openEuler OS Technical Whitepaper

Innovation Projects

November, 2025

OpenAtom openEuler Community

Contents

1 Introduction

OpenAtom openEuler (openEuler for short) is an all-scenario open source operating system (OS) project incubated by the OpenAtom Foundation. Designed for four core digital infrastructure scenarios — servers, cloud computing, edge computing, and embedded systems — it provides comprehensive support for diverse computing architectures including Arm, x86, RISC-V, LoongArch, PowerPC, and SW-64.

Since its contribution to the OpenAtom Foundation, openEuler has achieved significant progress in industry adoption, ecosystem development, and global collaboration:

- Industrial impact: Over 21 vendors have released commercial distributions based on openEuler, with deployments exceeding 10 million across finance, telecommunications, energy, government, and internet sectors.
- Ecosystem growth: Embracing principles of co-construction, co-governance, and sharing, the community now unites 2,000+ member organizations, with major contributors including Intel and Arm.
- Global footprint: Through technical collaboration with Linux Foundation and others, 40+ international projects natively support openEuler.
- Supply chain security: In 2024, openEuler became the first open source community to obtain ISO 18974 certification for software supply chain security, demonstrating leadership in this critical domain.

Looking ahead, openEuler will drive AI innovation by building a full-stack open source AI framework, democratizing model inference technologies, and serving as the next-generation intelligent digital foundation.

Milestones

- Sep 2019: Huawei unveiled openEuler, an open source version derived from its proprietary EulerOS, transitioning it from internal use to community-led growth.
- Dec 2019: openEuler community was established, fostering collaborative innovation.
- Nov 2021: The openEuler project was contributed to OpenAtom Foundation, shifting from corporate-driven leadership to a collaborative, industry-wide governance model.
- 2022: Won the World Internet Leading Technology Achievement Award for innovations in heterogeneous computing and cloud-native orchestration.
- Dec 2022: Designated as an OpenAtom Project Group, formalizing multi-stakeholder contribution framework.
- Dec 2024: Captured a dominant 50.2% market share in China's new server OS landscape, spearheading adoption across telecommunications, public services, finance, power, and energy.

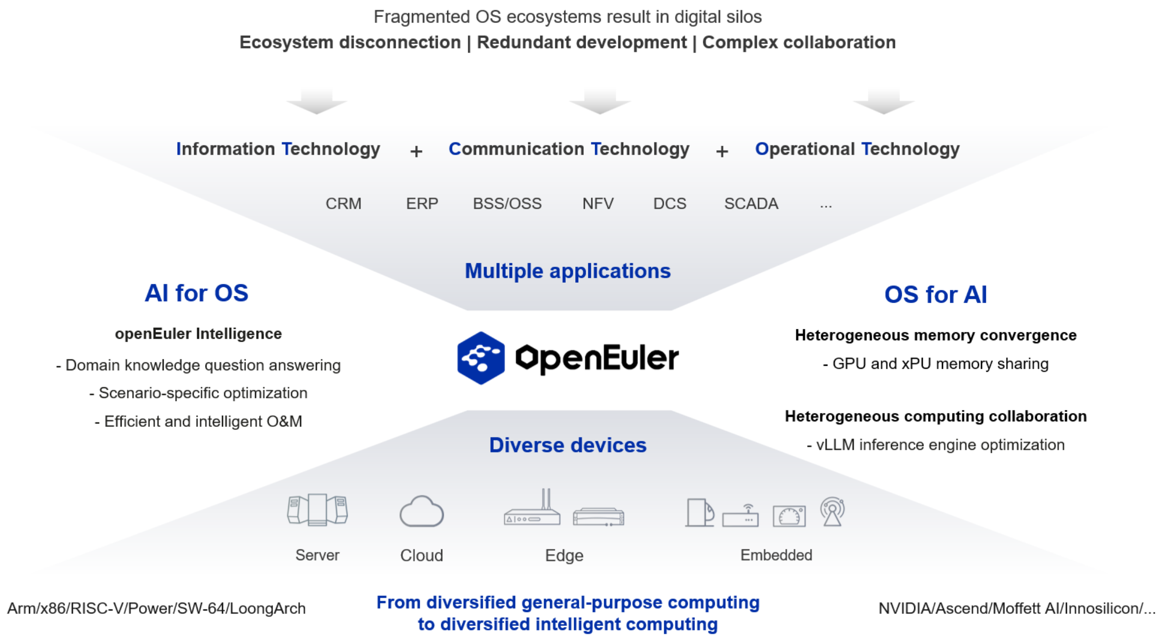
2 Technology Ecosystem

openEuler: An Innovative Platform Spanning All Scenarios

openEuler is an open source OS designed for digital infrastructure. It utilizes a unified OS architecture to support diverse devices at the hardware layer and encompass a comprehensive range of applications. It also interoperates with other OSs, such as OpenHarmony, by sharing capabilities to foster a broader ecosystem. openEuler represents a significant advancement, uniquely providing 100% support for all mainstream computing architectures under a single OS architecture. This establishes it as the premier open source OS for diverse computing power. openEuler pioneers the all-scenario OS concept through full-stack atomization and a highly modular architecture. This innovative design enables flexible version creation and free service combination, allowing a single architecture to effectively support server, cloud computing, edge computing, and embedded systems.

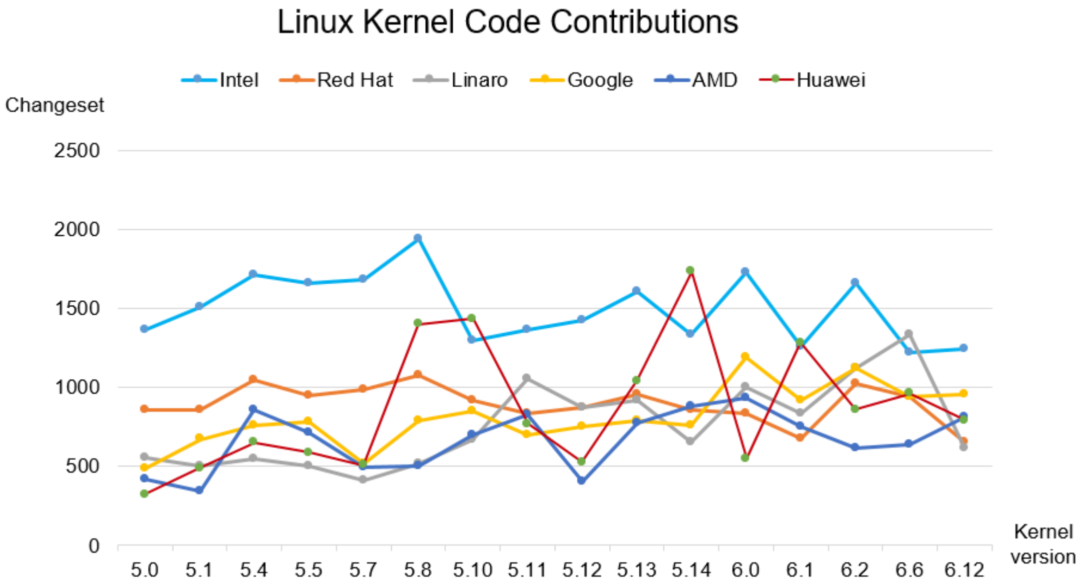
The openEuler OS is defined by the following characteristics:

- All-scenario coverage for digital infrastructure: It supports server, cloud, edge, and embedded deployments, aiming to provide a secure, stable, and easy-to-use OS. By offering deterministic assurance for applications, it supports operational technology (OT) domain applications and the essential convergence of OT and information and communications technology (ICT).
- Optimal support for diverse computing: openEuler supports a wide array of mainstream chips (including Intel, AMD, Kunpeng, Phytium, Zhaoxin, Loongson, Hygon, and Sunway). Crucially, it supports multiple computing architectures, such as Arm, x86, RISC-V, DPU, IPU, and NPU. This versatility intelligently matches various workloads to optimal computing resources, maximizing the performance of diverse computing power.
- Kernel innovation and leadership: openEuler actively drives kernel innovation. The 24.03 LTS version supports Linux Kernel 6.6, a milestone that ranks it first in China and among the global Top 3 in terms of contributions. The platform continues to advance technology by contributing its innovations—especially in diverse computing power and memory-centric architecture—back to the upstream community.


Fragmented OS ecosystems result in digital silos
Ecosystem disconnection | Redundant development | Complex collaboration
Information Technology
+
Communication Technology
+
Operational Technology
CRM
ERP
BSS/OSS
NFV
DCS
SCADA
...
Multiple applications
AI for OS
openEuler Intelligence
- Domain knowledge question answering
- Scenario-specific optimization
- Efficient and intelligent O&M
OpenEuler
OS for AI
Heterogeneous memory convergence
- GPU and xPU memory sharing
Heterogeneous computing collaboration
- vLLM inference engine optimization
Diverse devices
Server
Cloud
Edge
Embedded
Arm/x86/RISC-V/Power/SW-64/LoongArch
From diversified general-purpose computing to diversified intelligent computing
NVIDIA/Ascend/Moffett AI/Innosilicon/...

openEuler's Sustained Contributions to the Linux Kernel

The openEuler community—powered by industry leaders like Huawei, Loongson Technology, Kylinsoft, UnionTech, Phytium, Chengdu Jingrong Lianchuang, China Telecom, and China Mobile—consistently delivers impactful contributions to the upstream Linux Kernel community. Their efforts span critical domains such as chip architectures, ACPI optimization, memory management, file system enhancements, media integration, kernel documentation updates, and robust quality improvements through bug fixes and meticulous code refactoring.

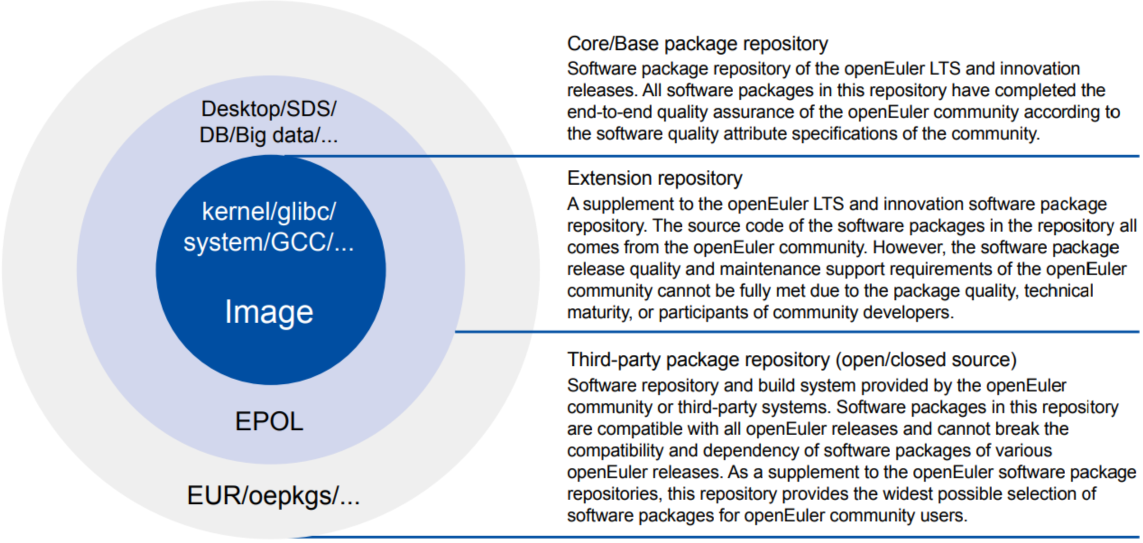


- **The community's significant impact is demonstrated by the following metrics: Huawei has consistently ranked among the Top 5 historical contributors to the Linux Kernel, achieving the #1 contributor ranking for versions 5.10, 5.14, and 6.1, and ranking Top 3 globally for version 6.6.**
- **The community is home to China's first GCC maintainer. Community contributions include over 13,000 Linux Kernel patches and a top 5 contributor ranking in the Docker community.**

Beyond the kernel, the openEuler community demonstrates robust growth and influence, boasting over 2,089 community partners, more than 22,000 contributors, a cumulative total of over 2.258 million PRs merged, and a global download count surpassing 4.24 million.

