS and increase the CPU usage.



When the service load is light, the preferred CPUs are used for the services to enhance resource locality. When the service load is heavy, the shared CPUs are used as well to improve the QoS.

Application Scenarios

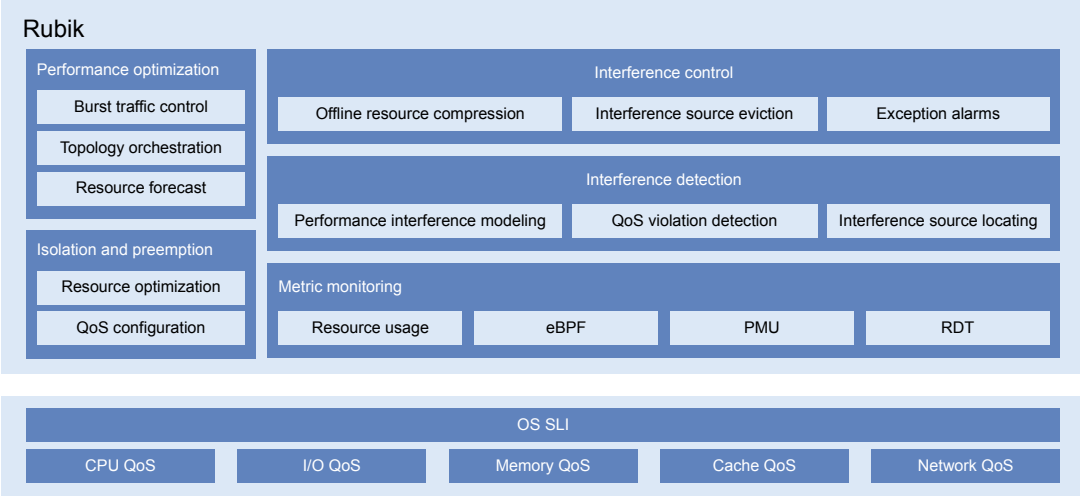
Core binding improves service performance for online-online and online-offline hybrid service deployments. When CPUs fail to be bound accurately due to dynamic load changes, tidal affinity is an effective solution to improve performance and reduce power consumption.

Cloud Base 06

Rubik

Low resource utilization of cloud data centers is a common problem in various industries. Improving resource utilization has become a pressing technical issue that must be resolved and several methods have been developed to do so. One such method is deploying services based on priorities (hybrid deployment). The core technology of hybrid deployment is resource isolation and control.

Feature Description



openEuler 22.03 LTS SP2 provides the following features:

- Enhanced cluster scheduling: Enhances OpenStack Nova to support priority-based semantic scheduling.
- Power consumption control: Limits the CPU bandwidth of low-priority VMs to reduce the overall system power consumption and ensure the QoS of high-priority VMs.
- Cache and memory bandwidth control: Limits the last level cache (LLC) and memory bandwidth of low-priority VMs. Currently, only static allocation is supported.
- CPU interference control: Supports CPU time slice preemption in microseconds, SMT interference isolation, and anti-priority-inversion.
- Memcg asynchronous memory reclamation: Limits the total memory available to offline applications in a hybrid deployment, and dynamically compresses the memory used by offline services when the online memory utilization increases.
- QuotaBurst traffic control: When the CPU traffic of key online services is limited, the limit can be exceeded in a short period of time to ensure the QoS of online services.
- Enhanced observation of pressure stall information (PSI): Collects pressure information at the cgroup v1 level, identifies and quantifies service interruption risks caused by resource contention, and improves hardware resource utilization.

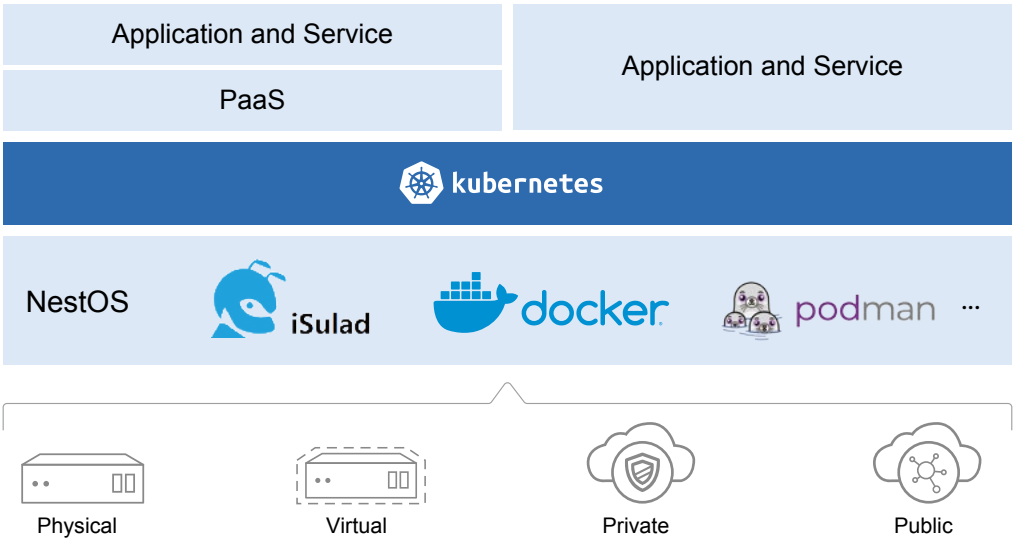
Application Scenarios

To improve resource utilization, services are classified into high- and low-priority services based on latency sensitivity, and deployed accordingly. Latency-sensitive services are recommended for high-priority VMs, such as web services, high-performance databases, real-time rendering, and machine learning inference; while services not limited by latency can be deployed on low-priority VMs, such as video encoding, big data processing, offline rendering, and machine learning training.

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 provides extensive cluster components 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 aims to satisfy the demands of containerized cloud applications by solving 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.

07 Enhanced Features

SysCare

In the world of Linux development, there is a long-standing problem: how to quickly and reliably fix vulnerabilities and rectify faults without causing interruptions to online services.

A common solution to this problem is hot patching. During service running, code-level repair is directly performed on faulty components without affecting services. However, hot patches are complex to make because they must match the source code, and are also difficult to manage. There was no simple and unified patching mechanism for user-mode components that need to adapt to different file forms, programming languages, compilation methods, and running modes.

To solve this problem, SysCare was developed.

SysCare is a system-level hotfix software that provides security patches and hot fixing for OSs. It can fix system errors without restarting hosts. SysCare combines kernel-mode and user-mode hot patching to take over system repair, saving time for users to focus on other aspects of their business. In the future, live OS upgrades will be provided to improve O&M efficiency.

Feature Description

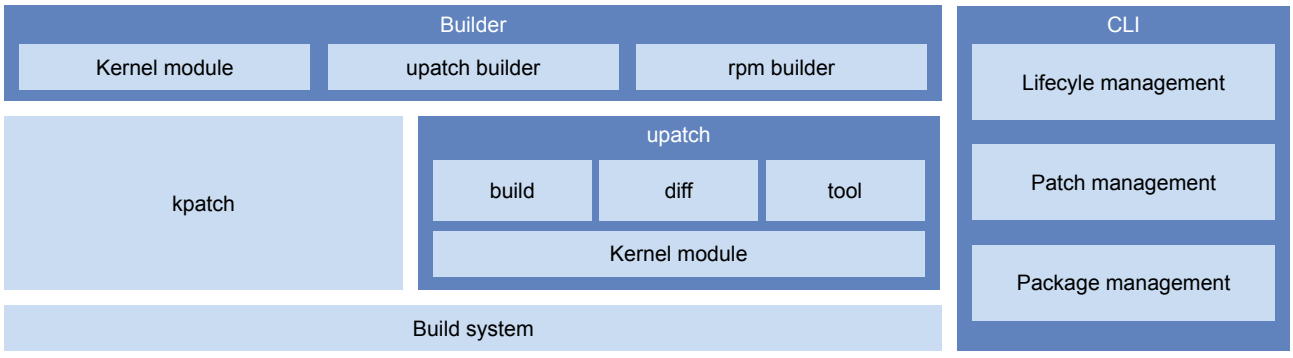
- **Hot patch making**

  To generate a hot patch RPM package, users only need to input the paths to the source RPM package, debuginfo RPM package, and patches to be installed of the target software without modifying the software source code.

