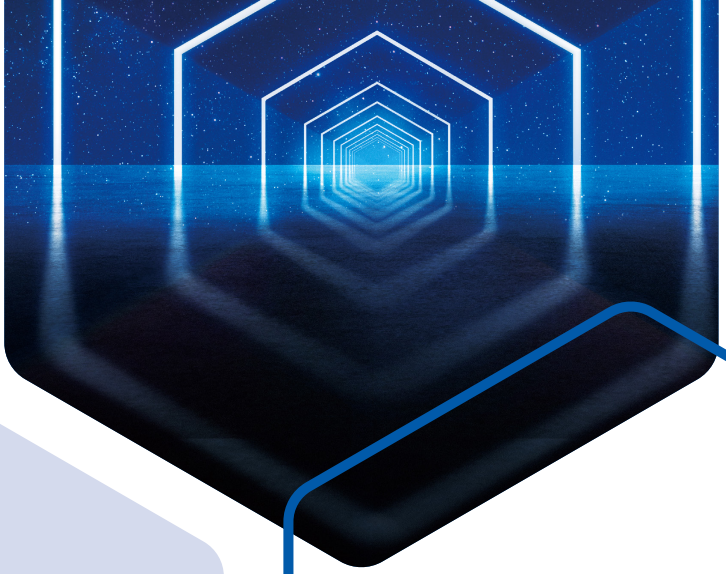
openEuler 25.03

Technical White Paper



CONTENTS

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.

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 features for server, cloud native, edge computing, and embedded scenarios.

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

On June 30, 2024, openEuler 22.03 LTS SP4 was released. Designed to improve developer efficiency, it further extends features for server, cloud native, edge computing, and embedded scenarios.

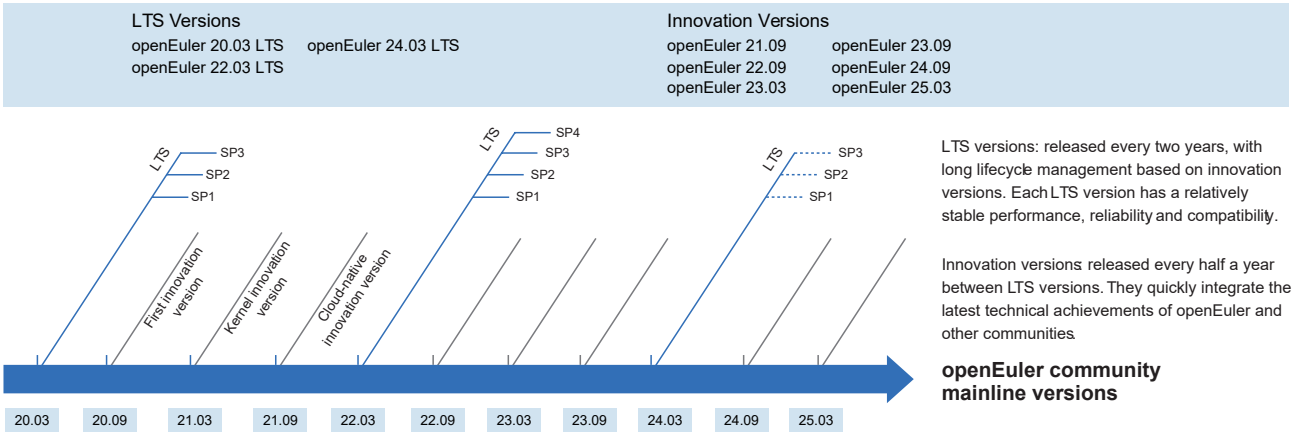
On September 30, 2024, openEuler 24.09 was released. It is an innovation version built based on Linux kernel 6.6 and brings more advanced features and functions.

On December 30, 2024, openEuler 24.03 LTS SP1 was released. This enhanced and extended version of openEuler 24.03 LTS

is developed on Linux kernel 6.6 and designed for server, cloud, edge computing, and embedded deployments. It offers new features and enhanced functions that streamline processes across a range of domains.

More recently on March 30, 2025, openEuler 25.03 was released. It is an innovation version designed based on Linux kernel 6.6 and is suited for server, cloud, edge, and embedded scenarios. It provides a variety of new features and functions and brings brand-new experience to developers and users in diverse industries.

openEuler Version Management

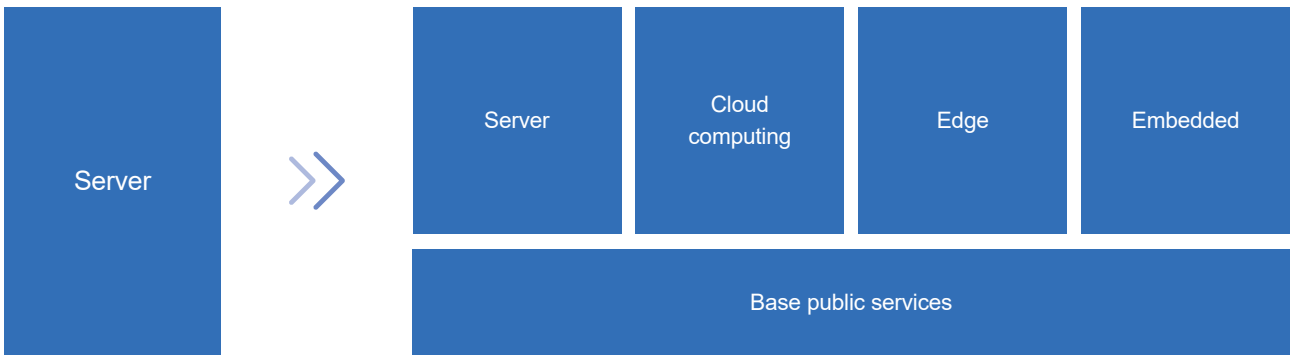


As an OS platform, openEuler releases an LTS version every two years. Each LTS version provides enhanced specifications and a secure, stable, and reliable OS for enterprise users.

openEuler is built on tried-and-tested technologies. A new openEuler innovation 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, LoongArch, and PowerPC), 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: 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.

Platform Architecture 02

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 25.03 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 25.03 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

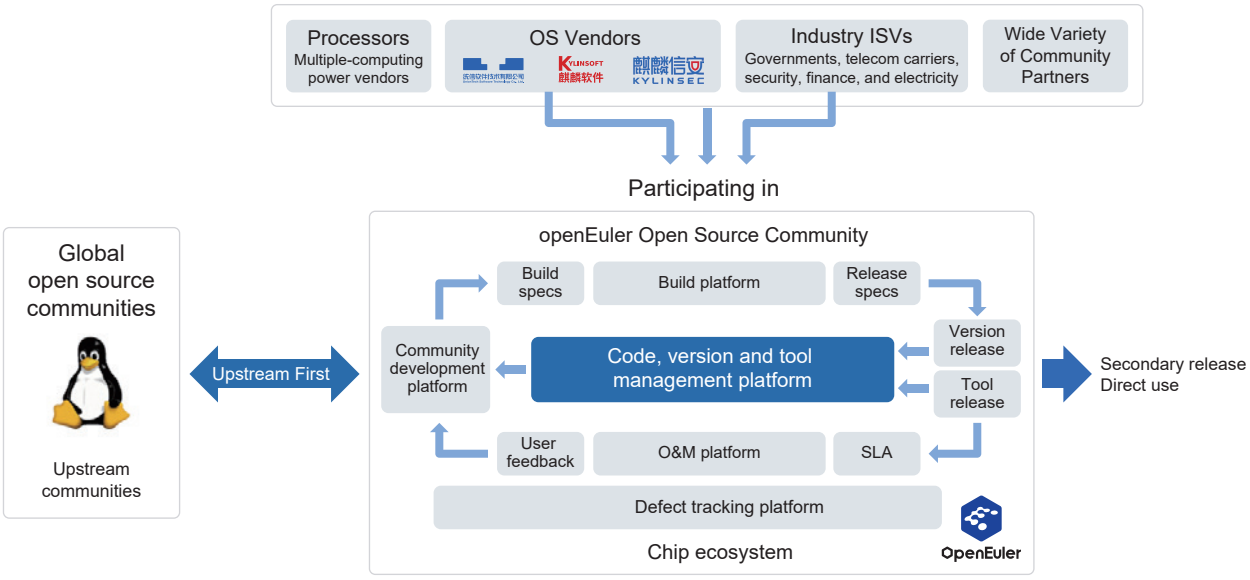
- **Edge computing**: openEuler 25.03 Embedded is released for edge computing scenarios. It integrates the KubeEdge+ edge-cloud collaboration framework to provide unified management, provisioning of edge and cloud applications, and other capabilities.
- **Embedded**: openEuler 25.03 Embedded is released for embedded scenarios, helping compress images to under 5 MB and shorten the image loading time to under 5 seconds.
- **AI OS**: The OS enables AI software stacks with out-of-the-box availability. Heterogeneous convergence of memory, scheduling, and training/inference resources reduces AI development costs and improves efficiency. The intelligent interaction platform of the OS streamlines development and administration.

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

Visit https://www.openeuler.org/en/compatibility/ to see the full hardware list.

03 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, RISC-V
Memory	≥ 4 GB
Drive	≥ 20 GB

VMs

Verify VM compatibility when installing openEuler.

Hosts running on openEuler 25.03 support the following software packages:

- libvirt-9.10.0-12.oe2409
- libvirt-client-9.10.0-12.oe2409
- libvirt-daemon-9.10.0-12.oe2409
- qemu-8.2.0-17.oe2409
- qemu-img-8.2.0-17.oe2409

openEuler 25.03 is compatible with the following guest OSs for VMs:

Host OS	Guest OS	Architecture
openEuler 25.03	CentOS 6	x86_64
openEuler 25.03	CentOS 7	AArch64
openEuler 25.03	CentOS 7	x86_64
openEuler 25.03	CentOS 8	AArch64
openEuler 25.03	CentOS 8	x86_64
openEuler 25.03	Windows Server 2016	x86_64
openEuler 25.03	Windows Server 2019	x86_64
openEuler RISC-V 25.03	Ubuntu 24.10	RISC-V
openEuler RISC-V 25.03	Fedora 41	RISC-V

Item	Configuration Requirement
Architecture	AArch64, x86_64, or RISC-V
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, or RISC-V
Memory	≥ 4 GB
Drive	≥ 20 GB

Embedded Devices

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

Item	Configuration Requirement
Architecture	AArch64, AArch32, or x86_64
Memory	≥ 512 MB
Drive	≥ 256 MB

Scenario-specific Innovations 04

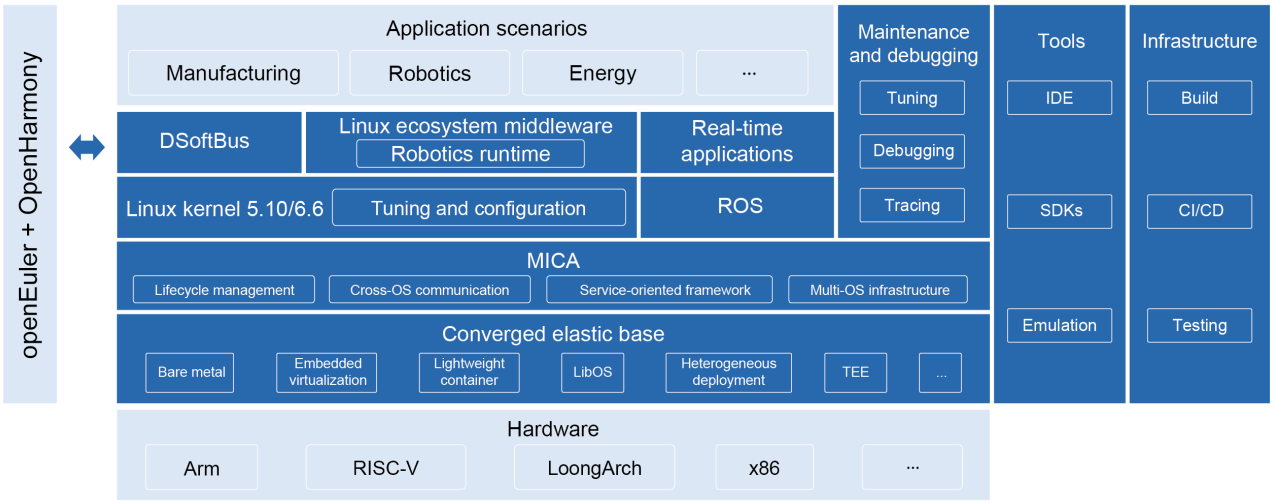
openEuler Embedded

openEuler 25.03 Embedded is designed for embedded applications, offering significant progress in southbound and northbound ecosystems as well as infrastructure over previous generations.

openEuler Embedded provides a closed loop framework often found in operational technology (OT) applications such as manufacturing and robotics, whereby innovations help optimize its embedded system software stack and ecosystem. In the southbound ecosystem, the openEuler Embedded ecosystem development initiative has strengthened for hardware such as Kunpeng 920 and TaiShan, while collaborating with STMicroelectronics and MYIR to adapt the high-performance STM32MP257 microprocessor for industrial applications. In the northbound ecosystem, openEuler Embedded has enriched its industrial and graphics middleware capabilities to promote adoption in industrial manufacturing and robotics.