Software Package Repositories

The openEuler community works with third-party developers to provide a vast array of user-friendly software packages. In the latest version, the total number of software packages has exceeded 36,000. The community has categorized its software repositories into three types based on the sources, quality attributes, and package maintenance modes. You can configure software repositories according to their specific needs. The redistribution of software packages across different repositories is subject to community rules, which are based on usage, stability, and maintenance status.



Core/Base package repository

Software package repository of the openEuler LTS and innovation releases. All software packages in this repository have completed the end-to-end quality assurance of the openEuler community according to the software quality attribute specifications of the community.

Extension repository

A supplement to the openEuler LTS and innovation software package repository. The source code of the software packages in the repository all comes from the openEuler community. However, the software package release quality and maintenance support requirements of the openEuler community cannot be fully met due to the package quality, technical maturity, or participants of community developers.

Third-party package repository (open/closed source)

Software repository and build system provided by the openEuler community or third-party systems. Software packages in this repository are compatible with all openEuler releases and cannot break the compatibility and dependency of software packages of various openEuler releases. As a supplement to the openEuler software package repositories, this repository provides the widest possible selection of software packages for openEuler community users.

Open and Transparent Management of the Open Source Software Supply Chain

The process of building an open source OS relies on supply chain aggregation and optimization. A reliable open source software supply chain is fundamental to a large-scale commercial OS. openEuler combs through its software dependencies based on real user scenarios, sorts out the upstream community addresses of all the software packages, and verifies its source code in comparison to the upstream communities. This is a complete lifecycle management throughout build, verification, and distribution. The build, runtime dependencies, and upstream communities of the open source software form a closed loop, realizing a complete, transparent software supply chain management.

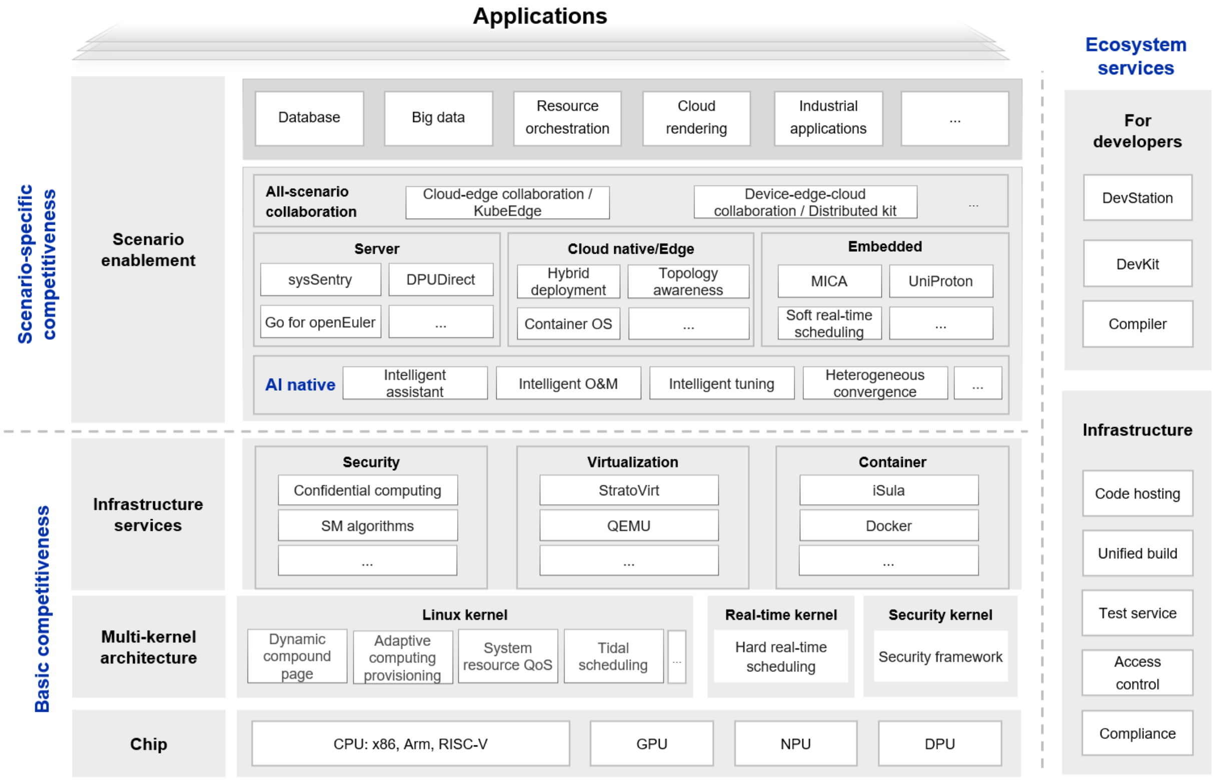
Community-certified openEuler Distributions

(Sorted by certification time): https://www.openeuler.openatom.cn/en/download/commercial-release/

Partner	System
Guangzhou Teligen Communication Technology Co., Ltd.	TeligenOS V3
Inspur Cloud Information Technology Co., Ltd.	InLinux 23.12 LTS SP2
Chengdu TD-tech Co., Ltd.	TDOS V1.0
Inspur Cloud Information Technology Co., Ltd.	InLinux 23.12 LTS SP1
Seaway Technology Co., Ltd.	SeawayEdge V1.00
Hunan Kylinsec Technology Co., Ltd.	Kylinsec OS V3.5.2

Partner	System
Chinasoft International Technology Services Co., Ltd.	CSIOS V1.0.0
Jiangsu HopeRun Software Co., Ltd.	HopeOS V22
Hundsun Technologies Inc.	HUNDSUN LightOS V1.0
Guangdong ZTE NewStart Technology Co., Ltd	NewStartOS Server V6
xFusion Digital Technologies Co., Ltd.	FusionOS 22 (free for use)
Inspur Cloud Information Technology Co., Ltd.	InLinux 23.12 LTS
iSoftStone Information Technology (Group) Co., Ltd.	ISSEOS V22
Beijing Linx Software Co., Ltd.	Linx secure OS V6.0
TurboLinux Inc.	TurboLinux Enterprise Server
UnionTech Software Technology Co., Ltd.	UnionTech OS Server 20
H3C Technologies Co., Ltd.	NingOS V3.0
China Mobile Cloud Energy Center	BCLinux for Euler V21.10
Red Flag Software Co., Ltd.	Asianux V8.1
Jiangsu HopeRun Software Co., Ltd.	HopeStage
Kylin Software Co., Ltd.	Kylin Advanced Server OS V10
Hunan Kylinsec Technology Co., Ltd.	Kylinsec V3.5.1
Hunan Kylinsec Technology Co., Ltd.	Kylinsec V3.4-5
Hunan Kylinsec Technology Co., Ltd.	Kylinsec V3.4-4
Jiangsu HopeRun Software Co., Ltd.	HopeEdge
Institute of Software, Chinese Academy of Sciences	EulixOS Server
iSOFT Infrastructure Software Co., Ltd.	iSoft Server OS V5.1
TongyuanOS	TYOS 8.1 Euler

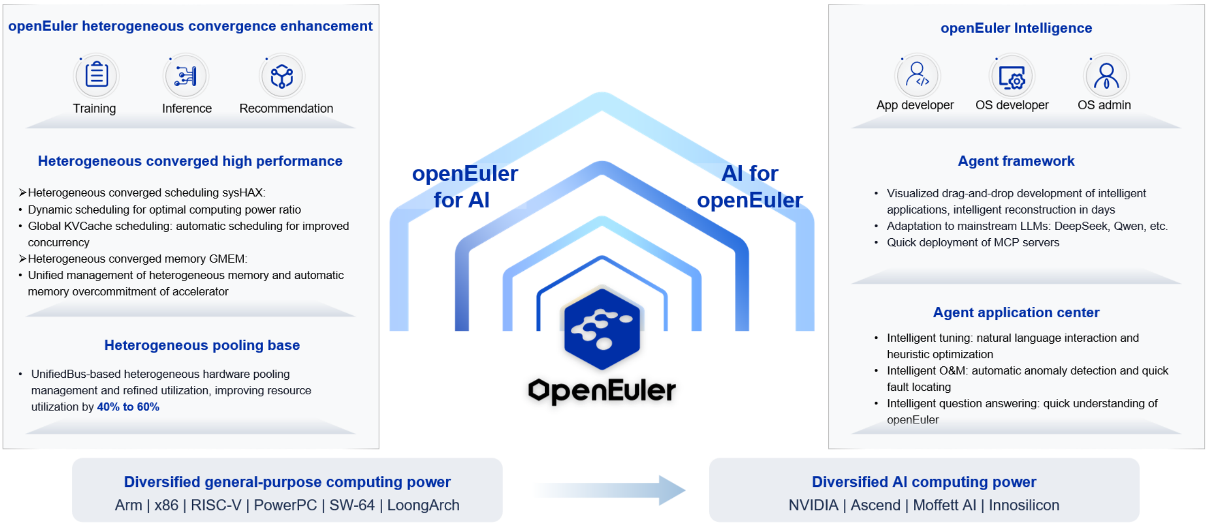
Architecture



3 AI Innovation

In the age of intelligence, OSs must continuously evolve to fully embrace AI. openEuler introduces a new design philosophy centered on AI for OS and OS for AI.