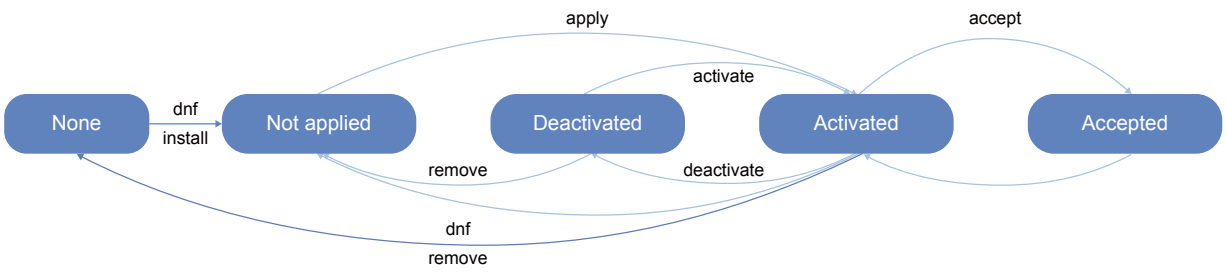
- **Patch lifecycle management**

  SysCare provides a complete and easy-to-use patch lifecycle management method to simplify usage. Users can manage hot patches by running a command. By utilizing the RPM system, SysCare can build hot patches with complete dependencies, so that hot patches can be integrated into the software repository, without the need for special processing for distribution, installation, update, and uninstallation.

SysCare architecture



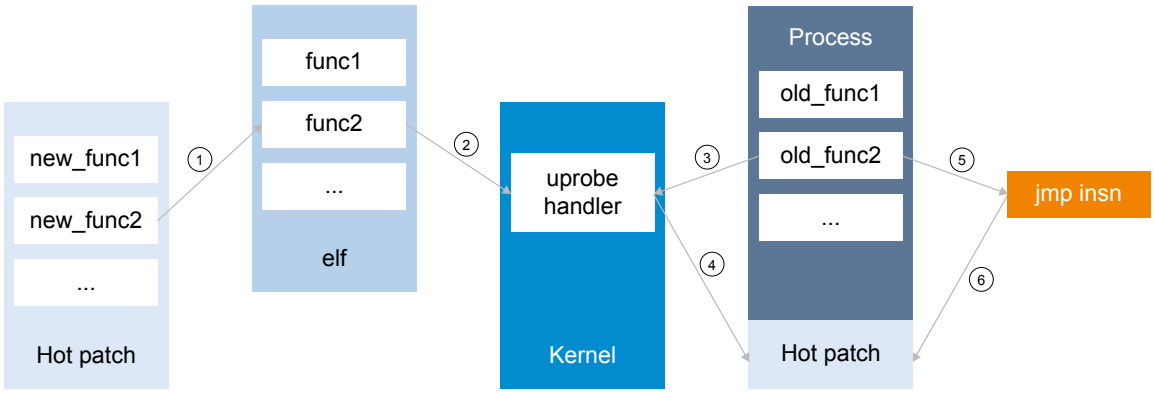
Lifecycle of a hot patch



- **User-mode hot patches for ELF files (program executable files)**

  SysCare uses the uprobe technology to bind hot patches to ELF files. When ELF files are running, uprobe can make the patches take effect. In this way, the patching process does not need to be monitored. The patches can take effect after being installed or when a new process is running, regardless of whether the user process is running. Apart from that, this technology can also install hot patches for dynamic libraries. The following figure shows how a patch takes effect.

Process of making a patch take effect

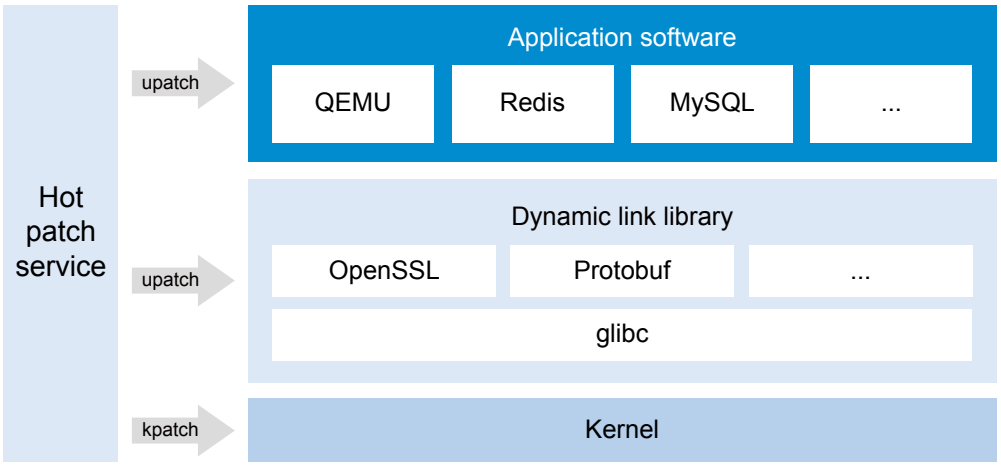


1. Execute the uprobe system call and add a uprobe breakpoint at the function to be modified.
2. Register a uprobe handler.
3. Call the uprobe handler when the function is running.
4. Map the patch to the address space of the current process.
5. Perform security check and change the first instruction of the function to a JUMP instruction, pointing to the patch address.
6. Jump to the patch address for execution.

- **Integrating kernel-mode and user-mode hot patches**

  By utilizing the upatch and kpatch technologies, SysCare streamlines the hot patch software stack from top to bottom for applications, dynamic libraries, and kernels, and provides seamless full-stack hot fixing.

Application scope of hot patches



- **New features**
  - » Compatibility with AArch64
  - » Automatic derivation of patch making parameters
  - » Saving and restoration of the patch status
  - » Restoration of the patch status after a restart
  - » Syslog capability
- **Constraints**
  - » Only available for 64-bit OSs
  - » Only available for ELF files, with no support for interpreted languages and pure assembly modification
  - » Only available for the GCC and G++ compilers, and cross compilation not supported
  - » LTO optimization not supported

Application Scenarios

Scenario 1: quick fix of common vulnerabilities and exposures (CVEs)

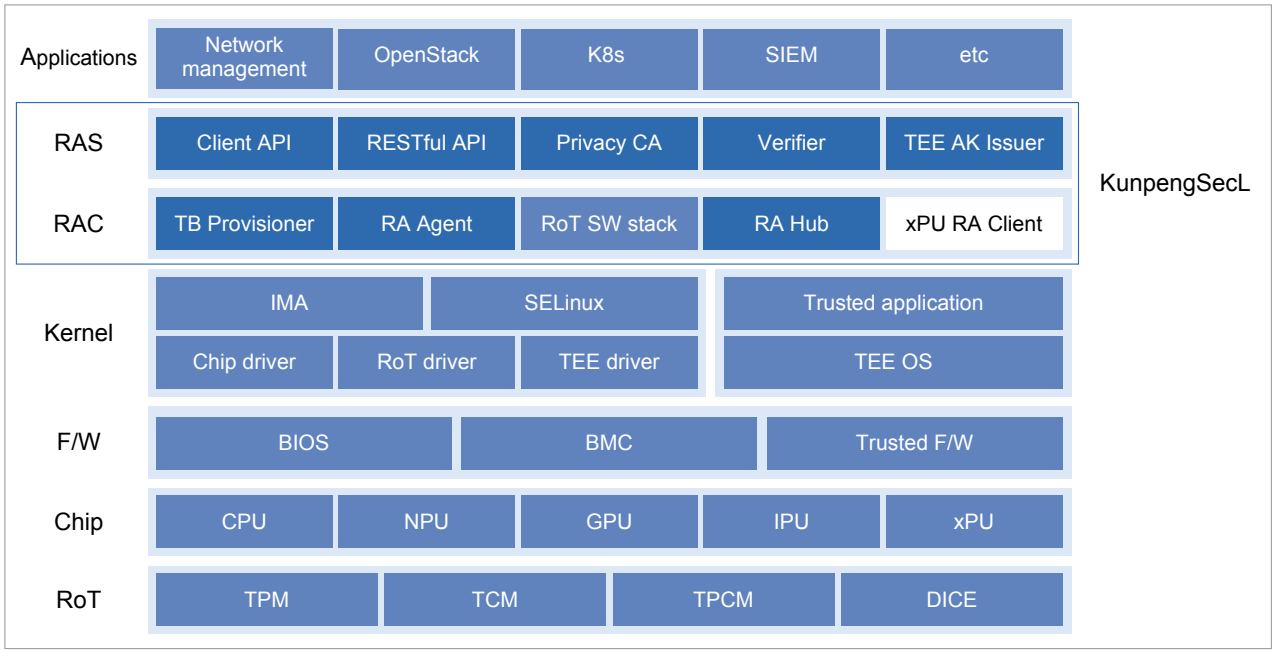
Scenario 2: temporary locating for live-network issues

KunpengSecL

Kunpeng Security Library (KunpengSecL) is a fundamental security software component running on Kunpeng processors. In the early stage, KunpengSecL focuses on trusted computing fields such as remote attestation.