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

System Architecture



Southbound Ecosystem

openEuler Embedded Linux supports mainstream processor architectures like AArch64, x86_64, AArch32, and RISC-V, and will extend support to LoongArch in the future. openEuler 24.03 and later versions have a rich southbound ecosystem and support chips from Raspberry Pi, HiSilicon, Kunpeng, Rockchip, Renesas, TI, Phytium, StarFive, Allwinner, and STMicroelectronics.

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 is available as two community hypervisors, ZVM (for real-time VM monitoring) and Rust-Shyper (for Type-I embedded VM monitoring).

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 or Zephyr) in partitioning-based virtualization mode

Northbound Ecosystem

- **Northbound software packages**: Over 600 common embedded software packages can be built using openEuler.
- **Soft real-time kernel**: This capability helps respond to soft real-time interrupts within microseconds.
- **DSoftBus**: The distributed soft bus system (DSoftBus) of openEuler 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.

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:

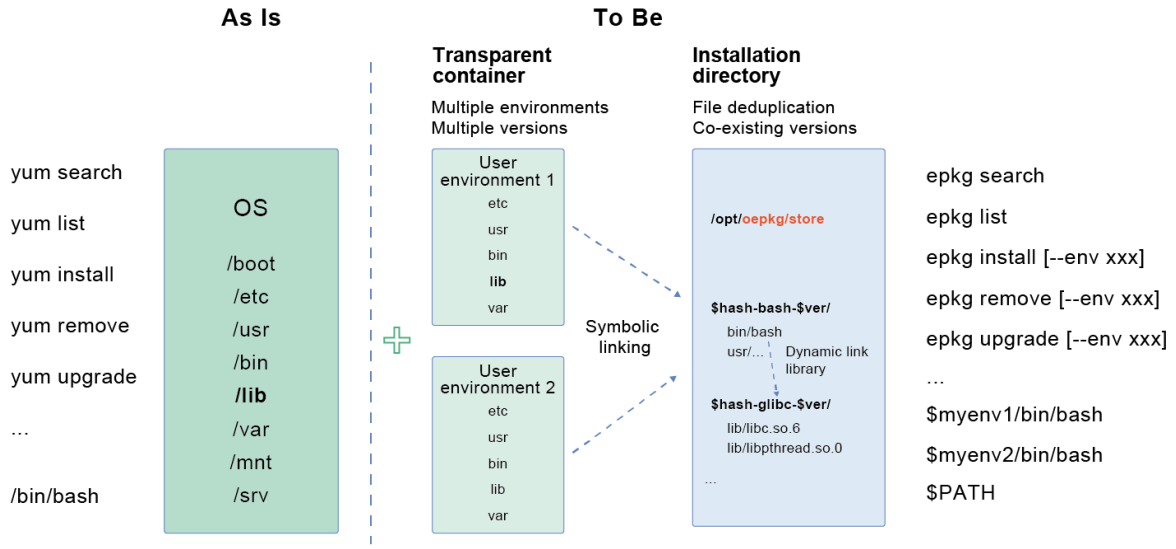
- Compatible with processor architectures like Cortex-M, AArch64, x86_64, and riscv64, and supports M4, RK3568, RK3588, x86_64, Hi3093, Raspberry Pi 4B, Kunpeng 920, Ascend 310, and Allwinner D1s.
- Connects with openEuler Embedded Linux on Raspberry Pi 4B, Hi3093, RK3588, and x86_64 devices in bare metal mode.
- Can be debugged using GDB on openEuler Embedded Linux.

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.

epkg

epkg is a new software package manager that supports the installation and use of non-service software packages. It solves version compatibility issues so that users can install and run software of different versions on the same OS by using simple commands to create, enable, and switch between environments.



Feature Description

Version compatibility: Allows multi-version installation, resolving version conflicts.

Flexible installation modes: Supports both privileged (system-wide) and unprivileged (user-specific) installations, enabling minimal-footprint deployments and self-contained installations.

Environment management: Supports environment lifecycle operations (create, delete, activate, register, and view), implements multi-version capabilities by configuring multiple environments with distinct software repositories, and enables runtime registration for multiple environments and exclusive environment activation for development debugging.

Environment rollback: Maintains operational history tracking and provides state restoration capabilities, allowing recovery from misoperations or faulty package installations.

Package management: Implements core package operations (install, remove, and query) with RPM/DNF-level functionality parity, meeting the daily usage requirements for typical users and scenarios.

Application Scenarios

epkg solves compatibility issues in installing multiple versions of the same software package. Users can switch between environments to use different package versions.

For details, refer to the epkg User Guide.

GCC Compilation and Linking Acceleration

To improve the compilation efficiency of openEuler software packages and enhance CI pipeline and developer productivity, optimization techniques for C/C++ components are implemented through compiler and linker enhancements.

Feature Description

GCC with profile-guided optimization (PGO) and link time optimization (LTO), alongside the modern mold linker, cuts 9.5% off the total compilation time for top 90+ software packages by accelerating C/C++ library compilation. The following key capabilities are supported:

- GCC 12.3 is configured to generate binaries with PGO and LTO, accelerating compilation.
- Applications specified in the trustlist can automatically switch to the mold linker to optimize linking efficiency.

Application Scenarios

This feature is ideal for compiling applications with a large number of C/C++ components. PGO and LTO accelerate C/C++ code compilation, thereby improving application development efficiency.

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

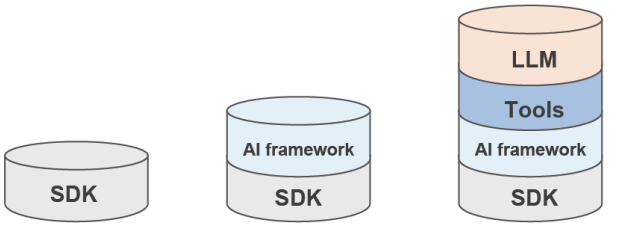
openEuler offers an efficient development and runtime environment that containerizes software stacks of AI platforms with out-of-the-box availability. It also provides various AI frameworks that encourage AI development.

Out-of-the-Box Availability

Feature Description

openEuler supports TensorFlow, PyTorch, and MindSpore 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 AI applications easier to develop and run.

Environment setup is further simplified by containerizing software stacks. openEuler provides three types of container images:



SDK image | AI framework image | Model application image

- **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 an SDK image as the base and install AI framework software, such as PyTorch and TensorFlow. You can use an AI framework image to quickly build a distributed AI framework, such as Ray.
- **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 are ideal for performing 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.

sysHAX

Feature Description

The sysHAX large language model (LLM) heterogeneous acceleration runtime enhances model inference performance in single-server, multi-xPU setups by optimizing Kunpeng + xPU (GPU/NPU) resource synergy.

- **Operator pushdown acceleration**: Adapts to vLLM v0.6.6 to optimize the scheduling engine, shortening latency on CPUs.
- **CPU inference acceleration**: Improves CPU throughput via NUMA-aware scheduling, parallelized matrix operations, and SVE-optimized inference operators.

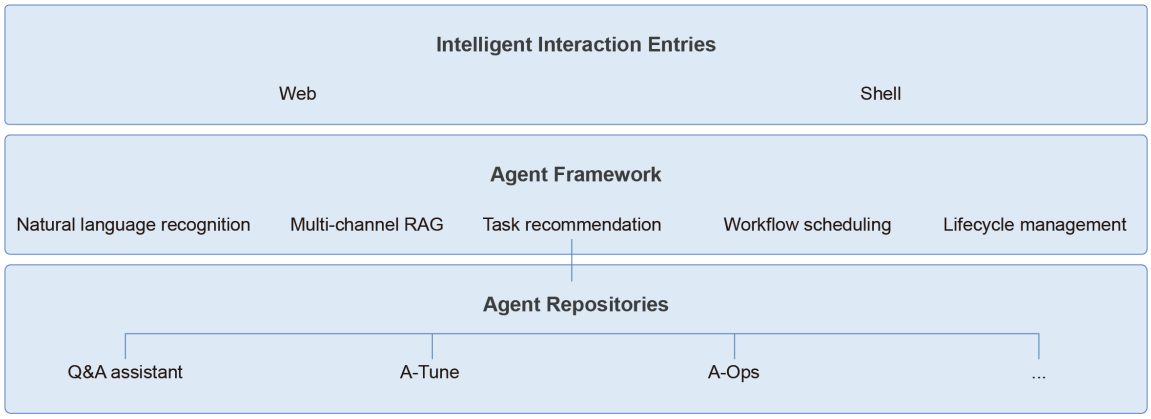
Application Scenarios

sysHAX is used to optimize Transformer models including DeepSeek, Qwen, Baichuan, and Llama. Its CPU inference acceleration capability has been adapted to DeepSeek 7B, 14B, and 32B and Qwen series models. sysHAX fits into the following application scenario:

Data centers: sysHAX assigns inference tasks to CPUs to fully utilize CPU resources and increase the concurrency and throughput of LLMs.

AI for OS

AI is making openEuler smarter. openEuler Intelligence is an intelligent Q&A platform developed on LLMs and openEuler data. It is designed to streamline code generation, troubleshooting, and O&M.



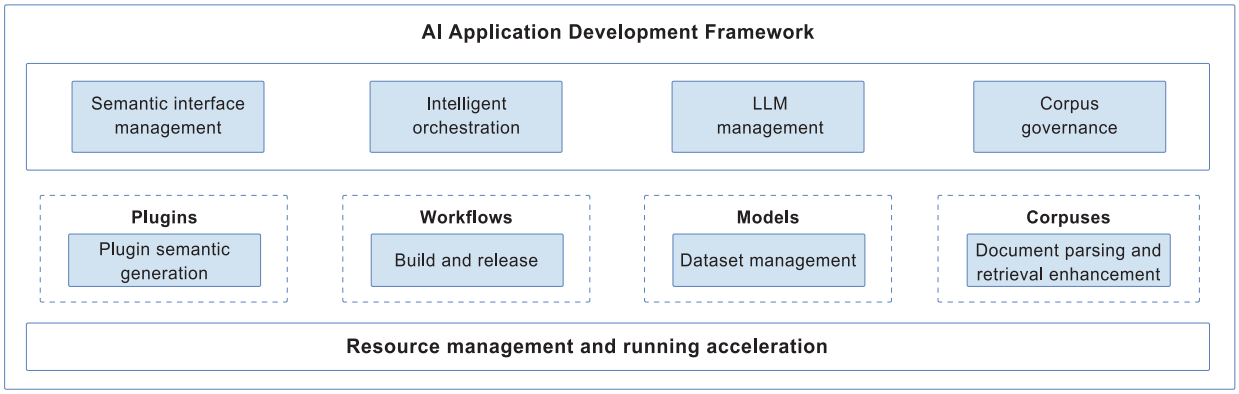
AI Application Development Framework

LLM applications are key to enterprises' AI adoption. Combining retrieval-augmented generation (RAG) with LLMs effectively addresses gaps in domain-specific data. AI application development platforms that incorporate the RAG technique are a great help to enterprises and developers. To satisfy this need, the openEuler community incubates an AI application development framework.

Feature Description

This framework provides intelligent AI development tools for individuals and enterprises. It gives developers greater control by lowering technical barriers, improving efficiency and data quality, and meeting diverse needs for data processing, model training, and content management. This framework provides the following features:

- **Multi-channel RAG**: Overcomes limitations of traditional native RAG (low accuracy and weak guidance) using techniques like corpus governance, prompt rewriting, and multi-channel retrieval comparison.
- **Document processing and optimization**: Allows incremental corpus updates, deduplication, sensitive data masking, and standardization (such as summarization, code annotation, and case organization).
- **Embedding model fine-tuning**: Tunes and evaluates embedding models (such as BGE models) quickly for domain-specific performance gains.



Application Scenarios

The AI application development framework is suitable for a wide range of intelligent development and data processing scenarios.

- **Data preprocessing and labeling**: Quickly generates high-quality Q&A data for AI model training, reducing the time spent on manual labeling.
- **Document quality improvement**: Efficiently anonymizes, deduplicates, and standardizes documents to strengthen document information security and consistency.
- **Industry-specific model building**: Fine-tunes embedding models to create high-performance models tailored for specific industries and use cases, enhancing model relevance and effectiveness.

Intelligent Q&A

Feature Description

The openEuler Intelligence system is accessible via web or shell.

- **Web**: The easy-to-use chatbot provides access to openEuler knowledge, community announcements, and O&M solutions. It uses various intelligent agents to deliver user-friendly functionality.
- **Shell**: This method allows users to interact with openEuler using human language, providing heuristic system tuning and fault diagnosis for easier system management.