- AI for OS: openEuler integrates AI across the entire OS lifecycle—from development, deployment, and operation to maintenance and optimization—making the OS itself more intelligent.
- OS for AI: openEuler supports all mainstream general computing architectures, including Arm, x86, and RISC-V. By implementing heterogeneous converged scheduling and memory management, it enhances collective computing power and supports major AI processors. This makes openEuler the platform of choice for enabling diverse computing power.


openEuler heterogeneous convergence enhancement
Training
Inference
Recommendation
Heterogeneous converged high performance
➢Heterogeneous converged scheduling sysHAX:
• Dynamic scheduling for optimal computing power ratio
• Global KVCache scheduling: automatic scheduling for improved concurrency
➢Heterogeneous converged memory GMEM:
• Unified management of heterogeneous memory and automatic memory overcommitment of accelerator
Heterogeneous pooling base
• UnifiedBus-based heterogeneous hardware pooling management and refined utilization, improving resource utilization by 40% to 60%
openEuler for AI
AI for openEuler
OpenEuler
openEuler Intelligence
App developer
OS developer
OS admin
Agent framework
• Visualized drag-and-drop development of intelligent applications, intelligent reconstruction in days
• Adaptation to mainstream LLMs: DeepSeek, Qwen, etc.
• Quick deployment of MCP servers
Agent application center
• Intelligent tuning: natural language interaction and heuristic optimization
• Intelligent O&M: automatic anomaly detection and quick fault locating
• Intelligent question answering: quick understanding of openEuler
Diversified general-purpose computing power
Arm | x86 | RISC-V | PowerPC | SW-64 | LoongArch
Diversified AI computing power
NVIDIA | Ascend | Moffett AI | Innosilicon

OS for AI

sysHAX

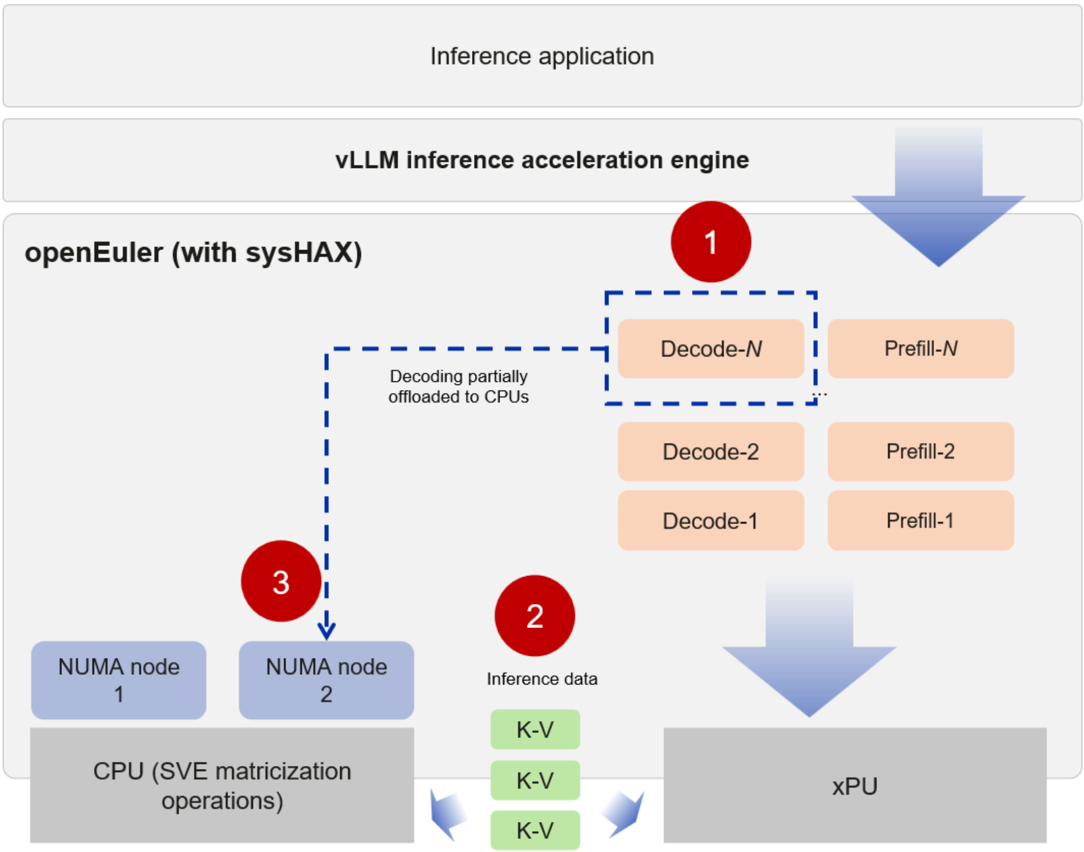
SIG	Intelligence

sysHAX boosts model inference performance by efficiently using both CPUs and xPUs. It enhances CPU functions, lowers computational demands for models, and increases their processing capacity.

Feature Description

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

- **Collaboration of CPU and xPU computing power**: Dynamically offloads partial decode tasks to CPUs based on xPU load conditions, including CPU-powered decode task filling (boosting throughput for small/medium LLMs) and CPU-executed MoE tasks (with dense MLA experts remaining on GPUs, enhancing MoE model throughput). The solution also provides parallel operator fusion capabilities to reduce CPU overhead.
- **CPU inference acceleration**: Improves CPU throughput via NUMA-aware scheduling, parallelized matricization operations, and SVE-optimized inference operators.



Application Scenarios

- The sysHAX LLM inference optimization solution is designed to support models based on the Transformer architecture, including DeepSeek, Qwen, Baichuan, and Llama. Specifically, the CPU inference acceleration capabilities have been adapted for the DeepSeek 7B, 14B, and 32B models, as well as the Qwen2.5 series.
- This solution is primarily suited for data center scenarios, where sysHAX utilizes its optimization techniques to offload inference tasks on CPUs. This strategy fully leverages available CPU resources, significantly increasing concurrency and throughput for LLMs.

Repository

https://gitee.com/openeuler/sysHAX

sysTrace

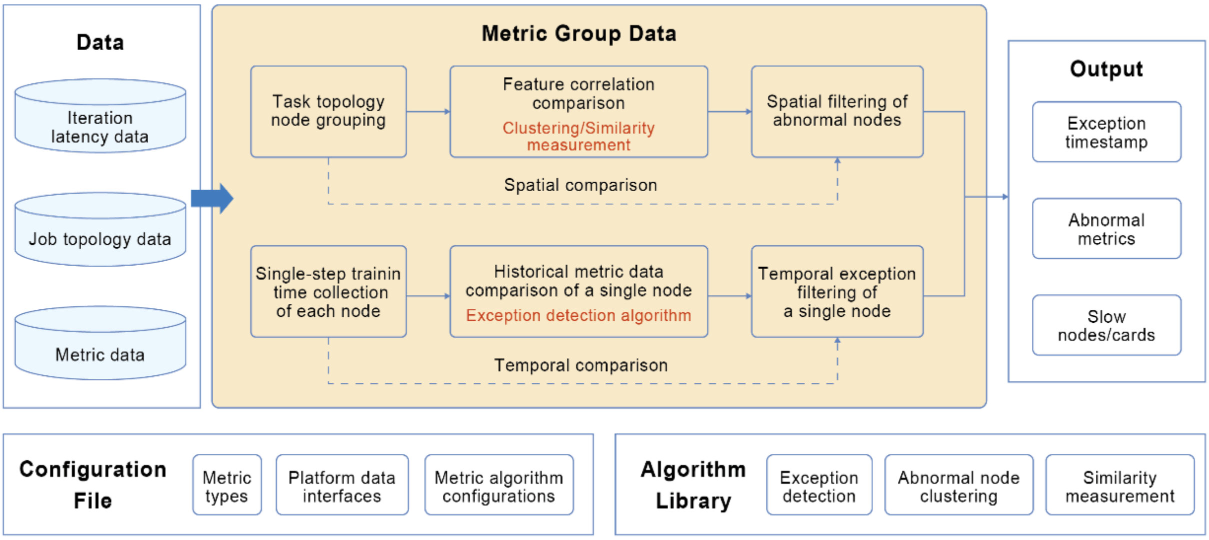
SIG	Ops

sysTrace is a performance monitoring tool designed for AI tasks. It enables efficient troubleshooting, comprehensive AI stack tracking, non-intrusive tracing, and quick issue diagnosis.

Challenges

Performance degradation during AI cluster training is inevitable and often results from a wide range of complex factors. Existing solutions rely on log analysis after performance degradation occurs. However, it can take 3 to 4 days from log collection to root cause diagnosis and issue resolution on the live network. To address the pain points, an online slow node detection solution is offered. This solution allows for real-time monitoring of key system metrics and uses model- and data-driven algorithms to analyze the observed data, pinpointing the location of slow or degraded nodes. This facilitates system self-healing or fault rectification by O&M personnel.

Feature Description


Data
Iteration latency data
Job topology data
Metric data
Metric Group Data
Task topology node grouping
Feature correlation comparison
Clustering/Similarity measurement
Spatial filtering of abnormal nodes
Spatial comparison
Single-step trainin time collection of each node
Historical metric data comparison of a single node
Exception detection algorithm
Temporal exception filtering of a single node
Temporal comparison
Output
Exception timestamp
Abnormal metrics
Slow nodes/cards
Configuration File
Metric types
Platform data interfaces
Metric algorithm configurations
Algorithm Library
Exception detection
Abnormal node clustering
Similarity measurement

Grouped metric comparison helps detect slow nodes and cards in AI cluster training. This technology is built on sysTrace and includes a configuration file, an algorithm library, and a slow node analysis mechanism based on both time and space dimensions. It outputs the exception timestamp, abnormal metrics, and IP addresses of slow nodes and cards. This technology enhances overall system stability and reliability. The following features are provided:

- Configuration file: Contains the types of metrics to be observed, configuration parameters for the metric algorithms, and data interfaces, which are used to initialize the slow node detection algorithms.
- Algorithm library: Includes common time series exception detection algorithms, such as streaming peaks-over-threshold (SPOT), k-sigma, abnormal node clustering, and similarity measurement.
- Data: Metric data collected from each node is represented by a time sequence.
- Grouped metric comparison: Supports spatial filtering of abnormal nodes and temporal exception filtering of a single node. Spatial filtering identifies abnormal nodes based on the exception clustering algorithm, while temporal exception filtering checks whether a node is abnormal based on the historical data of the node.

Application Scenarios

- sysTrace provides capabilities for detecting slow nodes, displaying alarms, and writing exception information to drives.

- AI model training: This feature quickly detects slow nodes for training jobs in large-scale AI clusters, facilitating system self-healing and fault rectification by O&M personnel.
- AI model inference: This feature identifies performance degradation across multiple instances of the same model. By comparing resource usage of multiple instances, it quickly locates underperforming instances to aid inference task scheduling and enhance resource utilization.

Repository

https://gitee.com/openeuler/sysTrace

GMEM

SIG	Ops

Generalized memory management (GMEM) is a unified memory management framework for heterogeneous computing. It provides centralized management for interconnected heterogeneous memories. The GMEM API is consistent with the native Linux APIs, offering strong usability, high performance, and good portability.