Each feature of KunpengSecL consists of two parts: components and services. Components are deployed on worker nodes that provide resources (compute, storage, and network) for user workloads. The components convert platform security and trustworthiness capabilities into software interfaces and provide them for services. Services are deployed on independent management nodes to aggregate security and trustworthiness capabilities from all worker nodes and provide them for users and specified management tools to meet users' specific requirements for system security and trustworthiness design.

As the first security feature of KunpengSecL, remote attestation is an end-to-end trusted computing solution that obtains the trustworthiness status of software and hardware on worker nodes. Various resource management tools can formulate policies based on trustworthiness reports to schedule and use server resources in a differentiated manner.



The remote attestation feature of KunpengSecL supports:

- TPM-based remote attestation for universal platforms.
- Remote attestation for the Kunpeng trusted execution environment (TEE).

For details, see the project's README at https://gitee.com/openeuler/kunpengsecl/blob/master/README.en.md.

Feature Description

- **Remote attestation for platforms**
  - » Remote Attestation Service (RAS) uses the privacy certification authority (PCA) to provide remote attestation key certificates for trusted platform modules (TPMs) on worker nodes, TrustMgr to manage trustworthiness data of worker nodes, Client API to receive trustworthiness reports from worker nodes, Verifier to verify the trustworthiness status of targets, Cache to cache the trustworthiness status, and Config to manage configuration information such as policies. It provides remote attestation for users through RESTful APIs.
  - » Remote Attestation Client (RAC) uses TB Provisioner to detect and enable trusted boot for the platform in the deployment phase, RAC tools to obtain data required for remote attestation, and RA Agent to communicate with RAS to complete registration and send trustworthiness reports.
  - » The RAC tools mask the interaction details with trusted modules and systems, and will extend the support for different trusted modules in the future.
  - » RA Hub aggregates communication and functions as the proxy of the RAC in the local domain when necessary. In the future, RA Hub will provide the communication channel between RAC and RAS.
- **Remote attestation for the TEE**
  - » TEE AK Service (TAS) provides a TEE AK issuer to deploy user-defined certificates and issue the certificates to remote attestation keys.
  - » TEE verifier lib and TEE Attester provide interfaces and tools for obtaining and verifying TEE remote attestation reports.
  - » RAC and RAS also integrate remote attestation for the TEE on the server platform into the overall remote attestation architecture. The trustworthiness status of a specific trusted application (TA) can be obtained through RAS.

Application Scenarios

Scenario 1: trusted cloud hosts

By combining trusted boot of cloud physical servers and remote attestation of the platform, trusted verification can be performed on the host environment where VMs are running to provide underlying security support for cloud host users. In addition, the virtual Trusted Platform Module (vTPM) is used to support trusted boot and remote attestation of VMs. In this way, users can perceive the security and trustworthiness status of trusted cloud hosts, thereby enhancing users' confidence in using cloud hosts.

Scenario 2: key cache management

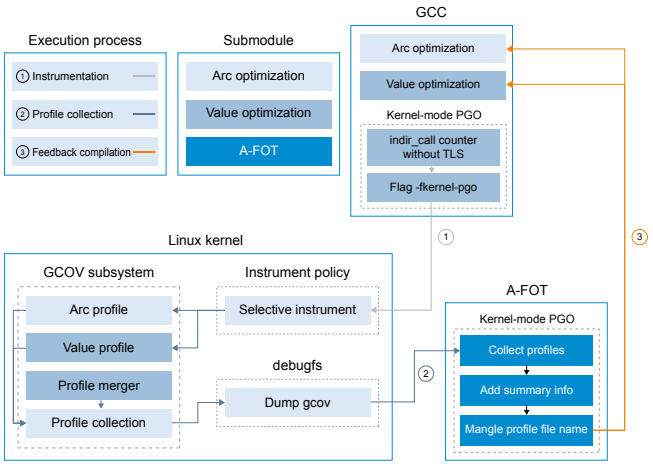
Remote attestation for the platform and TEE and local attestation for the TEE are used to harden security in scenarios where TAs obtain and cache keys from enterprises' or cloud Key Management Service (KMS), ensuring the security of keys in transmission, storage, and use.

GCC for openEuler

GCC for openEuler is developed based on the open source GCC 10.3 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 compiler fully utilizes the hardware features of the Arm architecture to achieve higher operating efficiency. In the benchmark tests such as SPEC CPU 2017, GCC for openEuler delivers higher performance than GCC 10.3 of the upstream community.
- What's more, it supports the mcmodel=medium, fp-model, quadruple-precision floating points, and vectorized math libraries.
- It also enables AutoFDO to improve the performance of the MySQL database at the application layer.
- Multiple GCC versions co-exist, including gcc-toolset-12 series whose installation packages run on GCC 12.2.0, to support Intel SPR features.
- Kernel-mode profile-guided optimization (PGO) is added. The kernel and GCC are enhanced to support compiler PGO. Users can use the A-FOT tool to build a kernel optimized for specific scenarios with a few clicks.

Overall architecture of kernel-mode PGO



Feature Description

Kernel-mode PGO:

- Kernel: supports PGO, including the arc and value profiles.
- GCC: added the **-fkernel-pgo** option to support kernel-mode PGO.
- FOT: provides kernel-mode PGO with a few clicks.

The performance improvement depends on the proportion of target application hotspots in the kernel.

Application Scenarios

In the general-purpose computing test of SPEC CPU 2017, GCC for openEuler brings about 20% performance gains compared with GCC 10.3 of the upstream community.

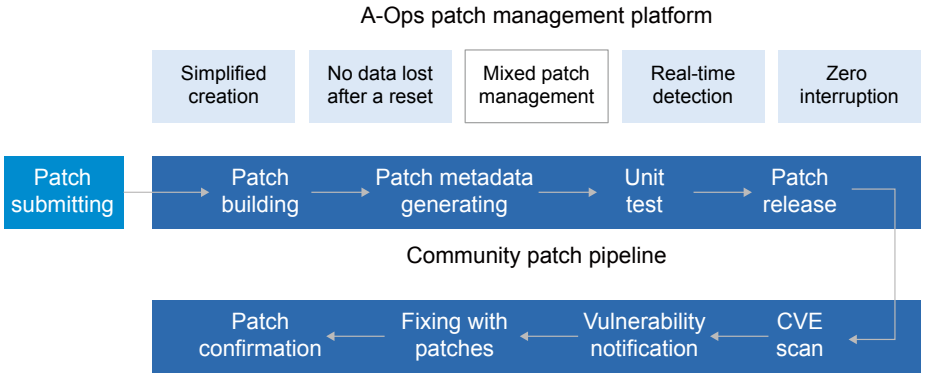
In the high-performance computing (HPC) test of Weather Research and Forecasting (WRF) and Microsystems Engineering and Materials Science (NEMO) applications, GCC for openEuler delivers 10% higher performance than GCC 10.3 of the upstream community.

In other scenarios, after AutoFDO is enabled, the MySQL performance is improved by more than 15%; after kernel-mode PGO is enabled, the UnixBench performance is improved by more than 3%.