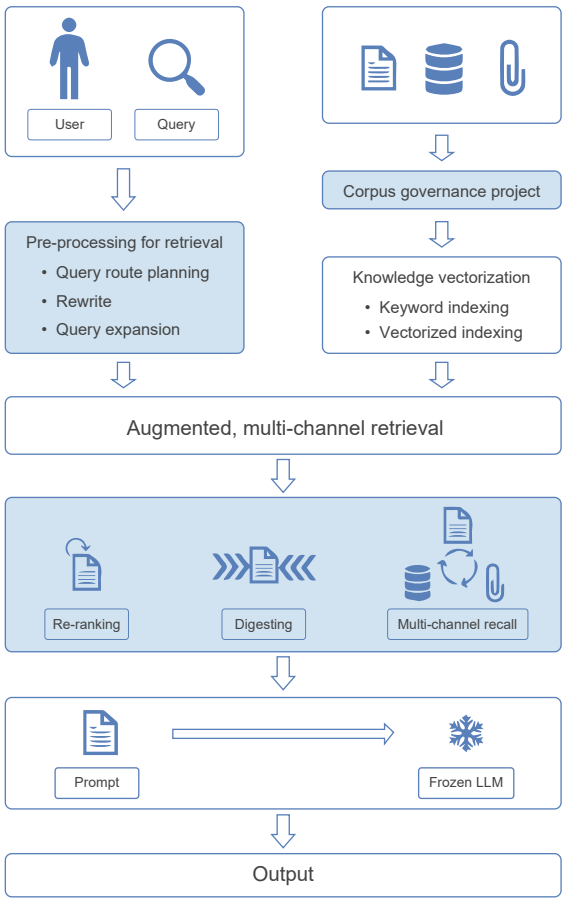
Workflow Scheduling

- **Atomic agent operations**: Multiple agent operations can be combined into a multi-step workflow that is internally ordered and associated, and is executed as an inseparable atomic operation. By supporting user-defined error recovery, this design ensures consistent, traceable, and stable operations when a complex task is executed.
- **Real-time data processing**: Data generated in each step of the workflow can be processed immediately and then transferred to the next step. Users can transfer information and access required data between steps with ease. The final results can be displayed on the front end in multiple forms, such as reports, images, and code snippets.
- **Intelligent interaction**: When openEuler Intelligence receives a vague or complex user instruction, it proactively asks the user to clarify and provide more details. With new information, the system continues to execute the workflow and meet the user's expectations.

Task Recommendation

- **Intelligent response**: openEuler Intelligence can analyze the semantic information entered in text format, determining the user's intent and selecting the most matched workflow. The system learns and adapts to users' operation habits and preferences to improve user experience.
- **Intelligent guidance**: openEuler Intelligence comprehensively analyzes the execution status, function requirements, and associated tasks of the current workflow, and provides next-step operation suggestions based on users' personal preferences and historical patterns. This helps inform user decisions and improves task efficiency on an intuitive front-end display.

RAG



Retrieval-augmented generation (RAG) is a technique for enhancing the long-term memory capability of large language models (LLMs). It is used in the openEuler Intelligence system to reduce model training costs. The RAG technique has the following highlights:

- **Pre-processing for retrieva**l: Users' compound query requests are rewritten and routed based on a preset plan.
- **Knowledge indexing**: Keywords are extracted and text is vectorized for diversified document content, and indexes can be built for fragments.
- **Multi-channel recall**: Vectorized retrieval, keyword retrieval, and Chat2DB capabilities are provided for diversified knowledge sources. Retrieval results can be re-ranked, filtered, and converged.

These capabilities enable RAG for the openEuler Copilot System to accommodate to more document formats and content, and enhance Q&A services without increasing system load.

Corpus Governance

Corpus governance is a core RAG capability. It imports corpuses into the knowledge base in a supported format using fragment relationship extraction, fragment derivative construction, and optical character recognition (OCR). This increases the retrieval hit rate.

- **Fragment relationship extraction**: In scenarios where document content continuity must be maintained, relative relationships between fragments are retained to associate context.
- **Fragment derivative generation**: Digests are provided for complicated fragments, making it possible to hit fragments through these digests.
- **OCR**: OCR and contextual digests are available for hybrid image and text scenarios, which allow extracting and abstracting text from images.

Corpus governance features enhance the Q&A experience in multi-round dialogs, content integrity, and image-text display.

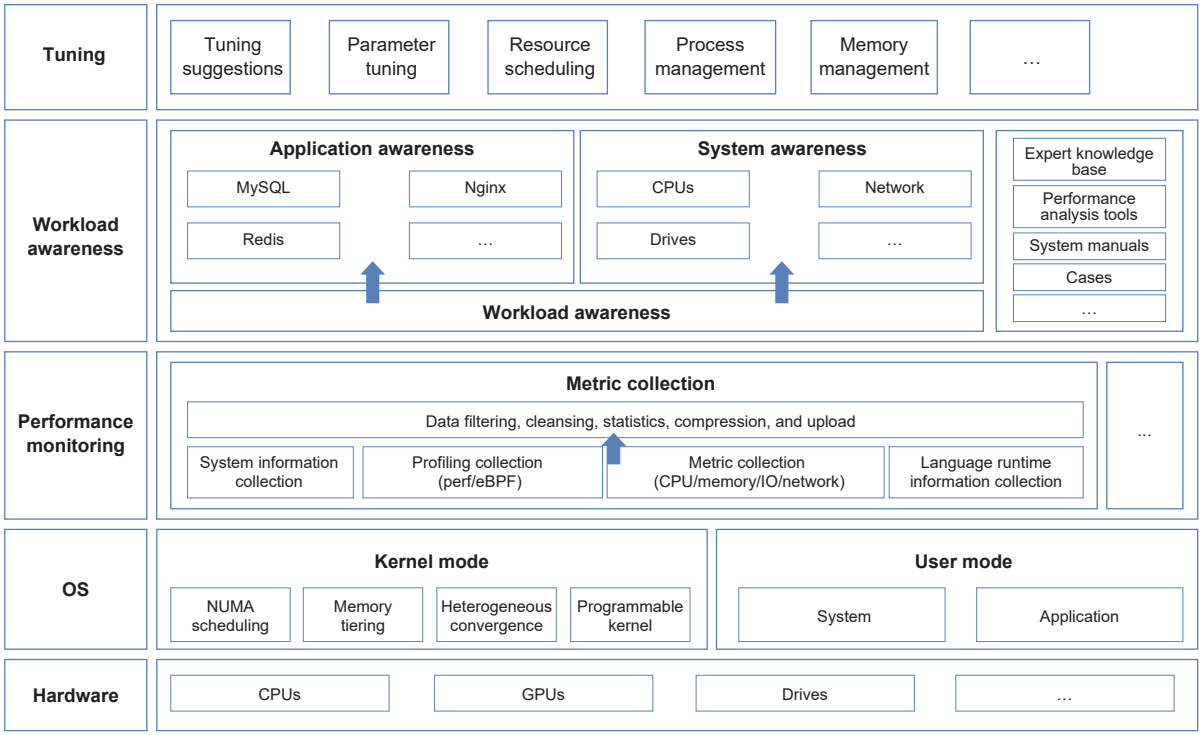
Application Scenarios

- **Common users** who are seeking best practices, such as porting applications to openEuler.
- **Developers** who want to learn contribution processes, key features, project development, and other extensive knowledge of openEuler.
- **O&M personnel** who aim to solve common problems and improve system management based on Q&A system suggestions.

Intelligent Tuning

Feature Description

The openEuler Intelligence system supports the intelligent shell entry. Through this entry, you can interact with openEuler Intelligence using a natural language and perform heuristic tuning operations such as performance data collection, system performance analysis, and system performance tuning.

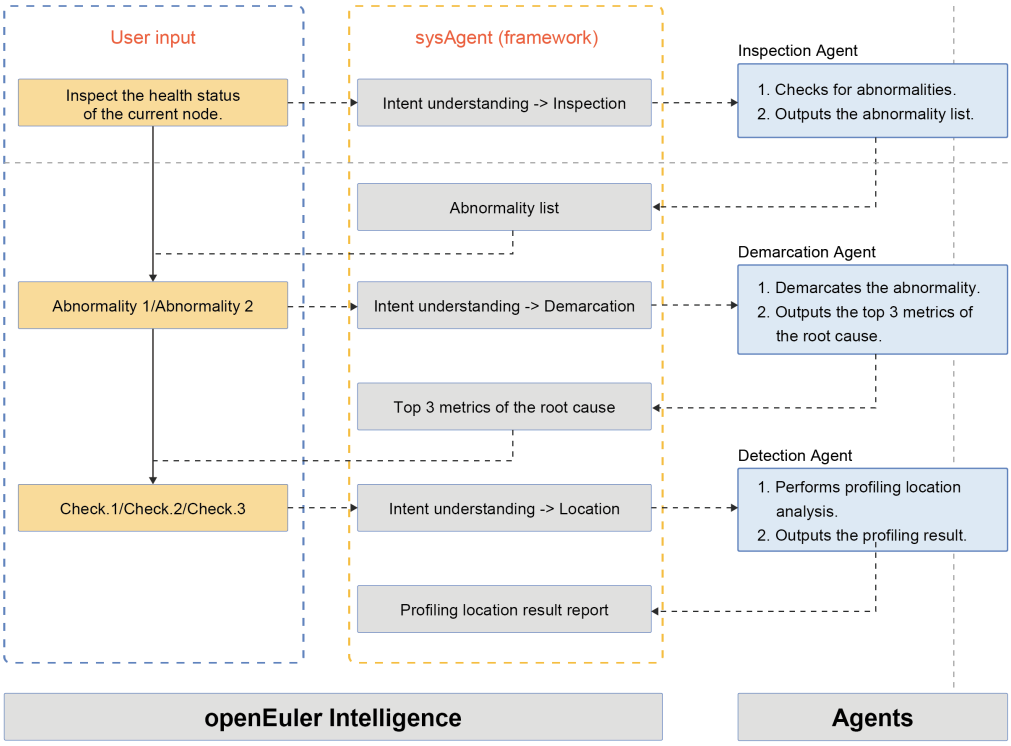


Application Scenarios

- **Gaining insights from key performance metrics**: You can learn about the system performance status based on collected performance metrics like CPU, I/O, drive, network, and application.
- **Analyzing system performance**: Performance analysis reports are generated, making it easier to locate performance bottlenecks across the entire system and in individual applications.
- **Receiving performance tuning suggestions**: The openEuler Copilot System generates a performance tuning script, which can be executed with one click to tune the system and specific applications.

Intelligent Diagnosis

Feature Description



- **Inspection**: The Inspection Agent checks for abnormalities of designated IP addresses and provides an abnormality list that contains associated container IDs and abnormal metrics (such as CPU and memory).
- **Demarcation**: The Demarcation Agent analyzes and demarcates a specified abnormality contained in the inspection result and outputs the top 3 metrics of the root cause.
- **Location**: The Detection Agent performs profiling location analysis on the root cause, and provides useful hotspot information such as the stack, system time, and performance metrics related to the root cause.

Application Scenarios

In openEuler 25.03, the intelligent shell entry enables capabilities like single-node abnormality inspection, demarcation, and profiling location.

- The inspection capabilities refer to single-node performance metric collection, performance analysis, and abnormality inspection.
- The demarcation capability is to locate the root cause based on the abnormality inspection result and output the top 3 metrics of the root cause.
- The profiling location capability refers to using a profiling tool to locate the faulty modules (code snippets) based on the root cause.

Intelligent Vulnerability Patching

Feature Description

The openEuler intelligent vulnerability patching tool automates vulnerability management and fixing for the openEuler kernel repository. It analyzes the impact of vulnerabilities on openEuler versions using the **/analyze** command and creates a patching pull request within minutes via the **/create_pr** command.

Application Scenarios

The openEuler intelligent vulnerability patching tool is used in a variety of system security maintenance and routine kernel management scenarios.

- **Vulnerability impact scope analysis**: Automatically analyzes the impact of Common Vulnerabilities and Exposures (CVE) on each version in the openEuler kernel repository and determines the vulnerability introduction and repair status.
- **Vulnerability patch linking**: Intelligently matches introduced vulnerabilities with patch links, helping maintenance personnel quickly locate code snippets that need to be repaired.
- **Automatic pull request creation**: Automatically creates vulnerability patches and pull requests to facilitate updates to target versions in the openEuler kernel repository.
- **Intelligent patch conflict detection**: Automatically identifies and warns compatibility conflicts between patches and existing code to ensure smooth and effective patching.

Intelligent Container Images

Feature Description

The openEuler Intelligence system can invoke environment resources through a natural language, assist in pulling container images for local physical resources, and establish a development environment suitable for debugging on existing compute devices.

This system supports three types of containers, and container images have been released on Docker Hub. You can manually pull and run these container images.