Challenges

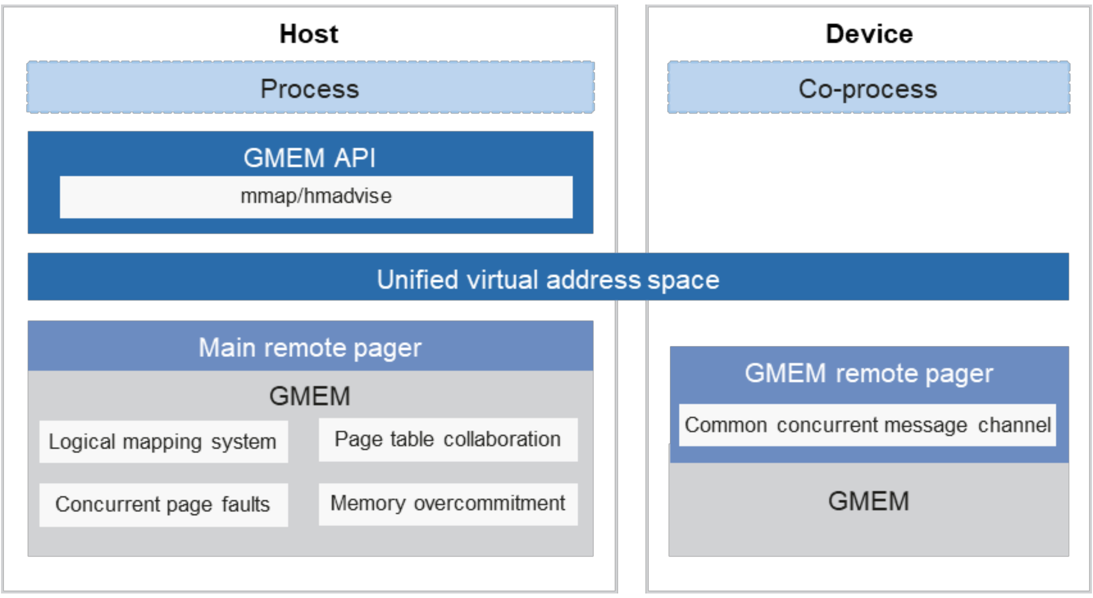
In the post-Moore era, specialized accelerators such as GPUs, TPUs, and FPGAs are rapidly emerging. Like CPUs, they rely on local memory (e.g., LPDDR or HBM) to achieve high computing efficiency. However, with diverse accelerator types, vendors often need to develop complex and customized memory management systems.

Modern memory systems have the following defects:

- Data management between CPUs and accelerators is separated. Explicit data migration makes it difficult to balance the usability and performance.
- The HBM memory on accelerators is often insufficient, and manual swapping leads to significant performance loss and poor portability.
- In workloads such as search, recommendation, and big data, a large number of invalid data movements occur due to the lack of efficient memory pooling.
- Existing Linux heterogeneous memory management (HMM) frameworks are complex, require manual tuning, and are no longer actively maintained. Despite integration efforts by NVIDIA and AMD, architectural incompatibilities have led to limited code portability and inconsistent performance across platforms, prompting concerns within the upstream OS communities.

The heterogeneous computing ecosystem urgently needs a unified and efficient memory management mechanism. To address these challenges, openEuler introduces GMEM, a generalized framework designed for ease of use, high performance, and strong portability.

Feature Description



GMEM provides centralized management for heterogeneous interconnect memories, redefining Linux kernel memory management architecture. Its logical mapping system masks address access differences between CPUs and accelerators, while the remote_pager framework abstracts memory message communication and device access. Under a unified address space, GMEM automatically migrates data to the OS or accelerator when data is to be accessed or paged.

- **Heterogeneous memory**

  To integrate the computing power of both accelerators and CPUs, GMEM introduces a unified virtual memory address space, merging the previously independent OS and accelerator address spaces.

  A logical page table is built to maintain this unified space, ensuring consistency across processors and microarchitectures. Based on this mechanism, memory pages can be automatically migrated between the host and accelerators during page faults. When accelerator memory is insufficient, it can borrow host memory while reclaiming its cold memory to achieve memory overcommitment, enabling cost-effective model training without being limited by accelerator memory capacity.

  GMEM provides high-level kernel APIs that allow accelerator drivers to register MMU functions defined by GMEM specifications. This decouples high-level memory management logic from CPU hardware dependencies, enabling different accelerators to reuse a unified memory management logic. In short, accelerators only need to register low-level functions without implementing complex unified address management logic.

- **Remote pager**

  The remote pager acts as an extended OS memory management framework. It provides mechanisms for message communication, process management, memory swapping, and prefetching between the host and accelerators. Enabled by the independent driver **remote_pager.ko**, it offers an abstraction layer that allows third-party accelerators to easily integrate with the GMEM system, greatly simplifying device adaptation.

- **User APIs**

  Users can allocate unified virtual memory directly using mmap, with GMEM introducing a new flag (**MMAP_PEER_SHARED**) for this purpose.

The user-mode library **libgmem** also provides an hmadvise API for memory prefetching, helping users optimize accelerator memory access efficiency.

- **Limitations**
  - Currently, only 2 MB huge pages are supported; transparent huge pages must be enabled by default on both hosts and NPUs to use GMEM.
  - Heterogeneous memory allocated with **MAP_PEER_SHARED** is not inherited during fork.

Application Scenarios

- **Unified heterogeneous memory programming**

  With GMEM, developers can allocate unified virtual memory shared between CPUs and accelerators, allowing both to access data through the same pointer. This greatly simplifies heterogeneous programming. Taking NPU as an example, driver integration with GMEM required only a few hundred lines of code—replacing around 4,000 lines of the original memory management logic.

- **Automatic accelerator memory overcommitment**

  When memory is allocated through GMEM, it is no longer limited by the accelerator's physical memory capacity. Applications can transparently over-commit memory up to the host's DRAM capacity. GMEM automatically swaps cold accelerator memory pages to CPU, achieving high-performance and easy-to-start training and inference. In ultra-large model training, GMEM has demonstrated superior performance to NVIDIA-UVM, with advantages increasing as memory usage scales. When the overcommitment ratio is 200%, GMEM achieves over 60% performance improvement compared with NVIDIA-UVM (tested on the Ascend 910 NPU and NVIDIA A100 GPU under the same HBM conditions).

Repositories

https://gitee.com/openeuler/libgmem

AI for OS

openEuler Intelligence

SIG	Intelligence

openEuler Intelligence is an LLM-powered platform built on openEuler. It combines features like semantic interface registration, workflow orchestration and scheduling, and a knowledge base to offer essential AI services. Users can also integrate local custom APIs for advanced intelligent services and intelligent application upgrade.

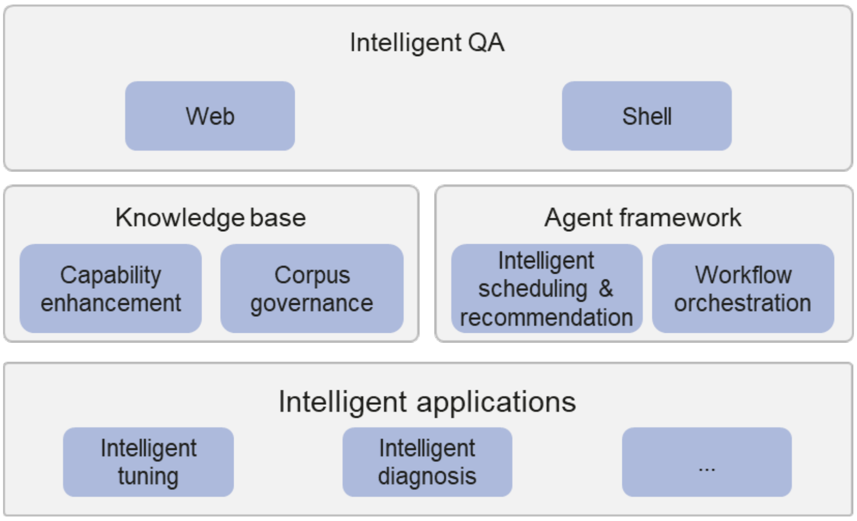
Agent Application Center

openEuler Intelligence offers an Agent Application Center, supporting intelligent question answering (QA), tuning, and diagnosis.

QA Agent

The QA agent helps users create and use knowledge bases, test their accuracy, and utilize workflow apps built on the agent framework. It simplifies openEuler knowledge access and AI adoption for beginners.

Feature Description



openEuler Intelligence currently provides two access options: Web and Shell.

- **Web**

  Users can interact with the system via a visualized QA interface.
- **Shell**

  Users can leverage their openai-api-key to access the intelligent agent framework for smart QA or execute predefined workflows.

Intelligent scheduling and recommendation

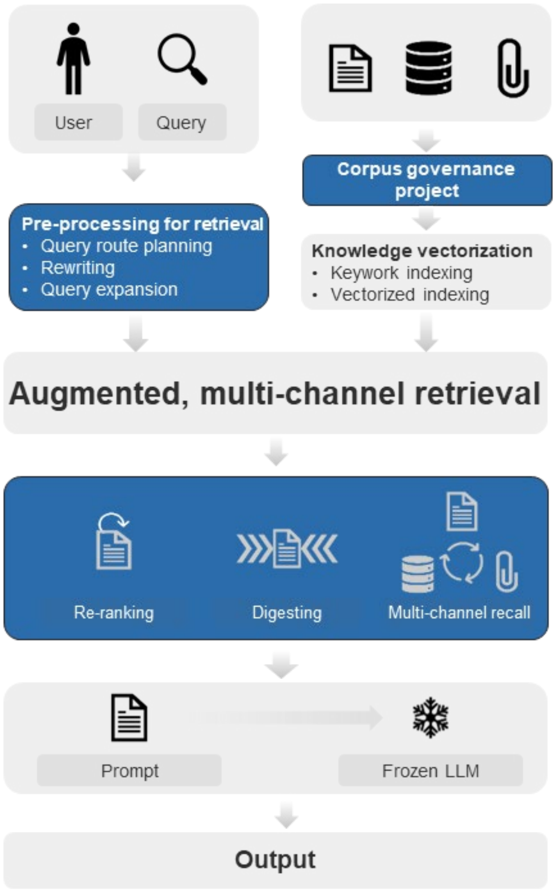
- **Intelligent scheduling**

  openEuler Intelligence allows users to define multiple workflows within an application. Based on the user's query, the system automatically extracts relevant parameters and selects the most suitable workflow to execute.
- **Intelligent recommendation**

  By analyzing user queries and workflow results, openEuler Intelligence recommends the next possible workflow, helping users complete tasks more efficiently and simplifying the overall experience.

Improved answering accuracy

Retrieval-augmented generation (RAG) enhances LLMs by allowing them to reference external knowledge bases beyond their original training data before generating responses. openEuler Intelligence leverages the RAG technology to improve pre- and post-processing for retrieval, knowledge indexing, and retrieval enhancement algorithms. It supports diverse document formats and content types, boosting answer accuracy and user experience with minimal impact on system load.



Corpus governance

Corpus governance is one of the basic RAG capabilities in openEuler Intelligence. It imports corpuses into the knowledge base in a supported format using context location extraction, text summarization, and OCR augmentation, increasing the retrieval hit rate.

- **Context location extraction**

  The relative location relationship within a document is retained. Specifically, the global and local relative offsets of each segment are recorded to support context completion.

- **Text summarization**

  For complex documents or text segments, sliding windows combined with LLMs generate summaries to provide foundational data for subsequent multi-level retrieval.