A-Ops

A-Ops is an OS-oriented O&M platform that provides intelligent O&M solutions covering data collection, health check, fault diagnosis, and fault rectification. The A-Ops project includes the following sub-projects: fault detection (Gala), fault locating (X-diagnosis), and defect rectification (Apollo).

The Apollo project is an intelligent patch management framework. It provides real-time scanning of CVEs and bugs and cold and hot patching, in order to implement automatic discovery and zero-interruption fixing.



Feature Description

- **Community hot patch pipeline**
  - » Hot patch preparation: The target version and patch file of the software package can be specified in the cold patch pull request (PR) to make a hot patch.
  - » Hot patch release: Hot patches to be released can be automatically collected based on hot patch issues, and then released by reusing the cold patch update release logic.
- **Enhanced vulnerability management**
  - » Intelligent patch inspection: CVE inspection and notification for a single-node system or cluster, and one-click fix and rollback are supported.
  - » Hot fixing: Some CVEs can be fixed using hot patches, ensuring zero service interruption.
  - » Patch service: Cold and hot patch subscription allows patches to be acquired online.

Application Scenarios

Scenario 1: community hot patch pipeline

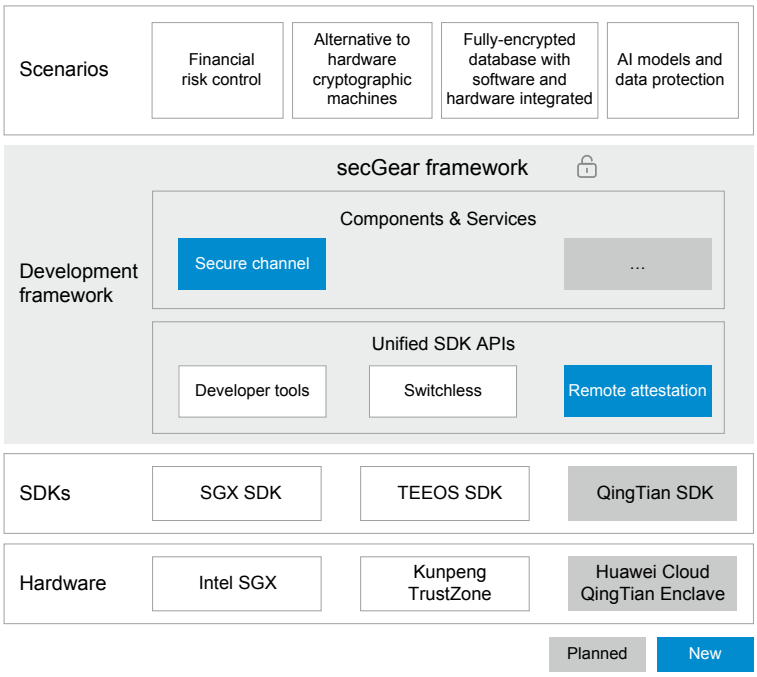
Some issues that occur on the live network cannot be resolved by simply restarting services or machines. A-Ops is a convenient tool that provides a hot patch pipeline to create, download, verify, and release hot patches based on patch files.

Scenario 2: vulnerability management

Administrators or individual developers can use the vulnerability management service to quickly detect and fix vulnerabilities, greatly reducing patch management costs, ensuring cluster security, and improving vulnerability fixing efficiency.

Enhanced secGear

secGear is a unified security software development kit (SDK) for confidential computing. The secGear unified framework masks the differences between SDKs in the TEE. Its development tools and security components help security software developers focus on services and improve development efficiency.



secGear features the following benefits:

- Architecture compatibility: It masks differences between different SDK APIs by sharing the same set of source code across multiple architectures.
- Easy development: The development tools and common security components allow users to focus on services, significantly improving development efficiency.
- High performance: The switchless feature improves the interaction performance between the rich execution environment (REE) and TEE by more than 10-fold in typical scenarios such as frequent interactions between the REE and TEE and big data interaction.

In openEuler 22.03 LTS SP2, remote attestation and secure channel are newly supported.

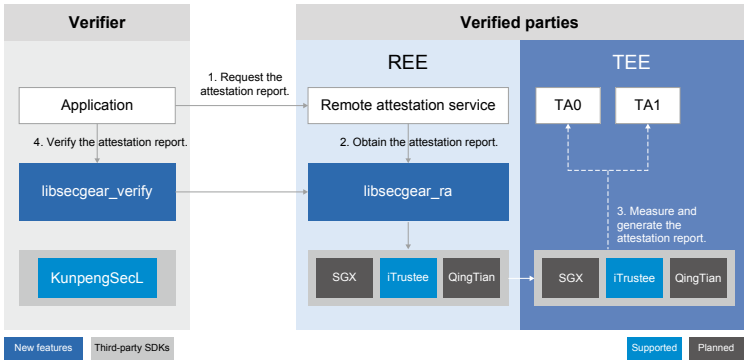
Remote Attestation

Confidential computing vendors have launched the remote attestation technology, which enables tenants to detect the trustworthiness status of the TEE and TAs on the cloud at any time.

Remote attestation is a real-time measurement technology that measures the TEE and applications running in the TEE, generates attestation reports, and uses the preset root key to sign the reports to prevent them from being tampered with or forged.

secGear encapsulates remote attestation APIs based on the remote attestation capability of each vendor's SDK. secGear must run on the Kunpeng platform. It depends on the TEE verification library of KunpengSecL to verify attestation reports.

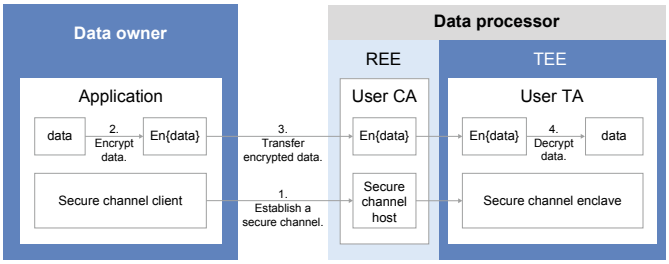
secGear and KunpengSecL TEE differ as follows: secGear is a unified development framework that provides unified remote attestation SDK APIs for obtaining and verifying remote attestation reports. Users can develop unified remote attestation services and verification programs based on secGear. KunpengSecL TEE provides the remote attestation service and attestation report verification based on the Kunpeng platform, which can be directly deployed and used by users.



Secure Channel

In cloud scenarios, data owners need to upload data to the TEE for processing. Because the TEE is not connected to the Internet, the data needs to be transferred to the REE over the network in plaintext and then transferred to the TEE. But in doing so, the plaintext data is exposed in the REE memory, posing security risks.

By combining remote attestation and key negotiation, the secGear secure channel negotiates a key between the data owner and TEE, and uses the negotiated key to encrypt and transfer data. Specifically, the REE receives the ciphertext data, after which it transfers the data to the TEE for decryption and processing.



Feature Description

- **Remote attestation**
  - » Provides remote attestation reports.
  - » Verifies remote attestation reports.
  - » Allows TAs on the same physical machine to initiate local attestation to other TAs.
- **Secure channel**

  The secure channel SDK consists of three parts: client, host, and enclave.
  - » The client establishes a secure channel and provides encryption and decryption services, which can be invoked by the data owner.
  - » The host initializes the secure channel for user CAs to invoke.
  - » The enclave supports encryption and decryption for user TAs to invoke.

Application Scenarios

Encrypted databases. An encrypted database provides SQL query and computing capabilities in the TEE. In a database client query, the client sends the ciphertext key to the TEE through the secure channel. Before secure channel negotiation, remote attestation is used to verify the TA. If the verification is successful, a secure channel is established and the key is transmitted. Otherwise, the process ends.

sysmonitor

sysmonitor is a system O&M and monitoring utility. It can monitor the usage of system resources (such as drives, CPUs, memory, and the number of processes, threads, and handles), filesystem exceptions, as well as network interface card (NIC) and file operations (which are also recorded in the logs). It can also monitor key processes and restore them when they are abnormal. In addition, sysmonitor allows you to define monitoring on custom resources.