- **SDK layer**: encapsulates only the component libraries that enable AI hardware resources, such as CUDA and CANN.
- **SDKs + training/inference frameworks**: accommodates TensorFlow, PyTorch, and other frameworks (for example, tensorflow2.15.0-cuda12.2.0 and pytorch2.1.0.a1-cann7.0.RC1) in addition to the SDK layer.
- **SDKs + training/inference frameworks + LLMs**: encapsulates several models (for example, llama2-7b and chatglm2-13b) based on the second type of containers.

The following table lists the container images supported by the openEuler Intelligence system:

Registry	Repository	Image Name	Tag
docker.io	openeuler	cann	8.0.RC1-oe2203sp4
			cann7.0.RC1.alpha002-oe2203sp2
docker.io	openeuler	oneapi-runtime	2024.2.0-oe2403lts
docker.io	openeuler	oneapi-basekit	2024.2.0-oe2403lts
docker.io	openeuler	llm-server	1.0.0-oe2203sp3
docker.io	openeuler	mlflow	2.11.1-oe2203sp3
			2.16.2-oe2203sp3
			2.17.0rc0-oe2203sp3
			2.17.1-oe2203sp1
			2.17.1-oe2203sp3
			2.17.2-oe2203sp1
			2.17.2-oe2203sp3
			2.17.2-oe2203sp4
			2.18.0-oe2203sp1
			2.18.0-oe2403lts
			2.19.0-oe2203sp1
			2.19.0-oe2203sp3
			2.19.0-oe2203sp4
			2.19.0-oe2403lts
			2.17.1-oe2403lts
			2.17.2-oe2403lts
docker.io	openeuler	llm	chatglm2_6b-pytorch2.1.0.a1-cann7.0.RC1.alpha002-oe2203sp2
			llama2-7b-q8_0-oe2203sp2
			chatglm2-6b-q8_0-oe2203sp2
			fastchat-pytorch2.1.0.a1-cann7.0.RC1.alpha002-oe2203sp2

docker.io	openeuler	tensorflow	tensorflow2.15.0-oe2203sp2
			tensorflow2.15.0-cuda12.2.0-devel-cudnn8.9.5.30-oe2203sp2
docker.io	openeuler	pytorch	pytorch2.1.0-oe2203sp2
			pytorch2.1.0-cuda12.2.0-devel-cudnn8.9.5.30-oe2203sp2
			pytorch2.1.0.a1-cann7.0.RC1.alpha002-oe2203sp2
docker.io	openeuler	cuda	cuda12.2.0-devel-cudnn8.9.5.30-oe2203sp2

Application Scenarios

- **Common openEuler operations**: Simplify the process of building a deep learning development environment while saving physical resources. For example, set up an Ascend development environment on openEuler.
- **openEuler development**: Developers familiarize themselves with the openEuler AI software stack to reduce the trial-and-error cost of installing components.

05 Kernel Innovations

openEuler 25.03 runs on Linux kernel 6.6 and inherits the competitive advantages of community versions and innovative features released in the openEuler community.

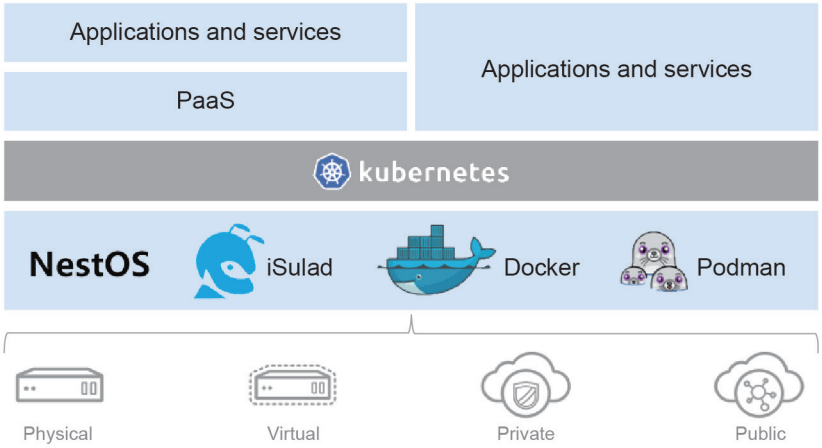
- **Kernel replication**: This feature optimizes Linux kernel performance bottlenecks in non-uniform memory access (NUMA) architectures. Research shows critical data center applications like Apache, MySQL, and Redis experience significant performance impacts from kernel operations. Kernel execution accounts for 61% of application CPU cycles, 57% of total instructions executed, 61% of I-cache misses, and 46% of I-TLB misses. Conventional Linux kernels restrict code segments, read-only data segments, and kernel page tables (**swapper_pg_dir**) to primary NUMA nodes without migration capability. This forces frequent cross-NUMA operations during system calls when processes or multi-threaded applications are deployed across multiple NUMA nodes, increasing memory access latency and degrading system performance. The kernel replication feature extends the **pgd** page global directory table in **mm_struct** by automatically creating NUMA-local replicas of kernel code segments, data segments, and page tables during kernel initialization. This mechanism maps identical kernel virtual addresses to physical addresses within their respective NUMA nodes, enhancing memory locality and reducing cross-NUMA overhead. Its implementation supports vmalloc, dynamic module loading, dynamic instruction injection mechanisms (Kprobe, KGDB, and BPF), security features (KPTI, KASLR, and KASAN), and 64 KB huge pages. A new boot-time cmdline configuration option (disabled by default) enables dynamic control for compatibility management. This feature benefits high-concurrency, multi-threaded server workloads.
- **HAOC 3.0 security feature**: Hardware-assisted OS compartmentalization (HAOC) leverages x86 and Arm processor capabilities to implement a dual-architecture kernel design. It creates isolated execution environments (IEE) within the kernel to prevent attackers from performing lateral movement and privilege escalation. The current version establishes IEE as a protected domain where sensitive resources can be incrementally isolated. These resources become accessible exclusively through controlled IEE interfaces, preventing unauthorized access by standard kernel code.

06 Cloud Base

NestOS

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

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.

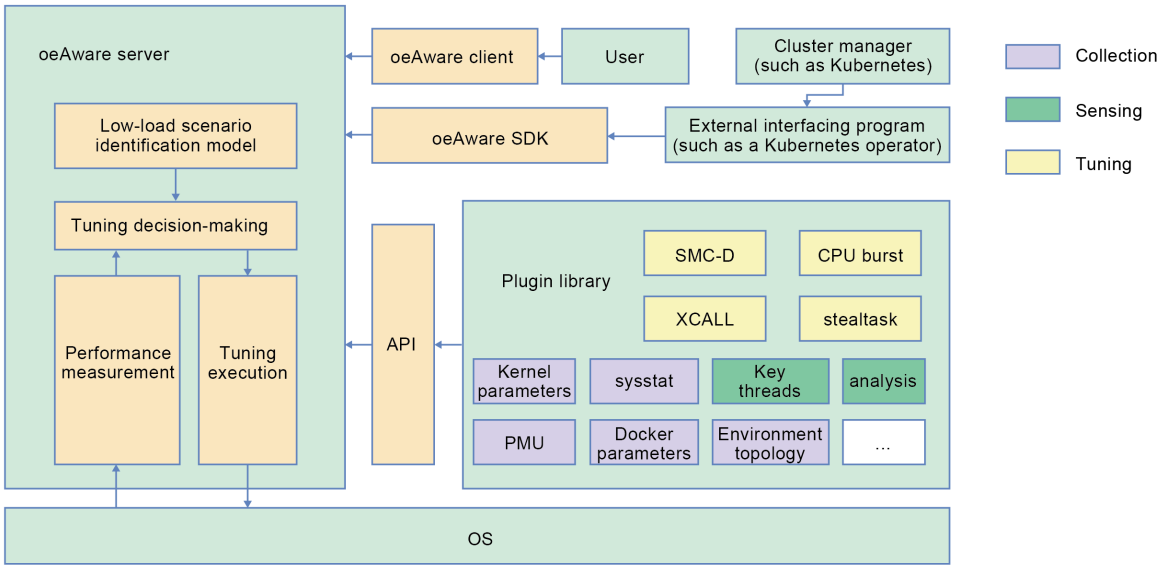
Application Scenarios

openEuler introduces the NestOS-Kubernetes-Deployer tool to resolve problems such as inconsistent and repeated O&M operations across stacks and platforms. These problems are typically caused by decoupling of containers from underlying environments when using container and container orchestration technologies for rollout and O&M. NestOS is developed for containerized cloud applications and ensures consistency between services and the base OS.

07 Enhanced Features

oeAware Enhancements

oeAware is a framework that provides low-load collection, sensing, and tuning upon detecting defined system behaviors on openEuler. The framework divides the tuning process into three layers: collection, sensing, and tuning. Each layer is associated through subscription and developed as plugins, overcoming the limitations of traditional tuning techniques that run independently and are statically enabled or disabled.



Feature Description

Every oeAware plugin is a dynamic library that utilizes oeAware interfaces. The plugins comprise multiple instances that each contains several topics and deliver collection or sensing results to other plugins or external applications for tuning and analysis purposes. openEuler 25.03 introduces the transparent_hugepage_tune and preload_tune plugins.

- The SDK enables subscription to plugin topics, with a callback function handling data from oeAware. This allows external applications to create tailored functionalities, such as cross-cluster information collection or local node analysis.
- The performance monitoring unit (PMU) information collection plugin gathers performance records from the system PMU.
- The Docker information collection plugin retrieves specific parameter details about Docker containers in the environment.
- The system information collection plugin captures kernel parameters, thread details, and resource information (CPUs, memory, I/Os, network) from the current environment.
- The thread sensing plugin monitors key information about threads.
- The evaluation plugin analyzes the overall system status during service operations and recommends optimal tuning methods.
- The system tuning plugins comprise stealtask for enhanced CPU tuning, smc_tune which leverages shared memory communication in the kernel space to boost network throughput and reduce latency, xcall_tune which bypasses non-essential code paths to minimize system call processing overhead, transparent_hugepage_tune which enables transparent huge pages to boost the TLB hit ratio, and preload_tune which seamlessly loads dynamic libraries.
- The Docker tuning plugin addresses CPU performance issues during sudden load spikes by utilizing the CPU burst feature.

Constraints

- **smc_tune**: SMC acceleration must be enabled before the server-client connection is established. This feature is most effective in scenarios with numerous persistent connections.
- Docker tuning is not compatible with Kubernetes containers.
- **xcall_tune**: The **FAST_SYSCALL** kernel configuration option must be activated.

Application Scenarios

stealtask is ideal for scenarios aiming to boost CPU utilization, such as in Doris. This tuning solution effectively increases CPU utilization and prevents idle CPU cycles.

xcall_tune is designed for applications with substantial system call overhead. It offers code paths that bypass non-critical processes, optimizing system call handling and reducing overhead. However, this approach may compromise some maintenance and security capabilities.

SMC-D excels in environments demanding high throughput and low latency, including HPC, big data processing, and cloud platforms. By leveraging DMA, SMC-D significantly reduces CPU load and accelerates interactive workloads.

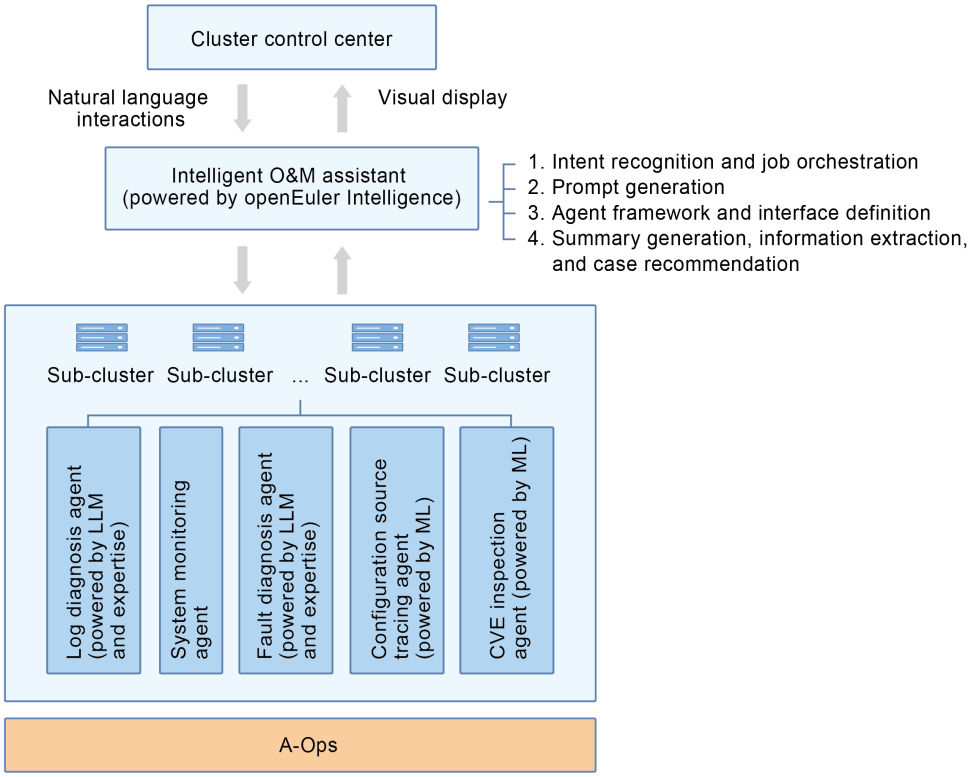
CPU burst is tailored for high-load container environments like Doris, addressing performance bottlenecks caused by CPU constraints.

transparent_hugepage_tune is applicable to scenarios with a large number of TLB misses.