- **OCR augmentation**

  For documents containing both images and text, summaries are generated based on image text and surrounding context to provide foundational data for answering image-related questions.

Application Scenarios

- **Knowledge acquisition**

  openEuler Intelligence provides users with access to essential knowledge of openEuler, including migration guides, system fundamentals, and dynamic data insights.

- **Development and contribution enablement**

  For developers, openEuler Intelligence serves as an intelligent assistant offering comprehensive guidance on contribution workflows, key feature exploration, and project development practices.

- **Intelligent O&M**

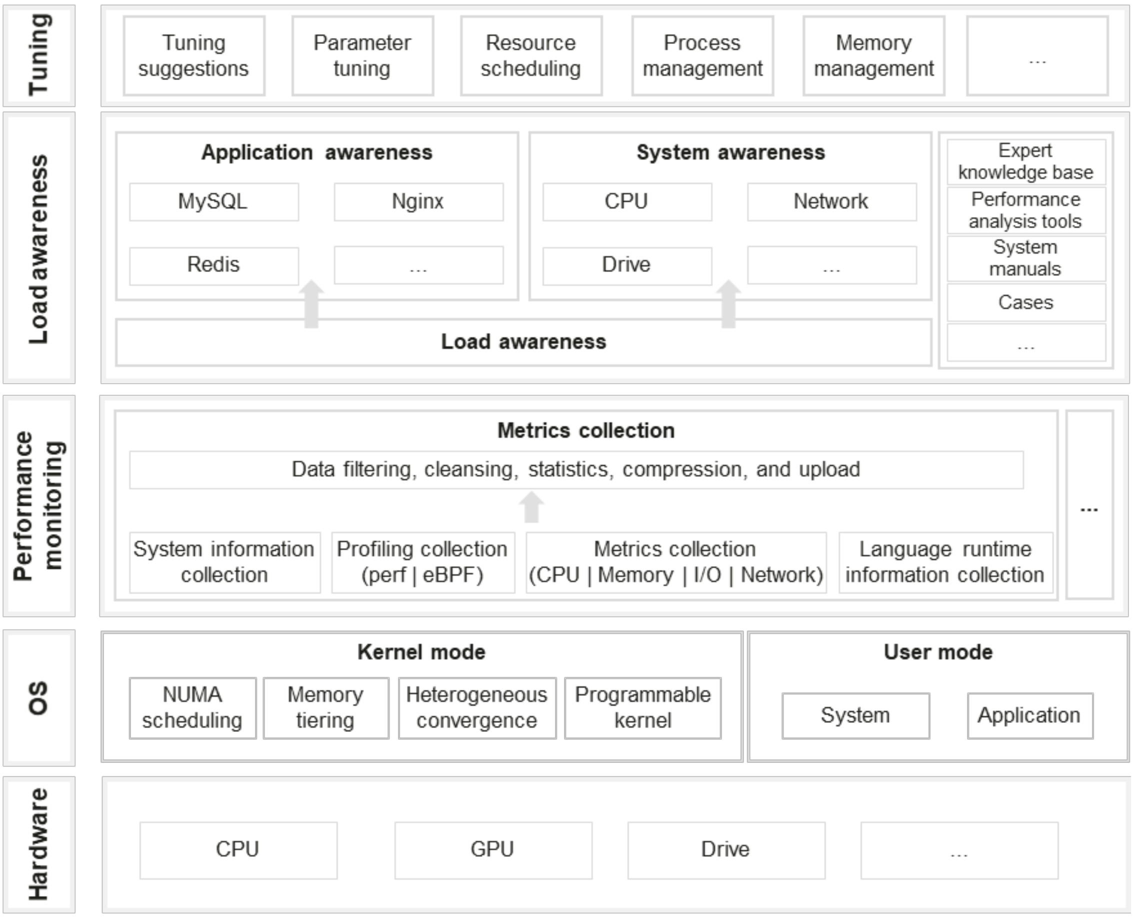
  For O&M personnel, openEuler Intelligence delivers data-driven solutions to diagnose and resolve common or complex issues.

Tuning Agent

Feature Description

Intelligent tuning is accessible via shell.

Users can interact with openEuler Intelligence using natural language to collect performance data, analyze system performance, and tune system performance.



Application Scenarios

- **Rapid acquisition of key performance metrics**

  Quickly obtain performance metrics across multiple dimensions from CPU, I/O, drive, and network as well as for specific applications, helping users gain a clear and comprehensive view of system performance data.

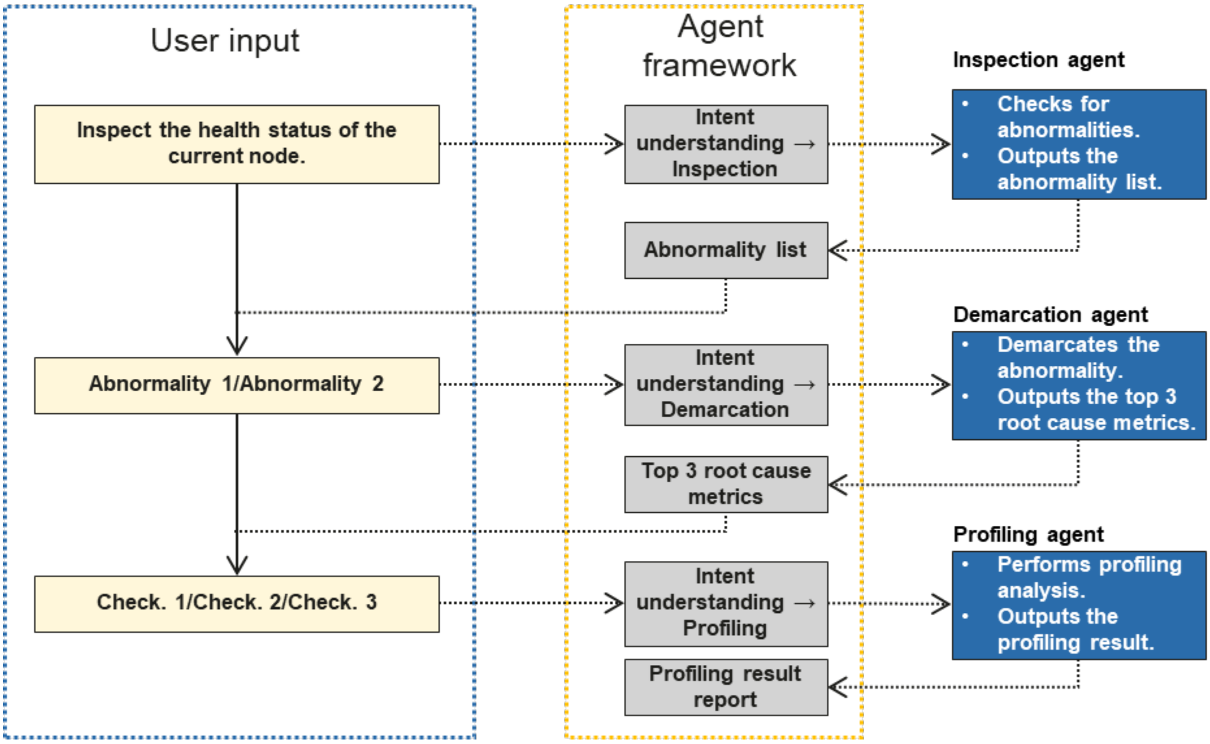
- **Analyzing system performance**

  Automatically generate performance analysis reports that evaluate system and application performance from multiple dimensions (CPU, I/O, drive, and network). The reports also identify potential performance bottlenecks within the system.

- **Performance Optimization Recommendations**

  Generate one-click executable optimization scripts based on the analysis results. After reviewing the script content, users can run it to optimize configurations for both the system and selected applications.

Diagnosis Agent


User input
Inspect the health status of the current node.
Abnormality 1/Abnormality 2
Check. 1/Check. 2/Check. 3
Agent framework
Intent understanding → Inspection
Abnormality list
Intent understanding → Demarcation
Top 3 root cause metrics
Intent understanding → Profiling
Profiling result report
Inspection agent
Checks for abnormalities.
Outputs the abnormality list.
Demarcation agent
Demarcates the abnormality.
Outputs the top 3 root cause metrics.
Profiling agent
Performs profiling analysis.
Outputs the profiling result.

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 root cause metrics.
- Profiling: The profiling agent performs profiling 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 24.03 LTS SP2, the intelligent diagnosis enables capabilities like single-node abnormality inspection, demarcation, and profiling.

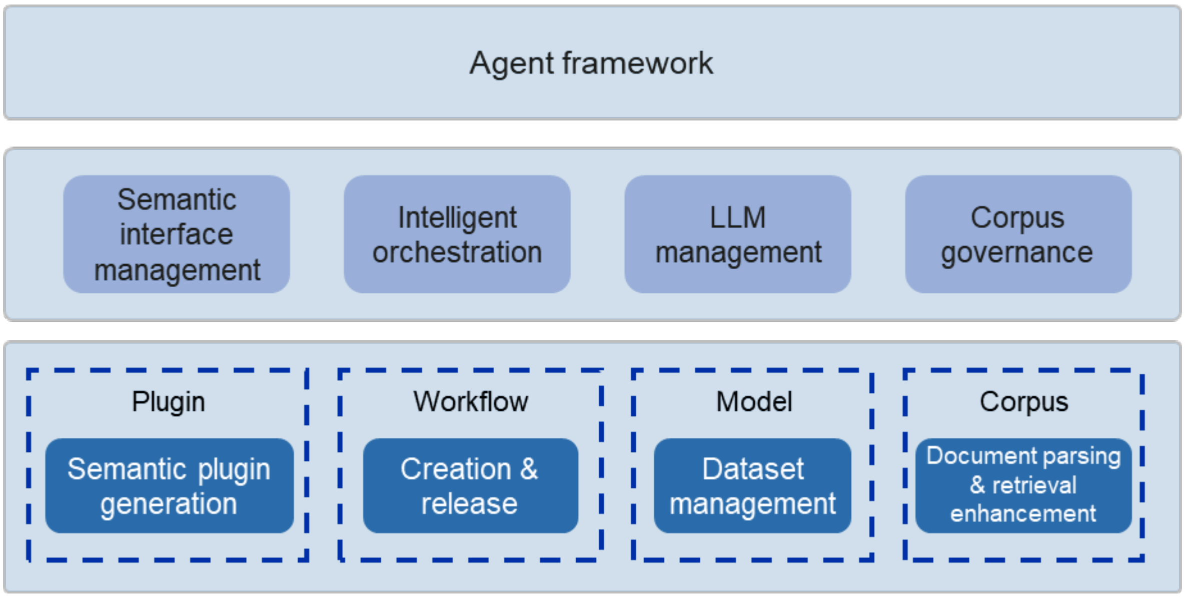
- The inspection capabilities refer to single-node performance metrics collection, performance analysis, and abnormality inspection.

- The demarcation capability is to locate the root cause based on the abnormality inspection result and output the top 3 root cause metrics.
- The profiling capability refers to using a profiling tool to locate the faulty modules (code snippets) based on the root cause.