Feature Description

Filesystems	Key processes	File operations	Drive partitions
NIC status	CPUs	Memory	Number of processes/threads
Number of handles	Inodes	Drive I/O latency	Zombie processes
Custom resources			

- Monitors the filesystems, drive partitions, NIC status, CPUs, memory, number of processes, and number of system handles.
- Monitors key processes and restores service processes quickly when they are abnormal.
- Monitors key files and logs file operations, facilitating file error locating.
- Allows you to customize the monitoring framework to extend monitoring capabilities.

Application Scenarios

In the OS O&M and monitoring scenario, sysmonitor helps O&M personnel monitor the OS running status for exceptions and assists in fault locating when the OS is abnormal.

In the NIC monitoring scenario, the status or IP address of the NIC may change due to human factors or exceptions during system running, causing services or the OS to malfunction. sysmonitor monitors NIC starts and stops and IP address additions and deletions, and then logs these operations and their operators. The logs allow O&M personnel to determine the time and operator of NIC changes.

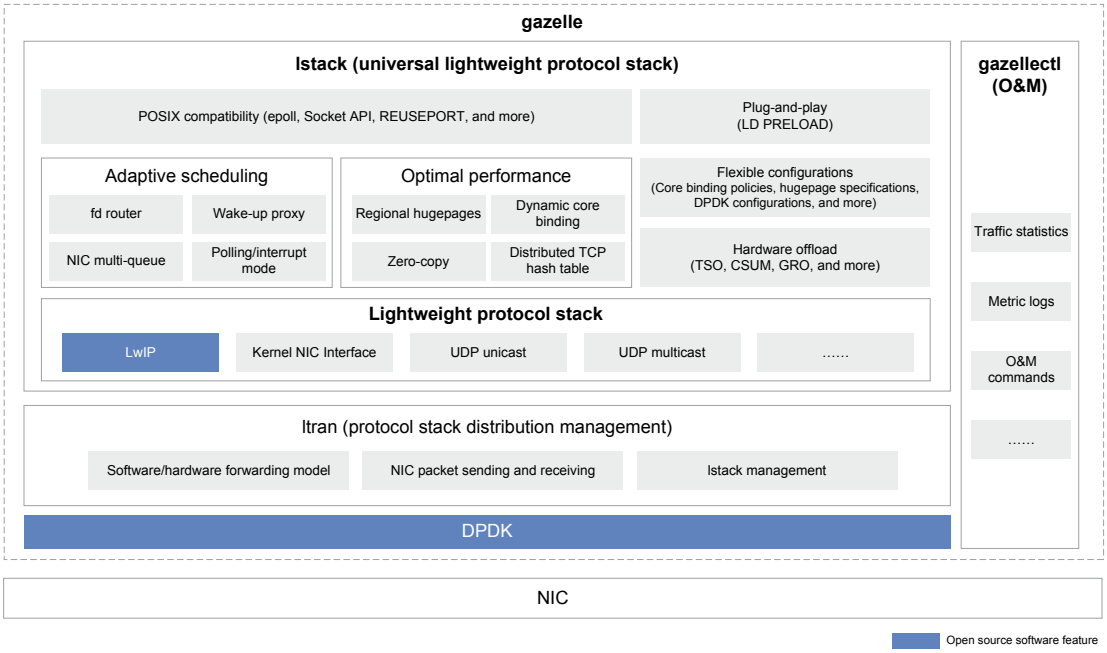
In the system resource monitoring scenario, sysmonitor monitors the usage of system resources and logs the resource usage beyond a configured threshold, helping O&M personnel locate system resource exceptions.

In the key process monitoring scenario, the normal running of key service processes is critical. However, processes inevitably break down due to system- or human-related factors. sysmonitor detects process exceptions and quickly recovers key processes to ensure the normal running of services.

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 22.03 LTS SP2, support for the UDP protocol and related interfaces is added for Gazelle to enrich the user-mode protocol stack.

Feature Description



- 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.

Application Scenarios

Gazelle improves service performance for applications where the network protocol stack is a performance bottleneck.

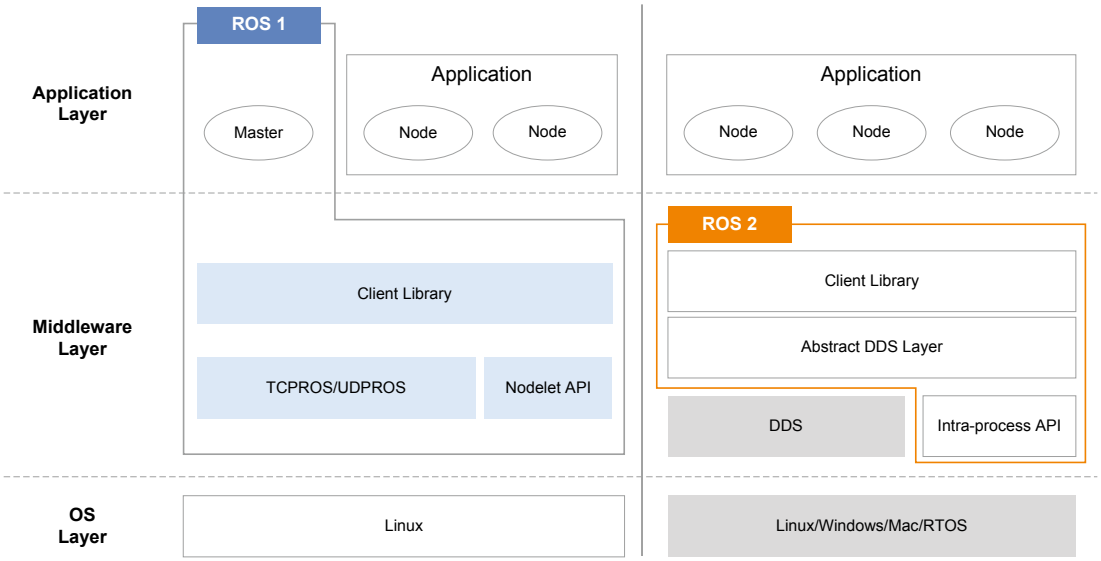
ROS

Robot Operating System (ROS) is a set of software libraries and tools designed to help you build robot applications. From drivers to algorithms and developer tools, ROS integrates standalone components to provide developers with a communication framework. ROS is not an actual OS, but a middleware that enables communication between the OS and developers' ROS applications. It can be seen as a runtime environment that runs on top of Linux, helping you efficiently organize and run robotic perception, decision making, and control algorithms.

ROS has two versions, ROS 1 and ROS 2. openEuler provides the ROS 2 communication framework and build tool.

Feature Description

The following figure illustrates the ROS architecture:



openEuler 22.03 LTS SP2 supports the ROS Humble version and provides the following features:

- All ros-core and ros-base software packages are available on the openEuler Server and Edge editions.
- SLAM is supported in the openEuler Embedded edition.
- ROS applications can be developed, built, and debugged based on openEuler. (The rqt series tools are supported, whereas RViz and Gazebo are not supported.)

Application Scenarios

Development of robot devices, such as humanoid robots, service robots, and industrial robots. You can install the required ROS function packages to develop your own robot applications.

openEuler WSL

Windows Subsystem for Linux (WSL) is an adaptation layer that allows you to run Linux user-mode software on Windows. You can download openEuler from the Microsoft Store, to enjoy a native experience on Windows.

Feature Description

- Out-of-the-box installation: On Windows devices that support WSL, you can download the latest openEuler LTS version from the Microsoft Store with just one click.
- Full lifecycle support: WSL applications of openEuler 22.03 LTS will be updated to openEuler 22.03 LTS SP2.
- User-friendly operations: The openEuler WSL package is available on the Microsoft Store, or you can build your own WSL applications using the open source code in the openEuler WSL repository.
- Metalink support: When you use DNF to install a software package on openEuler, the Metalink service guides DNF to download or update the software package from the mirror site near your IP address, increasing the download or update speed.