A-Ops with CVE Fixing and Configuration Source Tracing

A-Ops empowers intelligent O&M through conversations and wizard-based operations. The intelligent conversations, featuring CVE prompts and fixes, configuration source tracing, configuration exception tracing, and configuration baseline synchronization, enable the O&M assistant to streamline routine O&M operations.

Feature Description



A-Ops integrates an intelligent O&M assistant powered by the openEuler Intelligence system to enable intelligent CVE fixing and configuration source tracing.

- **CVE fixing**: A-Ops displays cluster CVE status, prompts high-score and high-severity CVEs, and offers corresponding fixes. You can apply these fixes and check results using the assistant or WebUI.
- **Configuration source tracing**: You can use the assistant to find the machines with abnormal baseline configurations. The interface shows these machines and incorrect configuration items. It then intelligently gives you summaries and suggests fixes. You can correct the configurations using the assistant or WebUI.

Application Scenarios

A-Ops enables intelligent O&M by delivering CVE fixing and configuration source tracing through conversations, replacing traditional methods. It provides expert guidance for routine O&M and offers recommendations based on current operations, streamlining O&M and enhancing user experience.

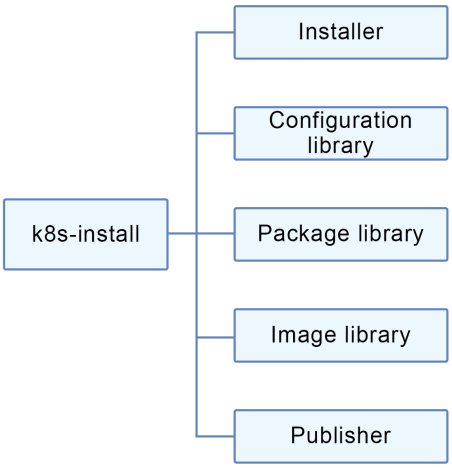
k8s-install

k8s-install is an online utility designed to provision cloud-native infrastructure on a wide range of Linux distributions and architectures. It also serves as a tool for creating offline installation packages. It supports installation, deployment, and secure updates of cloud-native infrastructure suites across multiple dimensions with just a few clicks, greatly reducing deployment and adaptation time while ensuring a standardized and traceable workflow.

Background and significance of the k8s-install project:

- openEuler suffered from outdated cloud-native toolchain versions and lacked maintenance for multiple version baselines (such as Kubernetes 1.20, 1.25, and 1.29) within the same release. Consequently, released branches could not be updated to major versions, requiring users to independently adapt and maintain later versions to meet business requirements.
- In actual service scenarios, tools such as Ansible are commonly employed for cloud infrastructure deployment, with non-standard packages, static binaries, and tarballs being frequently utilized rather than distribution-managed packages. This practice inherently lacks support for CVE fixes, thereby posing security risks.
- Version synchronization between offline and online installations is challenging. Furthermore, upgrading or modifying offline packages is difficult.
- The lack of standardized installation and deployment processes results in inconsistent component versions, leading to incompatibilities and configuration differences that make issue resolution time-consuming and root cause analysis difficult.

k8s-install Architecture



- The installer detects, installs, and updates the runC, containerd, Docker, and Kubernetes components and their dependent system libraries.
- The configuration library stores configuration file templates for Docker and Kubernetes software.
- The package library stores RPM packages for various versions and architectures of runC, containerd, Docker, Kubernetes, and their dependent system libraries.
- The image library stores images required for Kubernetes startup, such as various versions of kube-apiserver, kube-scheduler, etcd, and coredns. It also includes images for basic network plugins like Flannel.
- The publisher encapsulates the latest scripts, RPM packages, images, and configurations to create online and offline installation packages.
- Written as a Bash script, the main k8s-install program does not need to be compiled or linked. Its online installation package is encapsulated into an RPM package, built using spec files.

Module 1: k8s-install Installer

k8s-install is a tool used to install and securely update cloud-native infrastructure.

Version adaptation: openEuler suffered from outdated cloud-native toolchain versions from the upstream and released branches could not be updated to major versions, requiring users to independently adapt and maintain later versions to meet business requirements. k8s-install supports multiple baseline versions to meet service requirements, preventing deployment failures or function exceptions caused by version incompatibilities.

Improved deployment efficiency and standardization: The lack of standardized installation and deployment processes across departments or projects led to inconsistent component versions, resulting in frequent adaptation issues and time-consuming resolutions. k8s-install enables standardized deployment, ensuring component version compatibility, reducing fault locating time, and improving overall deployment efficiency.

Enhanced security and maintainability: In actual service scenarios, static binaries and tarballs are frequently deployed, with no support being provided for CVE fixes. k8s-install can fix CVEs in a timely manner, ensuring system security and stability. In addition, the code for all components has been committed to the openEuler repository, which facilitates version tracing and fault locating and enhances system maintainability.

Promoting open source and collaboration: By establishing and actively maintaining a repository within the openEuler community, the k8s-install project promotes technology sharing, fosters the growth of the community ecosystem, attracts more developers, enhances project influence, and promotes the continuous progress of cloud-native technologies.

Feature Description

The installer provides the following core functions:

Multi-version support: It supports multiple baseline Kubernetes versions, including 1.20, 1.25, and 1.29, to meet the version requirements of various business scenarios and enable on-demand deployment.

Multi-architecture support: With compatibility for various architectures including x86_64, AArch64, and LoongArch64, it is suitable for diverse hardware environments, thereby expanding its application scope.

Multi-component management: It integrates installation and configuration of Go, runC, containerd, Docker, Kubernetes, and related components, streamlining the deployment of complex components and improving efficiency.

Online and offline deployment: An online installation script **k8s-install** and an offline installation script **k8s-install-offline** are available. Combined with the **publish.sh** publisher, the installer ensures flexible and stable deployment across various network conditions.

Application Scenarios

Quick cluster deployment: k8s-install can install online or offline resource pools, create clusters, and add nodes to clusters.

CVE updates: k8s-install can apply CVE fixes to multiple resource pools to ensure system security and stability.

Troubleshooting: This tool helps troubleshoot issues related to Docker and Kubernetes, quickly pinpointing failure points and improving O&M efficiency.

Community engagement: The project is open-sourced on openEuler with an actively maintained repository, promoting technology sharing, fostering community ecosystem development, and enhancing project influence.

Module 2: k8s-install Publisher

publish.sh is the publisher in the k8s-install toolchain. It has the following advantages:

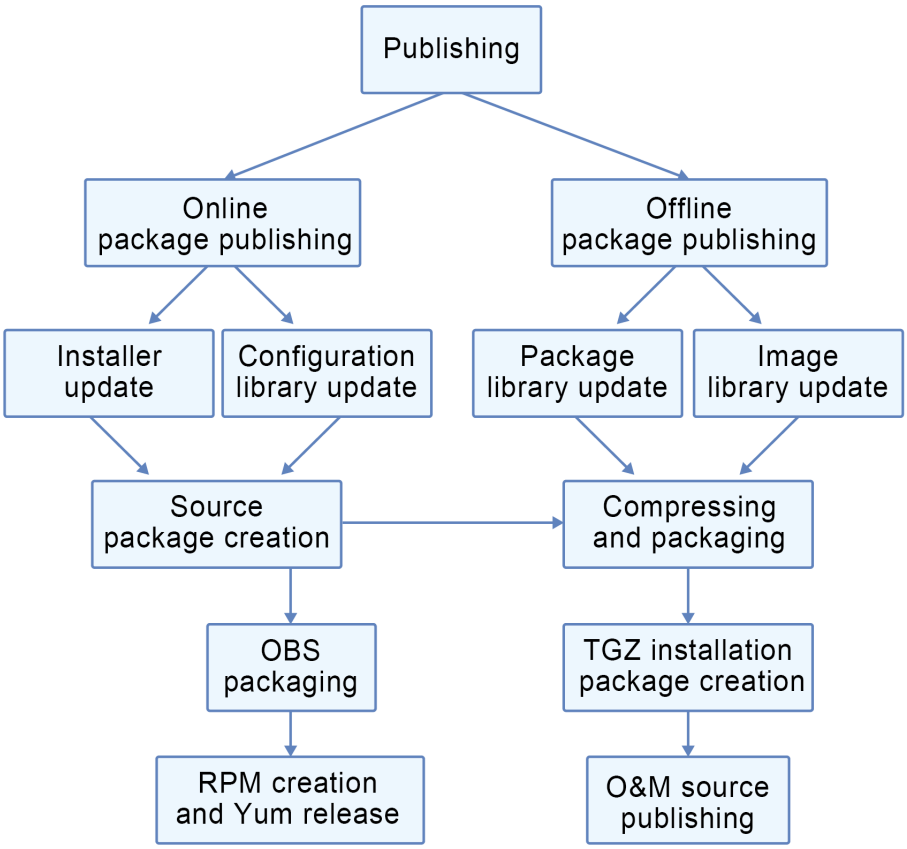
Offline deployment: In network-restricted or offline environments, such as certain data centers or specialized production setups, direct access to online repositories is not possible. **publish.sh** can generate complete offline installation packages, ensuring successful cloud-native infrastructure deployment in these scenarios and broadening the application scope of the tools.

Version iteration and release management: With the continuous updates of the k8s-install tool and its components, **publish.sh** enables automated build, test, and release processes. This enhances the efficiency of version iteration, ensures timely and accurate delivery of new versions to users, and facilitates the ongoing evolution of the system.

Stability and reliability of resource acquisition: Online repositories can face issues with package or image availability due to network fluctuations or delayed updates. **publish.sh** fetches resources from official or trusted online repositories and ensures their stability and reliability through integration and testing, preventing deployment failures caused by resource issues.

Multi-team collaboration and resource synchronization: In large projects, different teams may manage various components or modules. **publish.sh** can integrate and publish the updates from each team, ensuring resource consistency across teams. It facilitates collaboration and improves overall project progress and quality.

Feature Description



The publisher provides the following functions:

Offline package generation and release: It pulls the latest software packages and images from online Yum and image repositories, combines them with the latest configuration files and installer, and packages them into an offline **.tgz** installation package to meet the deployment needs of offline environments.

Online code update and release: It uploads the updated code to the Git repository, selects the configuration library and installer for source code packaging, uploads it to the OBS server for official compilation after local build testing, and publishes it to the Yum repository to achieve online resource update and synchronization.

Application Scenarios

Version iteration and release management: When updates are available for code, RPM packages, or component images on k8s-install, **publish.sh** facilitates efficient and reliable release management. This ensures the timely and accurate retrieval of updated software packages and images, the generation of offline installation packages, package uploads, packaging and testing, and source code pushes for new versions. This process supports the efficient and sustainable iterative evolution of system development and release.

GCC for openEuler

The baseline version of GCC for openEuler has been upgraded from open source GCC 10.3 to GCC 12.3, supporting 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-A65AE Cortex-A34	Armv8.6-a, +bf16, +i8mm Armv8.6-r Cortex-A78 Cortex-A78AE Cortex-A78C Cortex-X1	Armv8.7-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 12.3 of the upstream community.
- Further, it fuels AutoFDO to improve the performance of MySQL databases at the application layer.

Feature Description

- **SVE**: Significantly improves program running performance for Arm-based machines that support SVE instructions.
- **Memory layout**: Rearranges the structure members so that frequently accessed members are placed in continuous memory locations, boosting the cache hit ratio and enhancing program performance.
- **SLP transpose optimization**: Improves the analysis of loops with consecutive memory reads during loop splitting, and adds analysis to transpose grouped stores in the superword level parallelism (SLP) vectorization stage.
- **Redundant member elimination**: Eliminates structure members that are never read and deletes redundant write statements, which in turn reduces the memory footprint of the structure and alleviates subsequent bandwidth pressure, while improving performance.

- **Static compression optimization of structure members**: Reduces the overall memory footprint of the structure to improve the cache hit ratio.
- **Array comparison**: Implements parallel comparison of array elements to improve execution efficiency.
- **Arm instruction optimization**: Simplifies the pipeline of ccmp instructions for a wider range of deployments.
- **IF statement optimization**: Splits and optimizes the IF statement blocks to improve constant propagation within a program.
- **SLP vectorization**: Enhances SLP to cover more vectorization scenarios and improve performance.
- **AutoFDO**: Uses perf to collect and parse program information and implements FDO across the compilation and binary phases, boosting mainstream applications such as MySQL databases.