Agent Framework

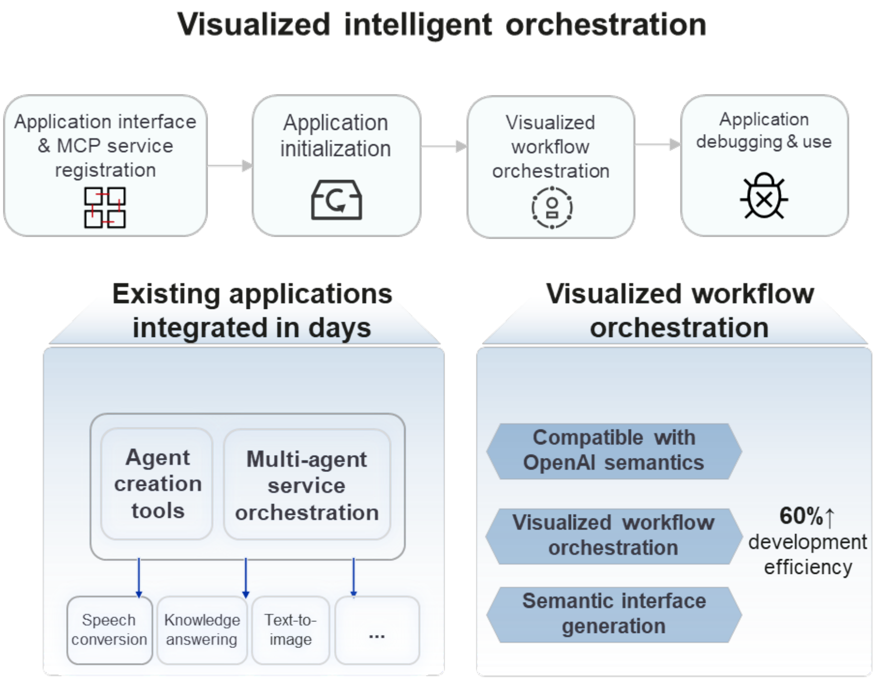
openEuler Intelligence's agent framework provides core features for developing agents. It allows enterprises to quickly build their own agents within days and supporting visual workflow orchestration.

The framework supports semantic interface registration (interfaces with natural language comments), MCP service, agent building, and workflow application. Both web and client modes are available, offering developers and enterprises flexible and efficient development experiences.


Agent framework
Semantic interface management
Intelligent orchestration
LLM management
Corpus governance
Plugin
Semantic plugin generation
Workflow
Creation & release
Model
Dataset management
Corpus
Document parsing & retrieval enhancement

Feature Description

openEuler Intelligence enables users to drag and drop system-provided semantic interfaces, user-registered interfaces, and MCP services to visually build, debug, and deploy workflows as applications. Intermediate results are displayed during debugging and execution to reduce costs and improve user experience.


Visualized intelligent orchestration
Application interface & MCP service registration
Application initialization
Visualized workflow orchestration
Application debugging & use
Existing applications integrated in days
Agent creation tools
Multi-agent service orchestration
Speech conversion
Knowledge answering
Text-to-image
...
Visualized workflow orchestration
Compatible with OpenAI semantics
Visualized workflow orchestration
Semantic interface generation
60%↑ development efficiency

Application Scenarios

The agent framework allows enterprises and individuals to easily build agent applications that enable intelligent workflow orchestration and scheduling in real-world production environments.

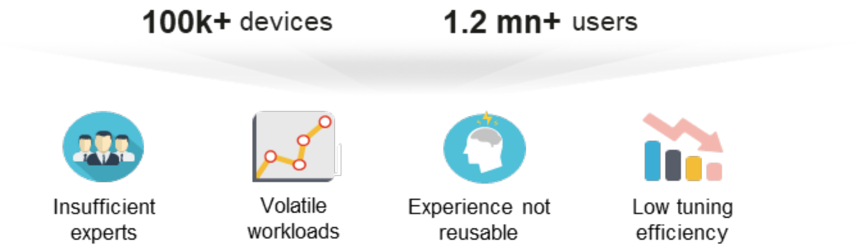
Success Cases

CTyunOS Achieves Intelligent Tuning Through openEuler Intelligence

CTyunOS is an operating system developed by eSurfing Cloud based on openEuler. Featuring intelligent tuning and other innovative technologies, it fuels China Telecom's cloudification & digital transformation strategy. The OS has achieved large-scale deployment across carrier, state-owned enterprises, and public sectors, effectively driving digital economy growth.

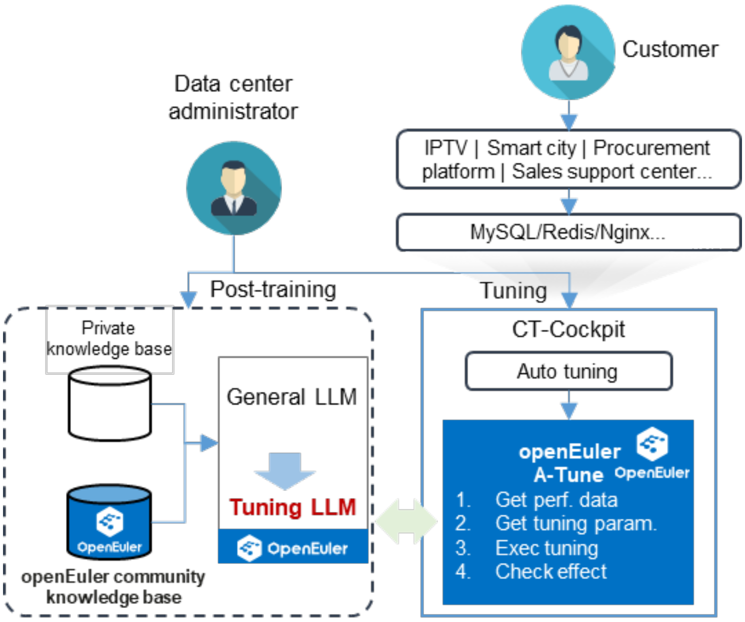
Application Scenarios

eSurfing Cloud is among China's biggest telecom clouds. It handles numerous operating systems, operates on a massive scale, and experiences constant scenario shifts, making application optimization challenging.



Solution

openEuler-based CTyunOS integrates openEuler Intelligence into CT-Cockpit for intelligent tuning.



Customer Benefits

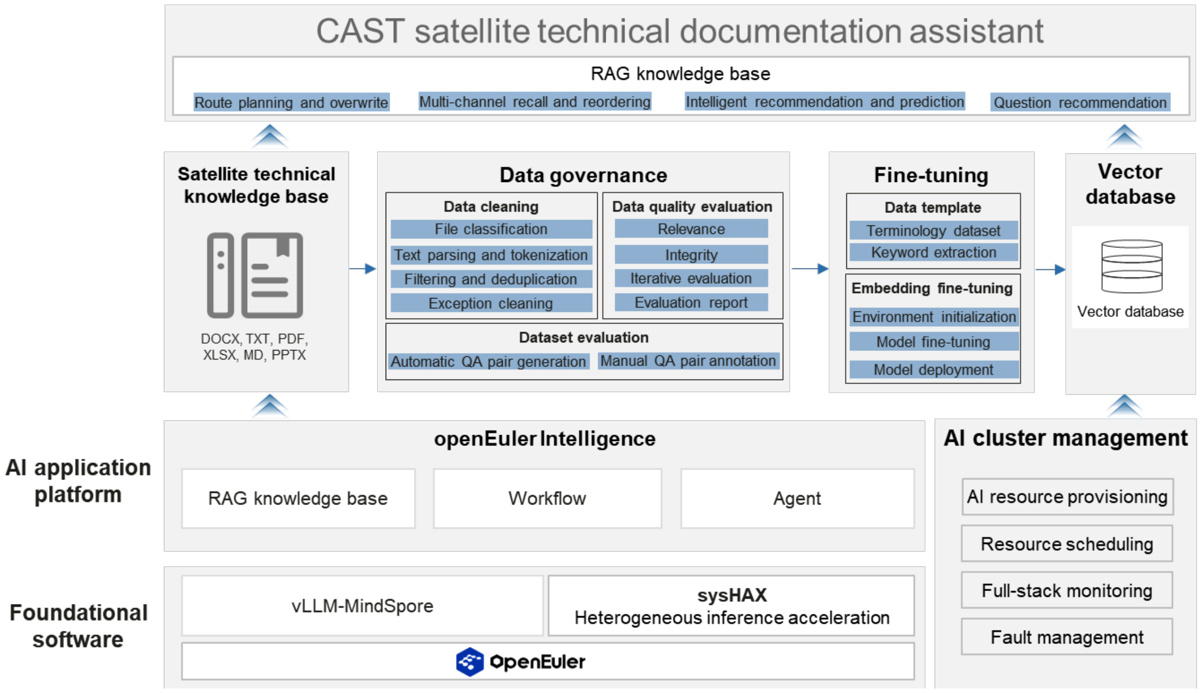
- Large-scale complex application tuning now takes hours instead of days, and no expert help is needed.
- AI-powered tuning improves performance by 10%.

openEuler Intelligence Empowers Huakun to Build a Satellite Technical Documentation Assistant for CAST

The satellite technical documentation assistant supports chatbot-style search, document generation, abstract extraction, event allocation, semantic matching, guidance, approval assistance, proofreading, meeting assistant, and supervision. This makes it a powerful tool for entire office-related workflows, improving work efficiency and quality.

Solution

openEuler Intelligence has empowered Sichuan Huakun Zhenyu Intelligent Technology (short as Huakun) to develop an interactive one-stop satellite technical documentation assistant for China Academy of Space Technology (CAST), which is the fifth research academy under China Aerospace Science and Technology Corporation (CASC).