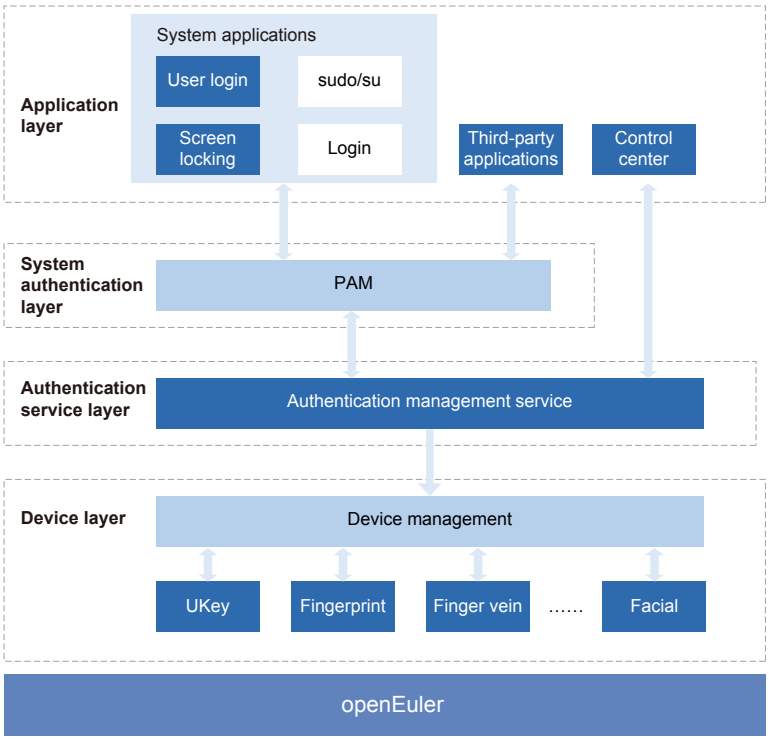
Application Scenarios

- Deploy and use an openEuler LTS version on Windows.
- Use Visual Studio Code and openEuler WSL to create a smooth cross-platform development experience.
- Build a Kubernetes cluster in openEuler WSL.
- Use openEuler command-line programs or scripts to process files and programs in Windows or WSL.

kiran-desktop 2.5

kiran-desktop 2.5 supports multi-factor authentication, which combines multiple authentication methods to authenticate users. The combination mode is OR or AND. In OR mode, users only have to pass one of the authentication methods, while in AND mode, users must pass all the authentication methods.

Feature Description



The control center performs the following functions:

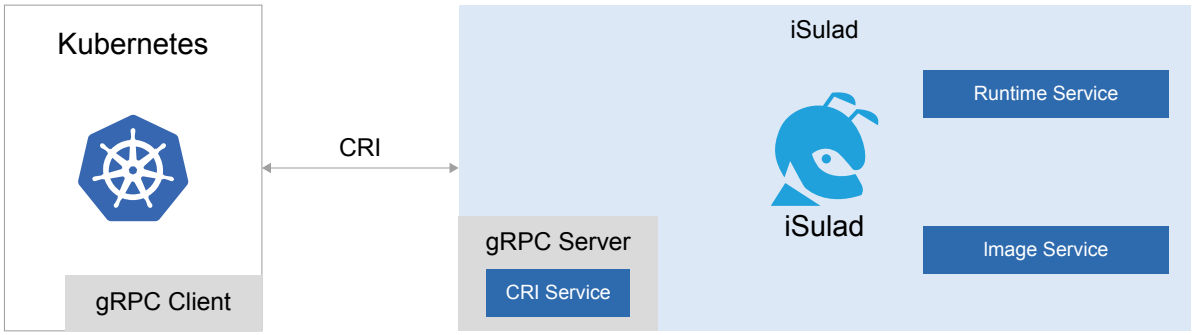
- Enables or disables an authentication method.
- Inputs and deletes features.
- Sets the default authentication device.
- Enables or disables the device drivers.
- Supports both graphical and non-graphical authentication.
- Graphical authentication scenarios include login, screen locking, and privilege escalation.
- Non-graphical authentication scenarios include sudo and su.
- Supports Ukey, iris, facial, fingerprint, and finger vein authentication.
- Provides the authentication management service to resolve the device preemption problem. When multiple authentication sessions are enabled, the authentication service layer can use a scheduling algorithm to enable these sessions to share the same authentication device.
- Provides the device management service. kiran-desktop 2.5 masks the differences between device interfaces and provides a unified device operation interface for upper-layer services.

iSulad for Kubernetes 1.24 and 1.25

It is a popular cloud native practice of using Kubernetes and a container engine to build cloud native applications. As a lightweight container engine, iSulad supports the northbound container runtime interface (CRI) and can be used as a container base for Kubernetes.

Feature Description

As the Kubernetes community evolves, the CRI standard has been continuously updated and iterated. To help improve the ecosystem and facilitate user operations, iSulad has upgraded its supported CRI versions. The v1alpha2 interface has been upgraded from 1.1X to 1.24/1.25.



The following interfaces are involved in the upgrade to version 1.25:

- Optimized interfaces: CreateContainer, UpdateContainerResources, ContainerStatus, ContainerStats, and ListContainerStats
- New interfaces: PodSandboxStats and ListPodSandboxStats

Application Scenarios

iSulad works in the scenarios that use Kubernetes 1.24/1.25 and a container engine. It also supports Kubernetes of earlier versions because the CRI is forward compatible.

sysMaster

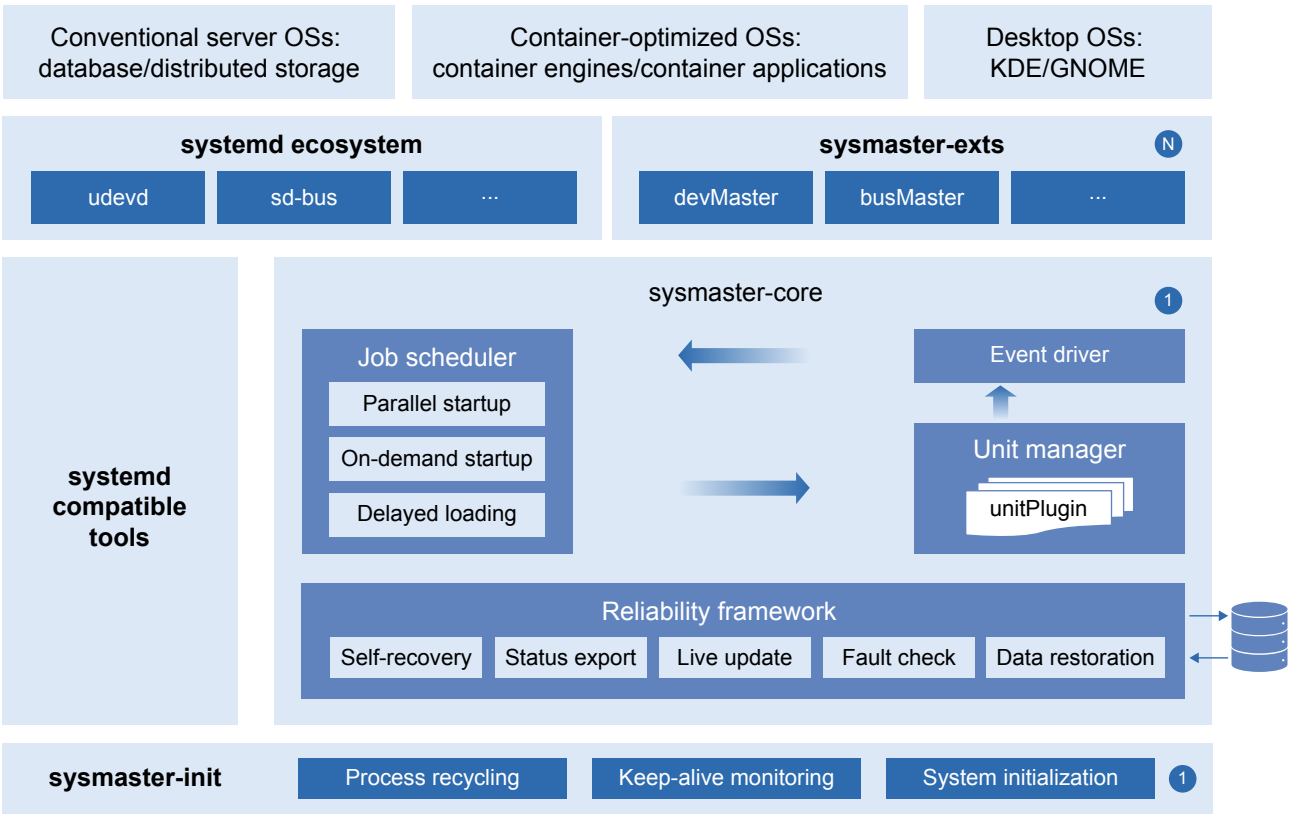
sysMaster is a collection of ultra-lightweight and highly reliable service management programs. It provides an innovative implementation of PID 1 to replace the conventional init process. Written in Rust, sysMaster is equipped with fault monitoring, second-level self-recovery, and quick startup capabilities, which help improve OS reliability and service availability.

sysMaster manages processes, containers, and VMs centrally and provides fault monitoring and self-healing mechanisms to help deal with Linux initialization and service management challenges. All these features make sysMaster an excellent choice for server, cloud computing, and embedded scenarios.