Application Scenarios

In general-purpose computing, GCC for openEuler showed a 20% performance gain over GCC 10.3 in the SPEC CPU 2017 benchmark.

In other scenarios, MySQL performance increases by 15% with AutoFDO enabled, while UnixBench performance is boosted by over 3% with kernel-mode PGO.

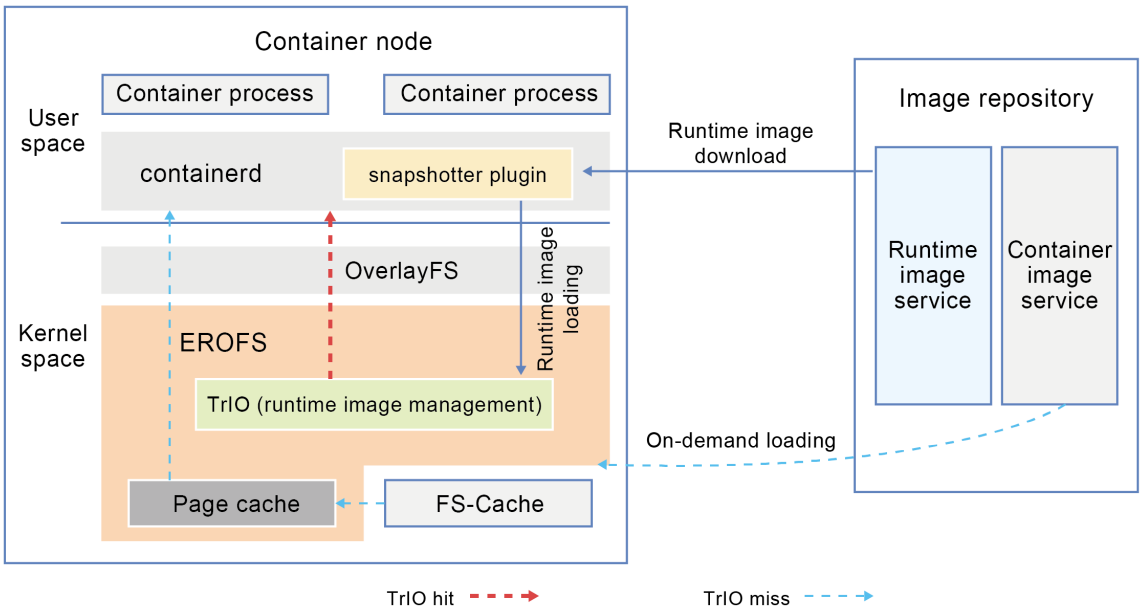
Trace IO

Trace IO (TrIO) is designed to optimize the on-demand loading of container images using EROFS over fscache. It achieves this by accurately tracking I/Os during container startup and efficiently orchestrating I/Os into container images to improve the cold startup process of containers. Compared with existing container image loading solutions, TrIO can significantly reduce the cold startup latency of container jobs and improve bandwidth utilization. TrIO comprises both kernel-space and user-space modules. The kernel-space module includes adaptations within the EROFS file system. The user-space module provides tools for capturing I/O traces during container runtime and offers an adaptation modification guidance based on Nydus snapshotter. This allows container users to leverage TrIO without modifying containerd and runC, ensuring compatibility with existing container management tools.

The core advantage of TrIO lies in its ability to aggregate I/O operations during on-demand container loading. By orchestrating the runtime I/O traces of container jobs, TrIO accurately fetches the necessary I/O data during container execution. This greatly improves the efficiency of pulling image data during container startup, thereby achieving low latency.

Feature Description

TrIO's functionality comprises two main aspects: capturing container runtime I/Os and utilizing the runtime I/Os during container startup. Container runtime I/Os are captured by using eBPF to trace I/O operations in the file system. This allows for obtaining the I/O read requests during container job startup, and orchestrating the corresponding data to build a minimal runtime image. During container startup, a custom snapshotter plugin module pulls the minimal runtime image using large I/O operations and imports it into the kernel. Subsequently, all I/O operations during container job execution will preferentially read from this minimal runtime image. The following figure shows the process of starting a container using TrIO.



Compared with the existing on-demand container loading solutions, TrIO has the following advantages:

- **No I/O amplification**: TrIO accurately captures runtime I/Os and use them for job startup. It ensures that I/Os are not amplified during container job startup.
- **I/O aggregation**: During container job startup, TrIO uses large I/O operations to pull all the necessary data for the startup process to the container node at once. This improves the efficiency of loading image data while reducing startup latency.

Application Scenarios

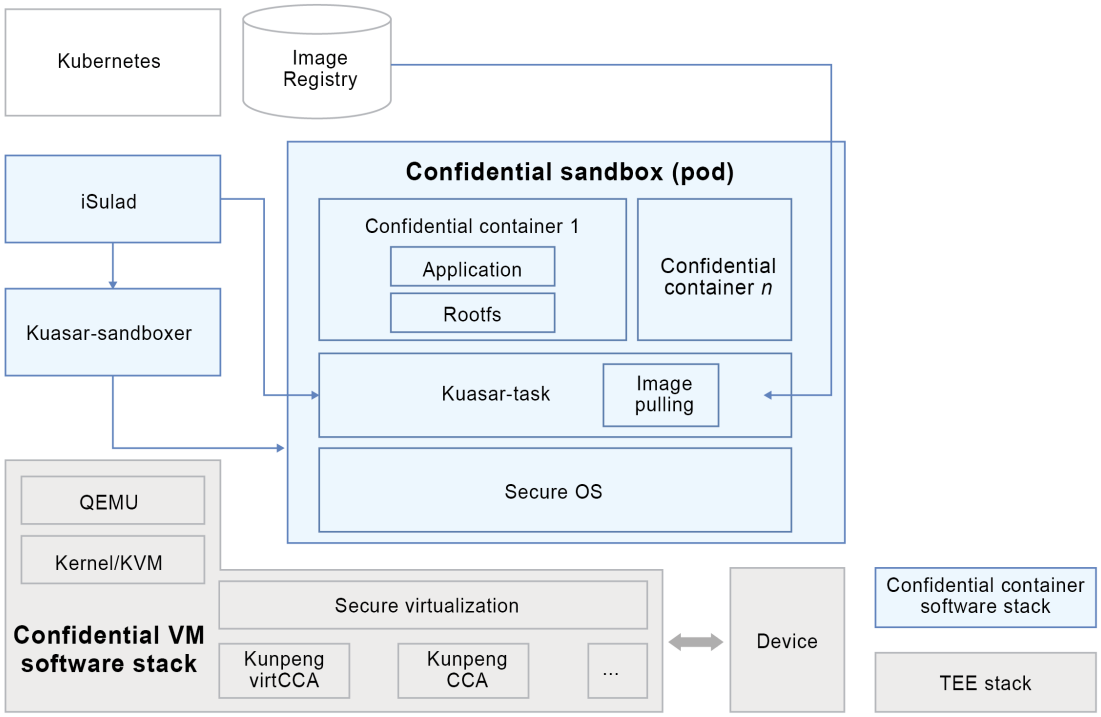
TrIO's core advantage is I/O aggregation, efficiently consolidating discrete network I/Os during the on-demand loading of container images. This makes it particularly well-suited for scenarios with a high volume of discrete I/Os during container service startup, leading to further performance gains in on-demand container image loading.

Kuasar Integration with virtCCA

The Kuasar confidential container leverages the virtCCA capability of Kunpeng 920 processors. It connects northbound to the iSulad container engine and southbound to Kunpeng virtCCA hardware, enabling seamless integration of Kunpeng confidential computing with the cloud-native technology stack.

Kuasar fully utilizes the advantages of the Sandboxer architecture to deliver a high-performance, low-overhead confidential container runtime. Kuasar-sandboxer integrates the virtCCA capability of openEuler QEMU to manage the lifecycle of confidential sandboxes, allowing you to create confidential sandboxes on confidential hardware and ensuring containers run within a trusted execution environment (TEE).

Kuasar-task offers a Task API for iSulad to manage lifecycles of containers within secure sandboxes. Container images are securely pulled into encrypted sandbox memory through Kuasar-task's image pulling capability.



Technical Constraints

- Remote attestation support of Kuasar has been integrated via secGear in the SP versions of openEuler 24.03 LTS.
- Image encryption/decryption capabilities will be added after secGear integration.

Feature Description

Kuasar has expanded its capabilities to include confidential container support while maintaining existing secure container functionality. You can enable this feature through iSulad runtime configuration.

Supported features:

- Native integration with the iSulad container engine preserves Kubernetes ecosystem compatibility.
- Hardware-level protection via Kunpeng virtCCA technology ensures confidential workloads are deployed in TEEs.

Application Scenarios

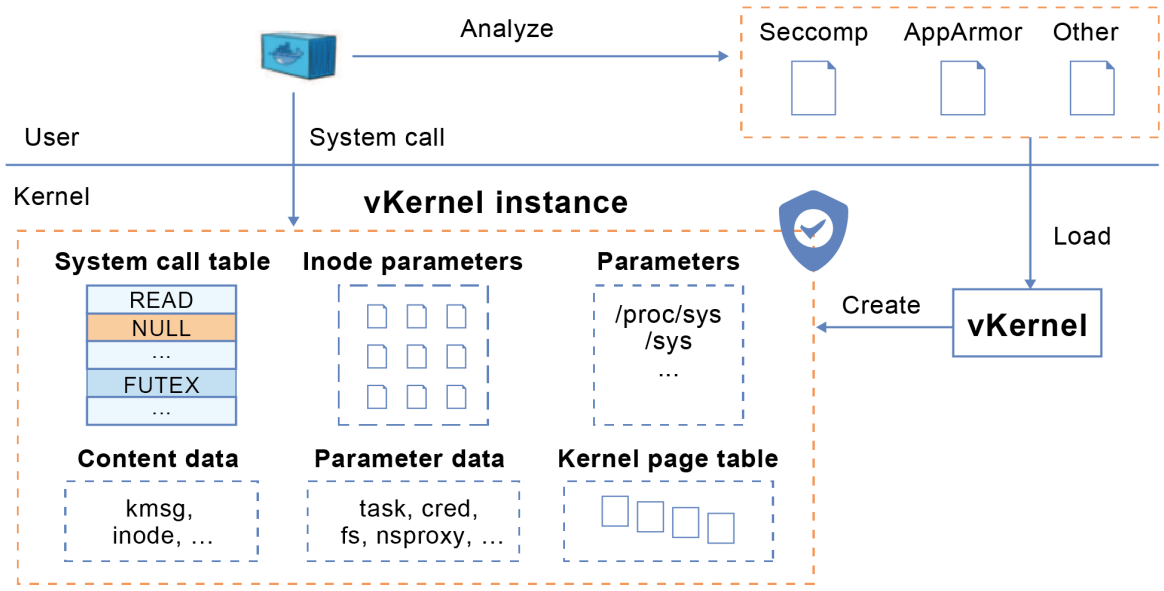
Kuasar addresses data security needs while seamlessly integrating with cloud-native ecosystems. It empowers confidential applications with cloud-native benefits including high availability, elastic scaling, and rapid deployment. It finds broad application in confidential computing scenarios spanning AI security, trusted data circulation, and privacy protection.

vKernel for Advanced Container Isolation

The virtual kernel (vKernel) architecture represents a breakthrough in container isolation, addressing the inherent limitations of shared-kernel architectures while preserving container performance efficiency.

Feature Description

vKernel creates independent system call tables and file permissions tables to enhance foundational security. It implements isolated kernel parameters, enabling containers to customize both macro-level resource policies and micro-level resource configurations. By partitioning kernel data ownership, leveraging hardware features to protect kernel privilege data, and building isolated kernel page tables for user data protection, vKernel further reinforces security. Future iterations will explore kernel data related to performance interference to strengthen container performance isolation capabilities.