Customer Benefits

- 10% higher throughput due to software and hardware collaboration that fully utilizes CPU/xPU computing power
- Network isolation and data encryption to ensure data security
- Low-latency concurrent processing, allowing more than 3000 documents to be imported to the database in minutes
- Reliable answers, with a high accuracy of at least 80%
- 50% quicker application development with efficient, visualized pipeline orchestration of AI applications and function call and MCP support

Repositories

https://gitee.com/openeuler/euler-copilot-framework

https://gitee.com/openeuler/euler-copilot-rag

https://gitee.com/openeuler/euler-copilot-witchaind-web

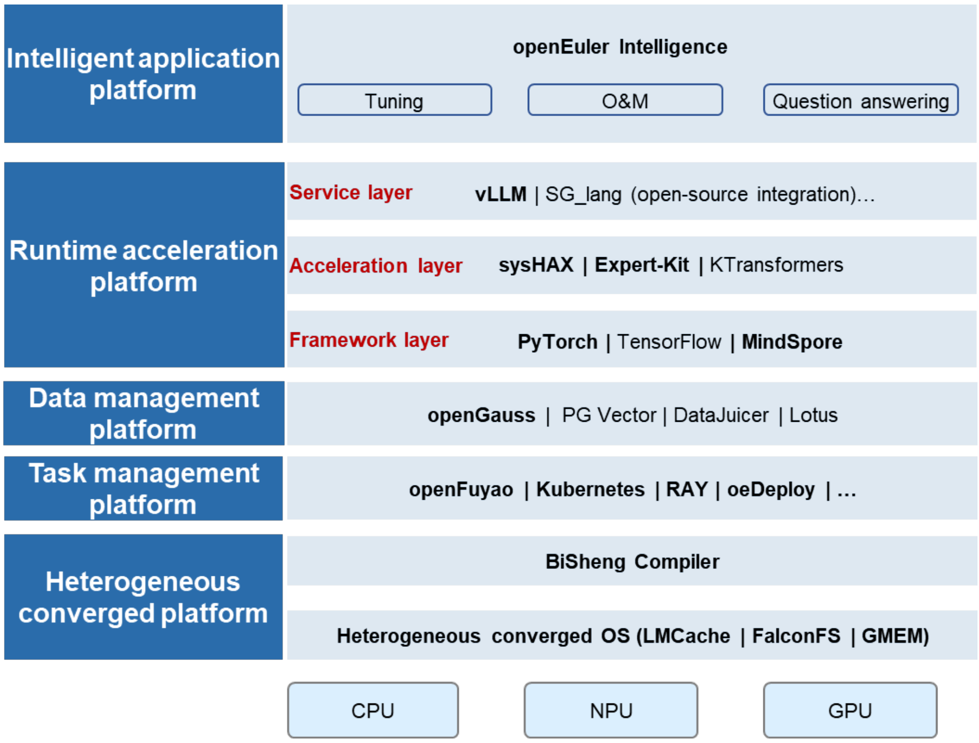
https://gitee.com/openeuler/euler-copilot-web

Ease of Use

openEuler provides an open-source AI software stack called Intelligence BooM, which integrates OS, databases, AI fabrics, and model optimization tools. openEuler enables AI-powered OS development, O&M, deployment, and tuning, delivering an efficient, out-of-the-box environment for development and running.

Feature Description

openEuler integrates installation, intelligent interaction, AI fabrics, and heterogeneous computing power, offering installable, out-of-the-box AI capabilities.


Intelligent application platform
openEuler Intelligence
Tuning
O&M
Question answering
Runtime acceleration platform
Service layer
vLLM | SG_lang (open-source integration)…
Acceleration layer
sysHAX | Expert-Kit | KTransformers
Framework layer
PyTorch | TensorFlow | MindSpore
Data management platform
openGauss | PG Vector | DataJuicer | Lotus
Task management platform
openFuyao | Kubernetes | RAY | oeDeploy | …
Heterogeneous converged platform
BiSheng Compiler
Heterogeneous converged OS (LMCache | FalconFS | GMEM)
CPU
NPU
GPU

1. openEuler Intelligence is the entry of openEuler for enterprises and developers to build AI applications for intelligent O&M, office work, and data analysis more efficiently through graphical operations and low-code development capabilities.
2. openEuler is compatible with major AI software stacks (drivers, SDKs, training and inference frameworks, and models) in both northbound (applications) and southbound (hardware) ecosystems to ensure seamless interconnection, providing a solid support for AI applications.
3. openEuler supports heterogeneous converged computing power, with Generalized Memory Management (GMEM) efficiently managing memory resources to reduce system memory fragments and sysHAX dynamically balancing CPU and xPU resources and offloading decoding to CPU for optimal computing efficiency.
4. openEuler provides a deployment tool called oeDeploy to enable common AI development software and toolchains to be quickly deployed.

Application Scenarios

Deeply integrated with AI and open-source AI software stack Intelligence BooM, openEuler is applicable where AI-native development is needed to make OS more intelligent and agent ecosystem more mature.

4 Scenario-specific Innovations

Server

sysSentry

SIG	Base Service

sysSentry is a fault inspection framework that enables fault inspection and diagnosis of system CPUs, memory, drives, and NPUs in the background.

Challenges

Unified OS-level fault management is not available for hardware faults, resulting in fragmented fault information and scattered reliability guarantee. System software and hardware faults cannot be identified in advance and quickly notified to O&M personnel, increasing the possibility of network incidents and weak reliability. This is where sysSentry was built to make a difference.

Feature Description

sysSentry features the following functions.

- Fast fault reporting

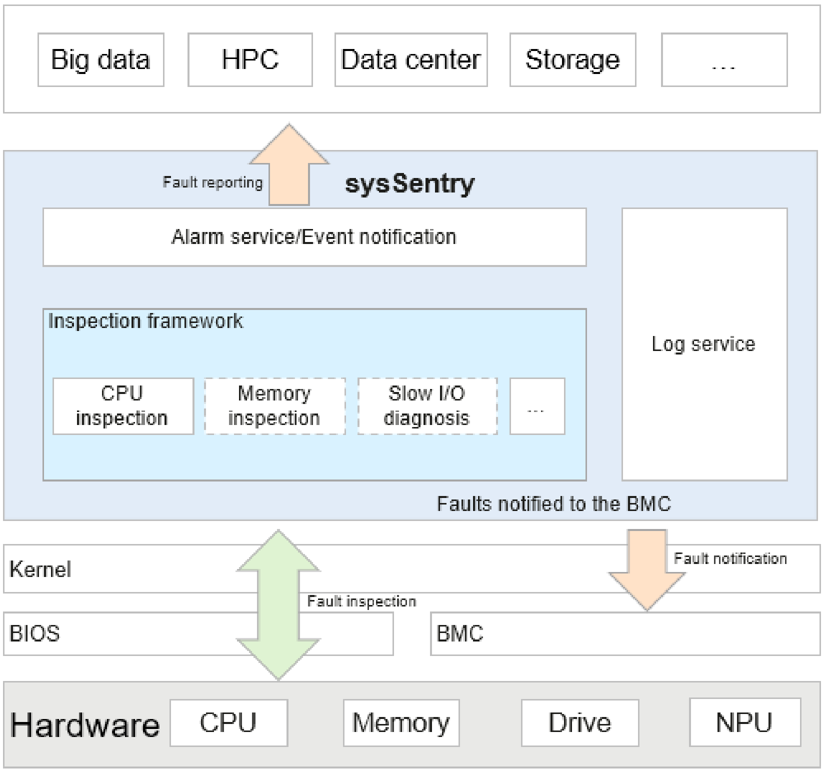
  Instead of being reported, faults are unsolicitedly notified through service management services. Based on the time specified for fault notification through subscription, faults are notified in seconds, which helps quickly restore services.

- High detection accuracy

  CPU fault detection holistically considers the kernel, BIOS, and chips at both the software and hardware levels to ensure high detection accuracy. 98% of known CPU hardware failures can be detected. The inspection capabilities of memory, drives, and xPU will also come soon.

- Plug-in management to ease expansion

  The inspection function supports plug-in management, allowing different hardware to be quickly supported in days by adding configurations, without modifying framework codes.



Application Scenarios

sysSentry is applicable to server maintenance where faults must be accurately detected and quick and proactive fault reporting are required alongside the ease of framework expansion.

Repositories

https://gitee.com/openeuler/sysSentry

DPUDirect

SIG	DPU

DPUDirect provides a collaborative operating environment for services to be flexibly offloaded and ported between the host and DPU. The feature supports process-level seamless offload and host-DPU collaboration, enabling management-plane processes to be split in a reconstruction-free fashion and seamlessly offloaded to the DPU. Though being offloaded, the processes can still manage service processes on hosts. Therefore, DPUDirect is in a good position to greatly reduce the cost of offloading services to a DPU, simplifying O&M and slashing maintenance costs.

Challenges

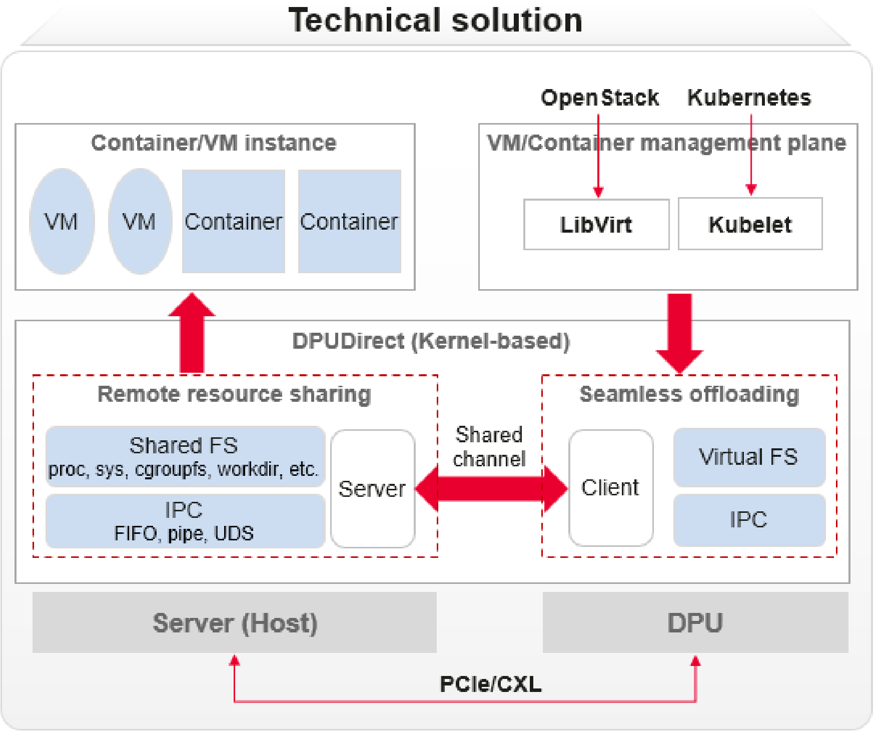
DPUs are becoming increasingly important for data centers. Offloading management-plane processes to DPUs based on userspace splitting schemes requires roughly 10,000 new lines of code. This complicates upgrades and maintenance and also risks CSP lock-in, pushing carriers and other customers to explore open, standard, and transparent seamless offloading in typical scenarios.

Feature Description

DPUDirect supports remote host resource sharing and cross-node maintenance of file and IPC access semantics. This enables management-plane services to be efficiently offloaded, while reducing the lines of code required for application adaptation from more than 10,000 to less than 500 during software porting and maintenance.

Remote resource sharing: Through a client/server resource-sharing architecture, VM and container management directories like proc, sys, cgroupfs, and working directories are shared across nodes (host and DPU), which provides seamless access channels.

Seamless offloading: File system and IPC access semantics are maintained across nodes, enabling efficient and seamless offloading of VM and container management planes from the host to DPU.



Application Scenarios

DPUDirect is applicable to the full seamless offloading of container/VM management processes (kubelet and dockerd for container and libvirtd for VM) to a DPU without splitting over 10,000 lines of code (that is, nearly 20 times less adaptation/maintenance workload) and altering the management-plane service logic to ensure compatibility and evolution.