Feature Description

sysMaster divides the functions of traditional PID 1 into a 1+1+*N* architecture based on application scenarios. As shown in the figure, sysMaster consists of three components:

- sysmaster-init, which is a new implementation of PID 1, features simplified functions, a thousand lines of code (KLOC), and ultimate reliability. It is applicable to embedded systems with functions such as system initialization, zombie process recycling, and keep-alive monitoring.
- sysmaster-core undertakes the core service management functions and incorporates the reliability framework to enable live updates and quick self-recovery in the event of crashes, ensuring 24/7 service availability.
- sysmaster-extends offers a collection of components (such as devMaster for device management and busMaster for bus communication) that deliver key system functions, which are coupled in traditional PID 1. You can choose the components to use as required.



Featuring a simple component architecture, sysMaster improves the scalability and adaptability of the overall system architecture while reducing development and maintenance costs. sysMaster provides the following advantages:

- Live updates and self-recovery in seconds in the event of crashes
- Faster startup speed with lower memory overhead
- Plug-in-based service types that can be dynamically loaded as required
- Migration tools that provide seamless migration from systemd to sysMaster
- Unified interfaces that work with the iSulad container engine and QEMU for management of container and virtualization instances

sysMaster 0.2.4 released with openEuler 22.03 LTS SP2 supports system service management in the container scenario.

New features:

- The AArch64 and x86_64 architectures are supported.
- Unit service types of service and target are supported.
- More than 10 service units can be configured.
- The **sctl** command can be used to manage the lifecycle of a service.
- Logs can be exported to files.

Constraints:

- Only 64-bit OSs are supported.
- sysMaster can run only in system containers.
- sysMaster configuration files must be in TOML format.

In the future, sysMaster will extend to more scenarios and have its architecture and performance further optimized for higher scalability and adaptability. In addition, new features and components will be developed to meet the requirements of container, virtualization, and edge computing scenarios. These features will make sysMaster a powerful, efficient, and user-friendly system management framework.

Application Scenarios

sysMaster implements PID 1 in containers, VMs, servers, and edge devices.

OpenStack Train and OpenStack Wallaby Support

OpenStack is the world's most widely deployed open source cloud infrastructure platform that has been validated in large-scale production environments. OpenStack consists of a range of software components that provide common services for cloud infrastructure.

Feature Description

OpenStack has a complex structure and is difficult to deploy because of its various cloud services. To address this issue, the openEuler community has released RPM packages to help users easily deploy OpenStack on openEuler. With a few simple commands, you can quickly deploy an OpenStack cluster that suits your needs. Based on user feedback, the openEuler community releases a selection of popular OpenStack versions with each openEuler LTS version to provide services for a wide range of users with multi-version support.

openEuler 22.03 LTS SP2 supports the following services of OpenStack Train and OpenStack Wallaby:

- Keystone, Glance, Nova, Cinder, Neutron, Tempest, Horizon, Ironic, Placement, Trove, Kolla, Rally, Swift, Ceilometer, Heat, Aodh, Cyborg, and Gnocchi are supported by both OpenStack Train and OpenStack Wallaby.
- Designate, Manila, Barbican, Octavia, Cloudkitty, Masakari, Mistral, Senlin, and Zaqar are supported by OpenStack Wallaby only.

Application Scenarios

The OpenStack RPM packages allow you to quickly deploy OpenStack components provided by openEuler, thereby building a cloud computing platform based on openEuler. OpenStack is widely used in private and hybrid clouds, cloud computing services, academic research and development, and the testing and evaluation of cloud computing technologies and applications.

Lustre Client Software Package

Lustre is an open source software platform for parallel filesystems. It features high scalability, high performance, and high availability. Lustre runs on Linux and provides POSIX-compliant UNIX filesystem interfaces.

Feature Description

Lustre v2.15.2 client components have been released with openEuler 22.03 LTS SP2. Server components can be compiled from source. The latest source code of the Lustre community supports openEuler 22.03 LTS. By using the Lustre client, you can access the Lustre parallel filesystem from openEuler.

Application Scenarios

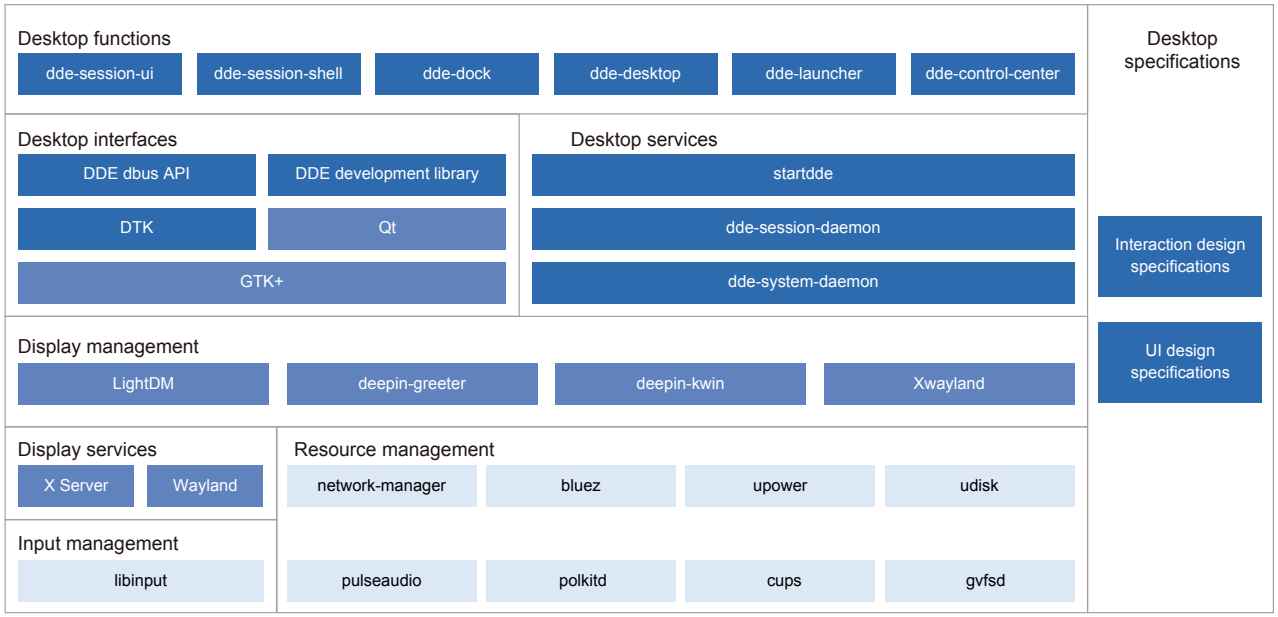
The Lustre parallel filesystem is suitable for systems that require massive and high-performance file storage. According to a community survey, Lustre is widely used in HPC and AI computing scenarios in industries such as academic research, scientific computing, media, manufacturing, finance, and education.

DDE

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.



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.

No-SVA Support of KAE

The Kunpeng Accelerator Engine (KAE) is an acceleration solution based on Kunpeng hardware capabilities. It contains the KAE encryption and decryption module and the KAE zlib compression and decompression module, which accelerate SSL and TLS applications and data compression, reduce processor usage, and boost processor efficiency. In addition, the application layer of KAE masks the internal implementation details, thereby allowing users to quickly migrate services through the standard interfaces of OpenSSL and zlib.