Application Scenarios

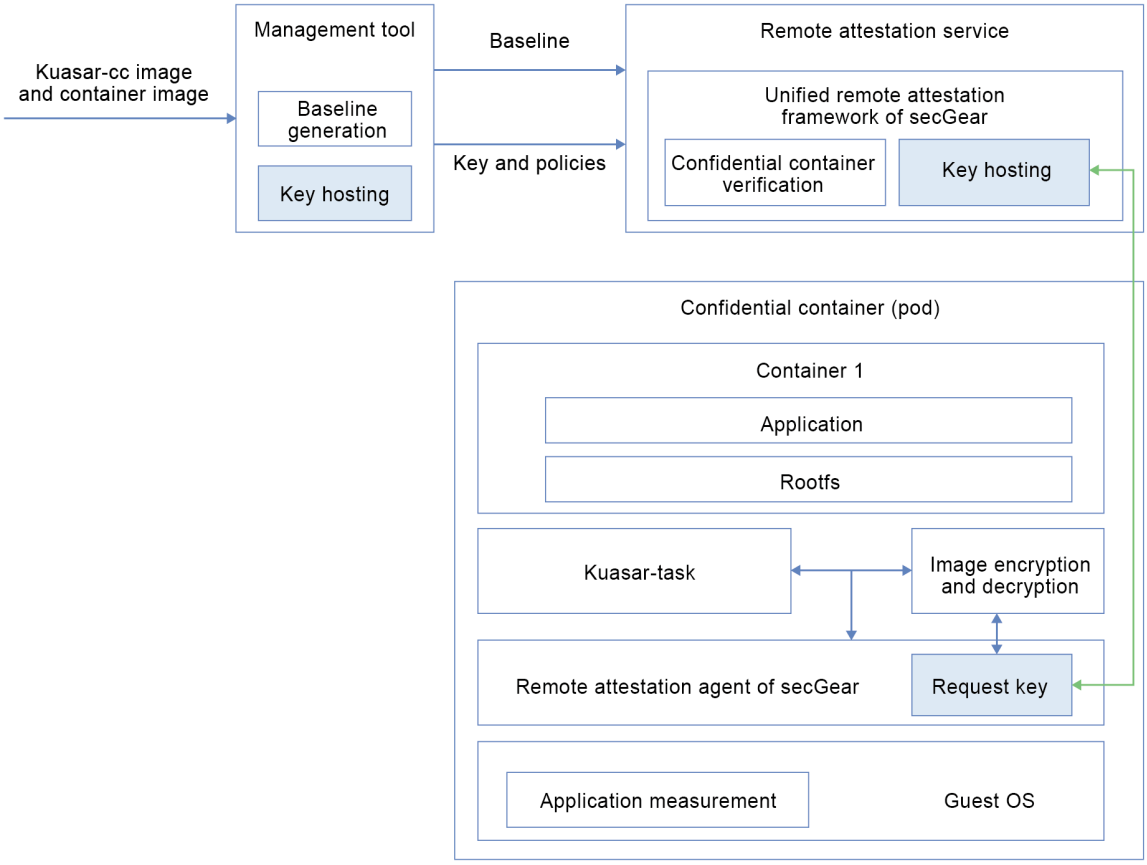
vKernel strengthens security isolation and customization for container deployments:

Supported features:

- **Enhanced security**: Implements flexible system call filtering (including parameter dereferencing), rapid inode-based file retrieval, and customizable interception policies.
- **Strict data isolation**: Restricts container access to its own kernel logs, eliminating sensitive data leakage.
- **Granular resource control**: Prevents resource exhaustion (such as file descriptors) by malicious containers, enables workload-specific kernel configurations even for conflicting requirements, and optimizes performance through tailored parameter tuning (such as transparent huge pages).

secGear with Secure Key Hosting for Confidential Container Images

The remote attestation service of secGear provides secure key hosting capabilities for confidential container images, establishing a management system that encompasses secure key storage, dynamic fine-grained authorization, and cross-environment collaborative distribution. By integrating zero-trust policies and automated auditing capabilities, secGear ensures data confidentiality and operational traceability while optimizing the balance between key governance and operational costs. This delivers a unified "encrypt by default, decrypt on demand" security framework for cloud-native environments.



Feature Description

secGear combines remote attestation technologies to build a layered key hosting architecture.

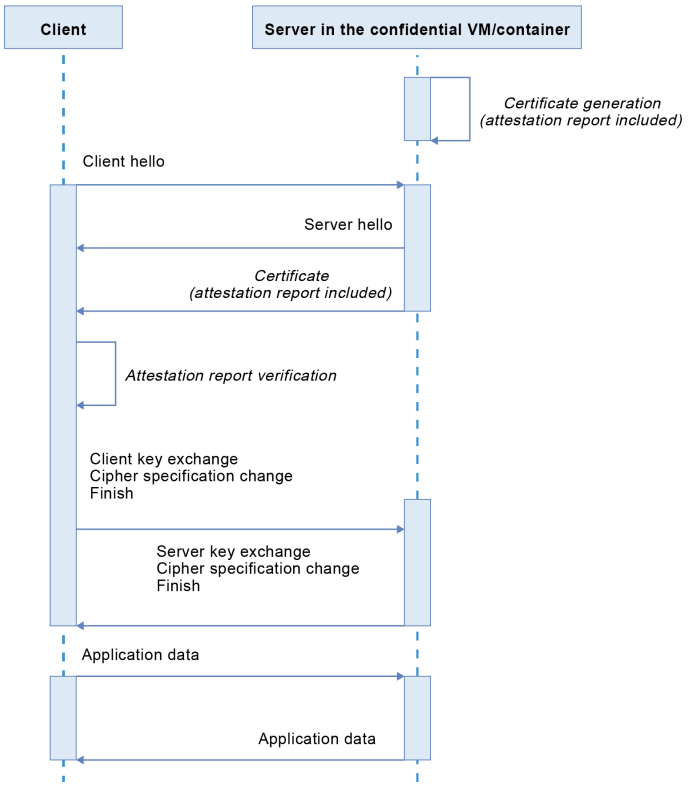
- **Attestation service**: A centralized key hosting server leverages the remote attestation mechanism of TEEs to securely store and manage image encryption keys throughout their lifecycle. It offers authorized users granular policy configuration interfaces for tailored access control.
- **Attestation agent**: Lightweight attestation agent components deployed within confidential compute nodes expose local RESTful APIs. The confidential container runtime invokes these APIs to validate the integrity of the TEE and establish secure dynamic sessions with the server, enabling secure key transmission.

Application Scenarios

The key hosting feature of secGear, combined with Kuasar, creates a comprehensive confidential container solution designed for applications demanding stringent data security and privacy protection. This solution proves particularly valuable across industries including finance, healthcare, public services, cloud computing, and Internet of Things (IoT). By implementing container image encryption and key hosting, the system safeguards image integrity, mitigates data breach risks, thwarts supply chain attacks, and ultimately strengthens user confidence in cloud-native service providers.

RA-TLS

RA-TLS integrates remote attestation of confidential computing into TLS negotiation procedures, ensuring secure transmission of sensitive data into TEEs while simplifying secure channel establishment for confidential computing workloads, thereby lowering adoption barriers. This figure shows the process.



Feature Description

One-way authentication: In deployments where TLS servers operate within confidential environments and clients reside in regular environments, RA-TLS validates the legitimacy of the server confidential environment and applications through remote attestation before TLS key negotiation.

Two-way authentication: For scenarios where both TLS servers and clients operate within confidential environments, RA-TLS enforces mutual validation of peer environments and applications via remote attestation before TLS key negotiation.

Technical constraints: Confidential computing environments must maintain network accessibility (such as virtCCA-enabled configurations).

Application Scenarios

RA-TLS is highly versatile for confidential computing applications, especially in cloud computing, big data, finance, and healthcare. It enables users to validate the legitimacy of cloud-based confidential computing environments and applications, set up encrypted communication channels, and safely process sensitive data within a protected environment, effectively preventing unauthorized access to private data.

openAMDC for High-Performance In-Memory Data Caching and KV Storage

openAMDC stands for open advanced in-memory data cache. It caches data in memory to accelerate access, enhance application concurrency, and minimize latency, and can serve as both a message broker and in-memory database.

Feature Description

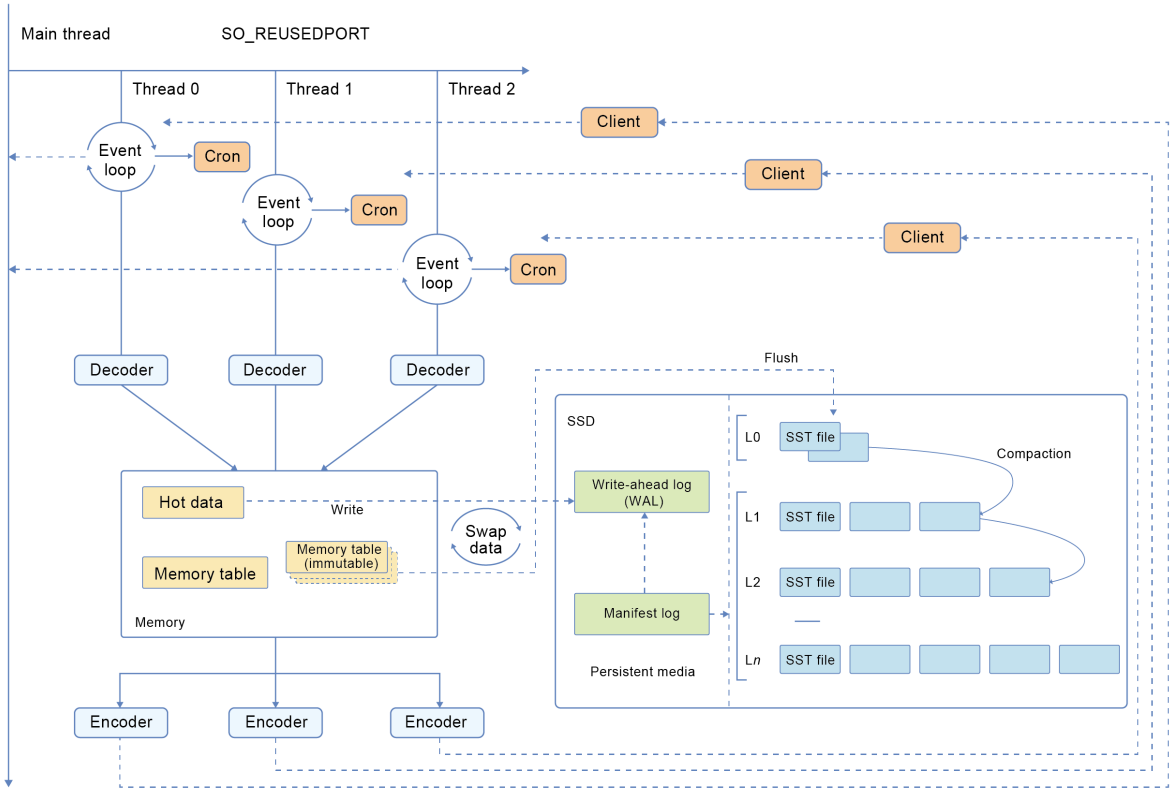
- Core capabilities

  openAMDC, compatible with the Redis Serialization Protocol (RESP), delivers comprehensive caching for strings, lists, hashes, and sets while supporting active-standby, cluster, and sentry deployment options. The following table lists the functions.

Function	Description
Multi-data type caching	Supports caching for string, list, hash, set, sorted set, GEO, HyperLogLog, and stream data types.
Publishing and subscription	Implements publishing and subscription to enhance system capabilities.
Access control lists	Provides secure service access mechanisms with fine-grained permission controls.
IP address allowlists	Supports allowlists for individual IP addresses and subnets to improve security.
Memory eviction policies	Offers multiple data eviction policies to meet various requirements and improve memory utilization.
Persistence	Enhances cache core reliability and prevents data loss during outages.
Lua script support	Allows operations on the cache core using Lua scripts.
SSL support	Supports encrypted communication via SSL.
Active-standby mode	Supports active-standby deployment.
Sentry mode	Provides node monitoring, automatic failover, fault notification, and configuration propagation for active-standby setups.
Cluster mode	Supports elastic scaling (memory expansion/reduction) with built-in failover capabilities.

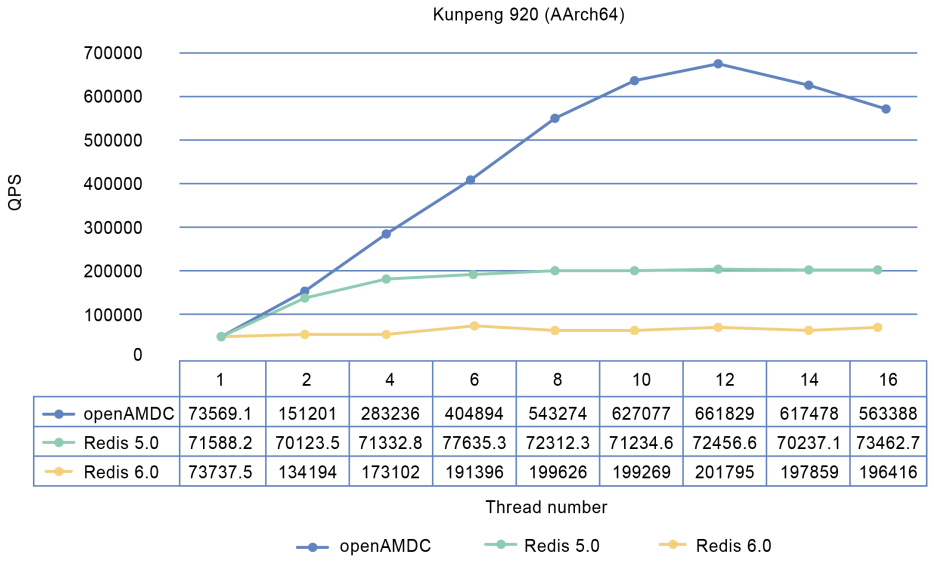
- Architecture features

  openAMDC, compatible with the Redis Serialization Protocol (RESP), delivers comprehensive caching for strings, lists, hashes, and sets while supporting active-standby, cluster, and sentry deployment options. The following table lists the functions.



openAMDC architecture

Multi-threaded architecture: During initialization, openAMDC spawns multiple worker threads, each running an event loop for network monitoring. By enabling SO_REUSEPORT for socket listeners, kernel-level load balancing is implemented across threads sharing the same port. This approach eliminates resource contention from shared listening sockets through dedicated per-thread socket queues, substantially improving concurrency.



	1	2	4	6	8	10	12	14	16
openAMDC	73569.1	151201	283236	404894	543274	627077	661829	617478	563388
Redis 5.0	71588.2	70123.5	71332.8	77635.3	72312.3	71234.6	72456.6	70237.1	73462.7
Redis 6.0	73737.5	134194	173102	191396	199626	199269	201795	197859	196416

Data exchange architecture: Built upon the multi-threaded foundation, openAMDC implements data exchange capabilities supporting hybrid memory-drive storage, effectively optimizing total cost of ownership while maintaining performance efficiency.

Application Scenarios

openAMDC accelerates data access through in-memory processing, supporting the following scenarios.

Scenario	Description
Hot data caching	Delivers high-performance data caching capabilities that securely support mission-critical applications with large-scale, high-concurrency, and high-availability requirements, ensuring optimal system performance.
Distribution locks	Provides lock mechanisms for distributed systems to prevent data corruption issues caused by concurrent operations from multiple nodes.
Data sharing	Enables seamless sharing of cached generic data across distributed systems and among multiple applications and services.
Lightweight message queue	Efficiently implements lightweight message queues through publishing/subscription, list data, and stream data.
Bit-wise big data operations	Specialized for massive data volume scenarios requiring statistics and deduplication, such as online user statistics and retention analysis.
Time-limited operations	Incorporates a time to live (TTL) mechanism that enables time-bound services which automatically expire if not completed within designated time parameters.

OpenStack 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. It provides the following main functions:

- **Compute service**: creates, deploys, and destroys 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 services. 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, providing capabilities for topology planning, subnet management, and security group configuration. These features enable the 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 deployment and application management and facilitates collaboration and integration between services. Orchestration services like Heat help streamline application deployment and management by automatically perform related tasks based on user-defined templates.

Application Scenarios

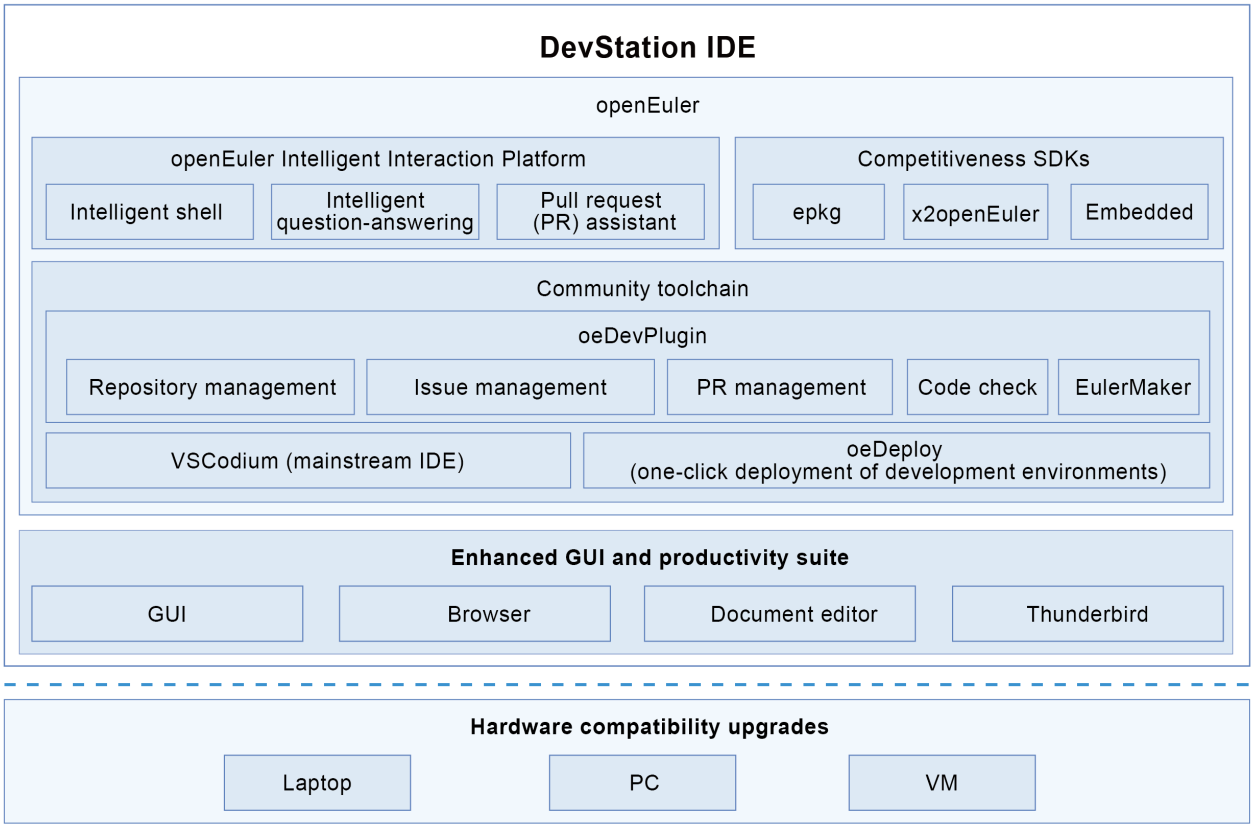
- **Private cloud**: Enterprises can deploy OpenStack in their own data centers or private cloud environments based on their requirements and IT resource status. In this way, OpenStack provides centralized resource management, automated deployment, auto scaling, and robust security measures such as access control, data encryption, and log audit.
- **Public cloud**: OpenStack implements resource pools for public clouds to meet 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.

openEuler DevStation

openEuler DevStation is a Linux desktop OS built for developers, streamlining workflows while ensuring ecosystem compatibility. The latest release delivers major upgrades across three dimensions: supercharged toolchain, smarter GUI, and extended hardware support. These improvements create a more powerful, secure, and versatile development platform.

Feature Description

- Developer-centric community toolchain
  - » **Comprehensive development suite**: Pre-configured with VSCodium (an open source, telemetry-free IDE) and development environments for major languages including Python, Java, Go, Rust, and C/C++.
  - » **Enhanced tool ecosystem**: Features innovative tools like oeDeploy for seamless deployment, epkg for extended package management, DevKit utilities, and an AI-powered coding assistant, delivering complete workflow support from environment configuration to production-ready code.
  - » **oeDevPlugin Extension**: A specialized VSCodium plugin for openEuler developers, providing, visual issue/PR dashboards, quick repository cloning and PR creation, automated code quality checks (such as license headers, formatting), and real-time community task tracking.
  - » **Intelligent assistant**: Generates code from natural language prompts, creates API documentation with few clicks, and explains Linux commands, with a privacy-focused offline operation mode.
- Enhanced GUI and productivity suite
  - » **Smart navigation and workspace**: Features an adaptive navigation bar that intelligently organizes shortcuts for development tools, system utilities, and common applications—all with customizable workspace layouts.
  - » **Built-in productivity applications**: Pre-installs the Thunderbird email client for seamless office workflows.
- Hardware compatibility upgrades
  - » **Laptop-ready support**: Comprehensive compatibility with modern laptop components, including precision touchpads, Wi-Fi 6/Bluetooth stacks, and multi-architecture drivers, delivering faster AI and rendering workloads.
  - » **Raspberry Pi DevStation image**: Provides an Arm-optimized development environment out of the box, featuring a lightweight desktop environment with pre-installed IoT development tools (VSCodium and oeDevPlugin) and accelerated performance for Python scientific computing libraries like NumPy and pandas.



Application Scenarios

Open source collaboration for community developers: New openEuler contributors can seamlessly clone community repositories within VSCodium using oeDevPlugin, while guided PR templates help them complete test cases efficiently.

Streamlined maintainer workflows: Maintainers leverage the integrated conflict detection engine of the plugin to identify potential merge conflicts across multiple PRs. The openEuler Intelligence system automatically generates code change summaries, enabling rapid quality assessment and significantly reducing manual review efforts.

Global team coordination: Distributed teams maintain dependency consistency by sharing custom package repositories through epkg, while oeDeploy enables one-click deployment of development environments across multiple distributed nodes.

Integrated CI/CD process: Developers can directly submit locally tested builds to the openEuler community pipeline after completing architecture-specific debugging, triggering comprehensive automated testing workflows.

oeDeploy for Simplified Software Deployment

oeDeploy revolutionizes software deployment as a lightweight yet powerful tool that accelerates environment setup across single-node and distributed systems with unmatched efficiency.

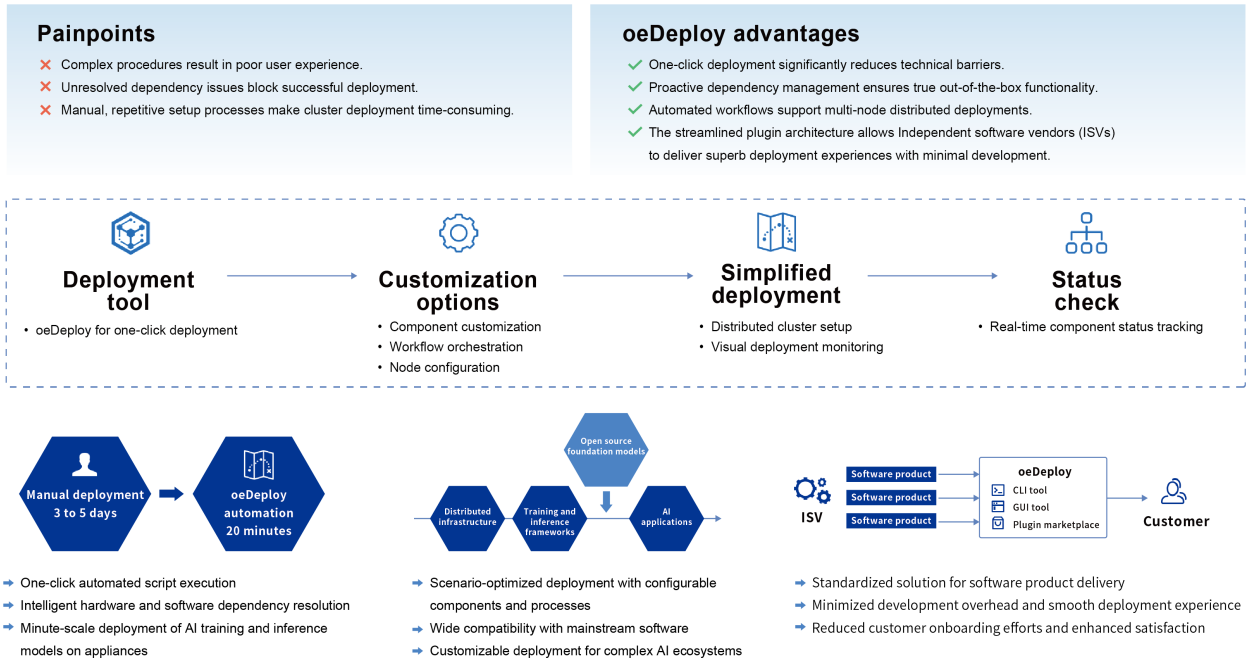
Feature Description

Universal deployment: Seamlessly handles both standalone and cluster deployments through automation, eliminating manual efforts and slashing setup times.

Pre-built software solutions: Comes with optimized deployment solutions for industry-standard software, with continuous expansion through a growing plugin ecosystem.

Customizable architecture: Features an open plugin framework that empowers developers to build tailored deployment solutions aligned with their unique technical requirements.

Developer-centric design: Combines robust CLI capabilities with upcoming GUI tools and a plugin marketplace, letting developers concentrate on innovation rather than infrastructure.



Application Scenarios

ISVs and development teams can adopt oeDeploy as a standardized solution for software product delivery. Its CLI tools and plugin framework minimize development overhead while ensuring smooth delivery, reducing customer onboarding efforts and enhancing satisfaction. Today, oeDeploy supports rapid deployment for applications including EulerMaker, DeepSeek on MindSpore, and various AI agents.

oeDeploy enables instant setup of complex environments, deploying AI frameworks like Kuberay, TensorFlow, and PyTorch in minutes. This eliminates tedious configuration work and accelerates the learning curve. Developers can also extend oeDeploy by creating and sharing custom deployment templates, democratizing one-click deployment for broader user communities.

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/docs/en/vulnerability-management-process/security-process-en.md
Security disclosure	https://gitee.com/openeuler/security-committee/blob/master/docs/en/vulnerability-management-process/security-disclosure-en.md