Feature Description

KAE Encryption and Decryption

The KAE encryption and decryption module uses the Kunpeng hardware acceleration engine to implement the RSA, SM3, SM4, DH, MD5, and AES algorithms. It provides high-performance symmetric and asymmetric encryption and decryption based on the lossless user-mode driver framework. It is compatible with OpenSSL 1.1.1a and later versions and supports both synchronous and asynchronous mechanisms.

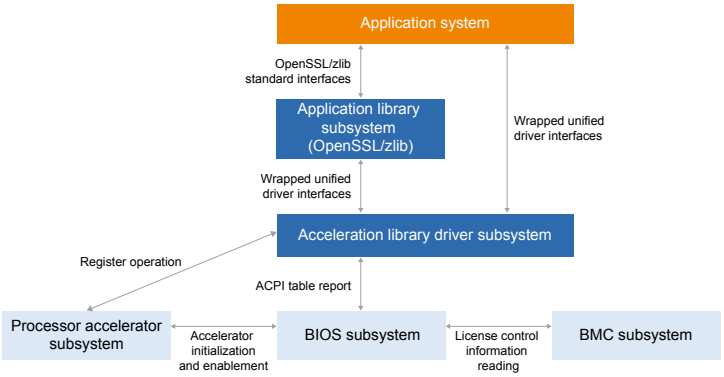
KAE supports the following algorithms:

- Digest algorithms SM3 and MD5, supporting asynchronous models.
- Symmetric encryption algorithm SM4, supporting asynchronous models and CTR, XTS, CBC, ECB, and OFB modes.
- Symmetric encryption algorithm AES, supporting asynchronous models and ECB, CTR, XTS, and CBC modes.
- Asymmetric algorithm RSA, supporting asynchronous models and key sizes 1024, 2048, 3072, and 4096.
- Key negotiation algorithm DH, supporting asynchronous models and key sizes 768, 1024, 1536, 2048, 3072, and 4096.

KAEzip

KAEzip is the compression and decompression module of KAE. It uses the Kunpeng hardware acceleration module to implement the Deflate algorithm and works with the lossless user-mode driver framework to provide an interface for high-performance compression in Gzip or zlib format.

- zlib and Gzip formats are supported, complying with RFC1950 and RFC1952 specifications.
- The Deflate algorithm is supported.
- The synchronous mode is supported.
- A single Kunpeng 920 processor can use KAEzip to achieve a maximum compression bandwidth of 7 GB/s and a maximum decompression bandwidth of 8 GB/s.
- The supported compression ratio is around 2, which is the same as the zlib 1.2.11 interface.



Application Scenarios

The KAE encryption and decryption module provides high-performance symmetric and asymmetric encryption and decryption. It is compatible with OpenSSL 1.1.1a and later versions and supports both synchronous and asynchronous mechanisms.

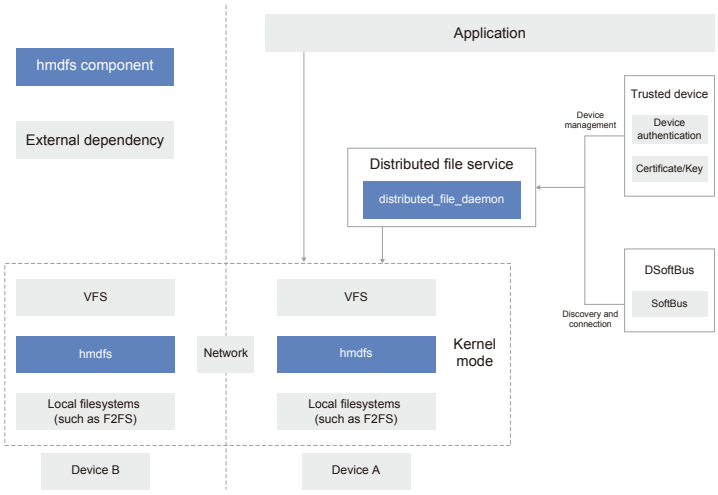
KAE can be used to improve application performance in different scenarios. For example, in distributed storage scenarios, the zlib library is used to accelerate data compression and decompression.

hmdfs Based on Soft Bus

hmdfs stands for HarmonyOS Distributed File System. It is a soft bus-based distributed filesystem ported from the OpenHarmony community. hmdfs provides a globally consistent access view for each device dynamically connected to a network via distributed soft bus (DSoftBus) and allows you to implement high-performance read and write operations on files with low latency by using basic filesystem APIs.

Feature Description

- distributed_file_daemon: user-mode daemon for distributed file management, which controls networking interfaces of access devices, mounts hmdfs, and manages permissions.
- Trusted device: manages the trust relationships between the local device and other devices established for different services in a unified manner.
- DSoftBus: discovers and connects devices at the network link layer.
- Virtual filesystem (VFS): kernel-mode software abstraction layer between users and filesystems on physical storage media.
- hmdfs: core module of the distributed filesystem. It is a high-performance layered filesystem in kernel mode for mobile distributed scenarios.



File Views

- Unified view: hmdfs integrates local filesystem directories of different devices into a unified view for users to view files across devices in a unified manner.
- Device-based view: The device-based view enables users and applications to view the directories of each device, decouples local and remote operations, and supports remote file creation.

Data Synchronization Method

hmdfs adopts a subscription-publication design.

- Subscriptions and publications are cached and then synchronized on demand. By default, only directory trees are synchronized. Subscription relationships are accessed and managed based on directory trees.
- Consistency is ensured by an event-triggered and timeout-based consistency check mechanism.
- The lazy loading mode is used, in which only expiration notifications are sent to subscribers. Metadata is obtained upon the next access.

Application Scenarios

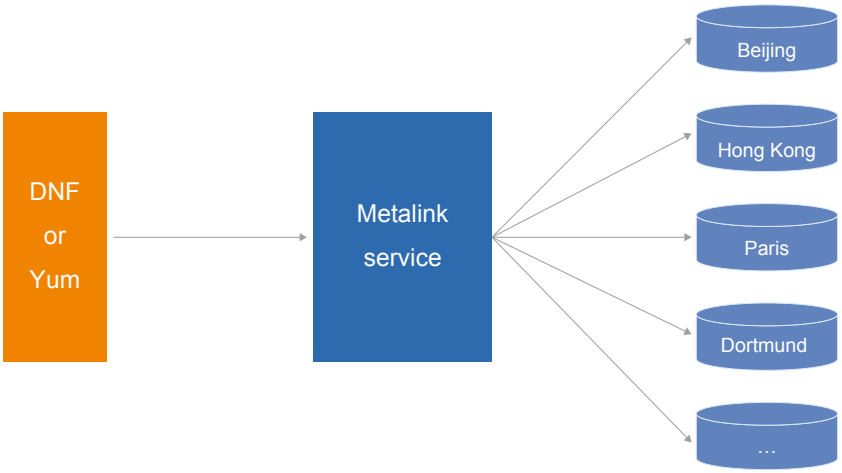
The openEuler-based DSoftBus technology is integrated into the industrial interconnection bus SDK to synchronize data (collected images and configuration information) from embedded devices in an industrial production line to edge servers in real time for processing.

Automatic Optimization for Software Package Downloads

30 openEuler mirror sites are distributed across Asia, Europe, and North America. Software packages can be downloaded from the nearest mirror sites to improve the download speed.

Feature Description

A metalink, whose value is the URL of the API provided by the metalink service, is configured in the DNF or Yum configuration file shipped with openEuler releases. When a user tries to download a software package, the DNF or Yum client sends a request to the metalink URL. The metalink service returns data in XML format that contains the addresses of nearest mirror sites. The DNF or Yum client then selects the optimal site from the addresses to download the software package, ensuring a fast download speed.



Application Scenarios

Users use DNF or Yum to download software packages.

Copyright Statement 08

09 Trademark

All trademarks and logos used and displayed on this document are all owned by the openEuler community, except for trademarks, logos, and trade names that are owned by other parties. Without the written permission of the openEuler community or other parties, any content in this document shall not be deemed as granting the permission or right to use any of the aforementioned trademarks and logos by implication, no objection, or other means. Without prior written consent, no one is allowed to use the name, trademark, or logo of the openEuler community in any form.

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


OpenEuler