Repositories

https://gitee.com/openeuler/dpu-utilities

vDPA

SIG	Virt

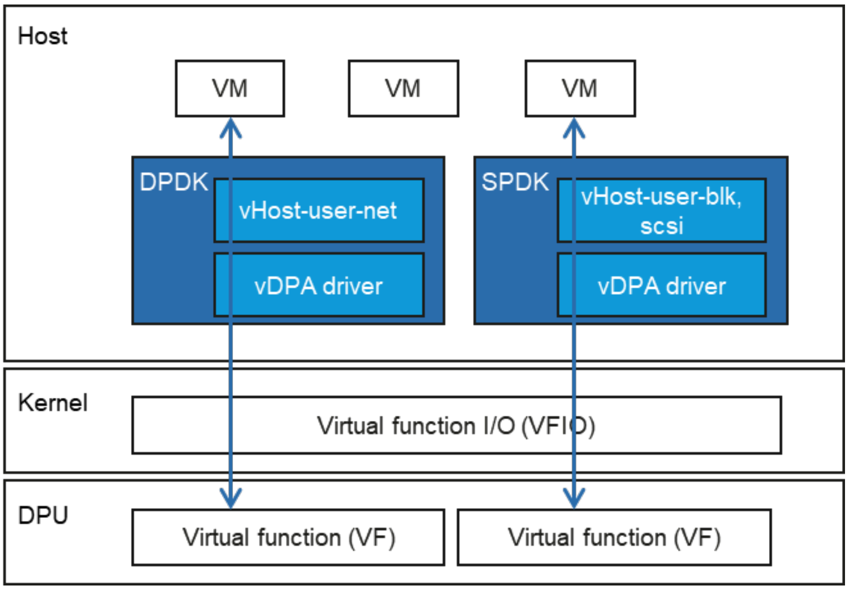
Device virtualization is changing from full virtualization to virtual I/O (VirtIO) paravirtualization, vHost data-plane offloading, and virtual function I/O (VFIO) passthrough. This trend benefits data planes with higher I/O performance and control planes with greater device management flexibility. The kernel-mode vDPA framework enables efficient data-plane passthrough while supporting network, storage, and other devices and ensuring high-performance live VM migration.

Challenges

Common open-source vHost Data Path Acceleration (vDPA) solutions are built on data plane development kit (DPDK) and storage performance development kit (SPDK), as illustrated in the figure below. A DPU creates a virtual function (VF) on the host-side devices through single-root I/O virtualization (SR-IOV). With user-mode vDPA, the VF is managed by the DPDK and SPDK through the Virtual Function I/O (VFIO) driver. The DPDK and SPDK present vHost-user socket devices externally and uses backend types (qemu vhost-user-net, vhost-user-blk, and vhost-user-scsi) to present VirtIO devices to VMs.

The user-mode vDPA solution faces a number of problems as below.

- Extensive host resource consumption: User-mode vDPA needs to run DPDK and SPDK components on the host to present user-mode vHost devices. This means extra CPU and memory resource consumption on the host, offsetting the benefits of using DPUs to offload management process from the host.
- Inconsistent management interface: The vhost-user-net, vhost-user-blk, and SCSI devices are managed by DPDK and SPDK components, as shown in the architecture diagram, making it impossible to use a unified solution. Besides, only storage and network devices are supported, meaning other VirtIO devices such as VirtIO-FS will not be covered.
- Prolonged live migration: The live migration of user-mode vDPA VMs takes more than one second, causing obvious service suspension during upgrade and O&M.

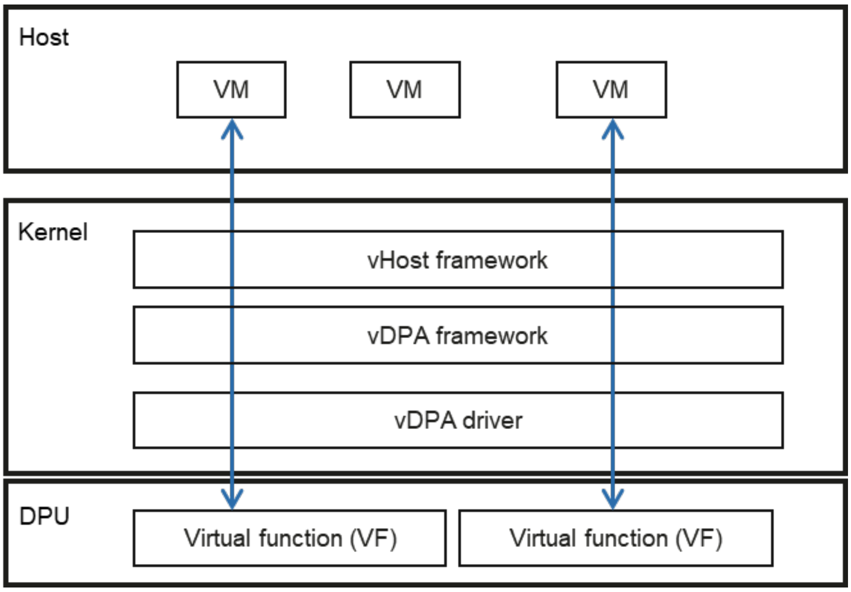


Feature Description

Similar to building the user-mode DPDK, openEuler uses the kernel's existing subsystem vHost to build the kernel-mode vDPA that provides APIs for upper-layer user-mode software.

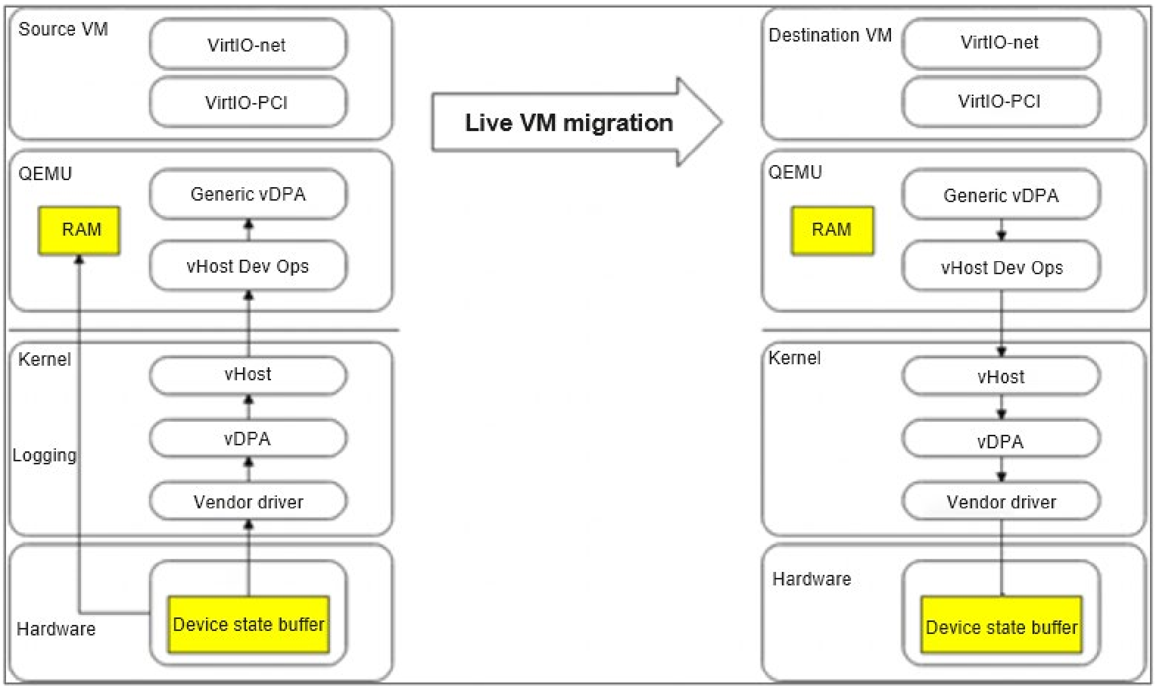
The vHost module receives input/output control (IOCTL) commands from the QEMU and uses the vHost-vDPA module to forward them to the vDPA framework, which then delivers them to a device-bound vDPA driver to implement hardware settings.

Therefore, kernel-mode vDPA requires no additional user-mode process on the host to consume extra resources like CPU and memory, increasing the utilization of host resources. Kernel-mode vDPA does not consider the device types of the VF so that VirtIO devices like virtio-net, blk, scsi, and fs can be presented to VMs using the same solution.



Combined with the following capabilities, kernel-mode vDPA supports efficient live VM migration.

- vDPA device dirty page marking: The vDPA device supports hardware dirty page marking to ensure memory consistency between the source and destination hosts after live VM migration.
- vDPA device suspension and resumption: The vDPA device can be suspended during live VM migration and quickly resumed after the migration.
- vDPA device status migration: The vDPA device status can be migrated from the source host to the destination host to ensure consistency.



Application Scenarios

vDPA is applicable to intelligent NIC (iNIC) and DPU offloading in cloud computing, where a full offloading of host management resources (supporting network, storage, and other devices) and efficient live VM migration are needed to increase sellable host resources while reducing the costs for cloud vendors and service downtime caused by VM migration.

Repositories

https://gitee.com/openeuler/qemu

https://gitee.com/openeuler/kernel

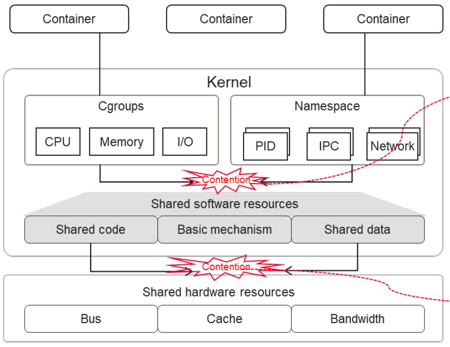
Cloud Native and Virtualization

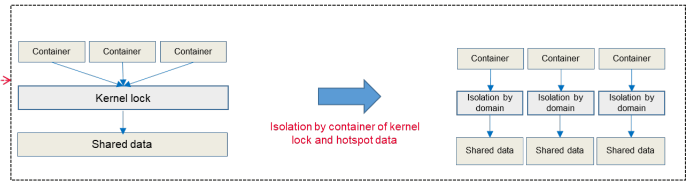
Many-Core High-Density Architecture

SIG	Kernel

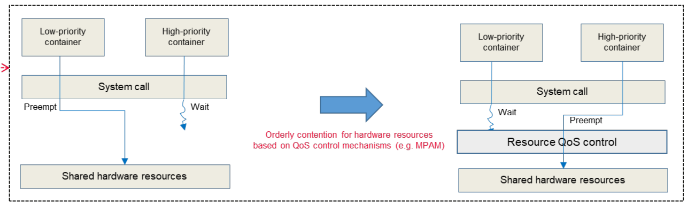
Server chips have evolved from multi-core to many-core architectures (with over 256 cores), pushing operating system limits. Many-core servers dominate modern internet infrastructure, boosting per-rack computing density while slashing data center TCO. With advancing cloud technology and growing service demands, containerized deployments have become popular. However, in container environments, system scalability is crippled by high serialization and synchronization overheads resulting from contention for shared hardware or software resources. In consequence, interference and inefficient resource utilization become increasingly prominent.

Feature Description





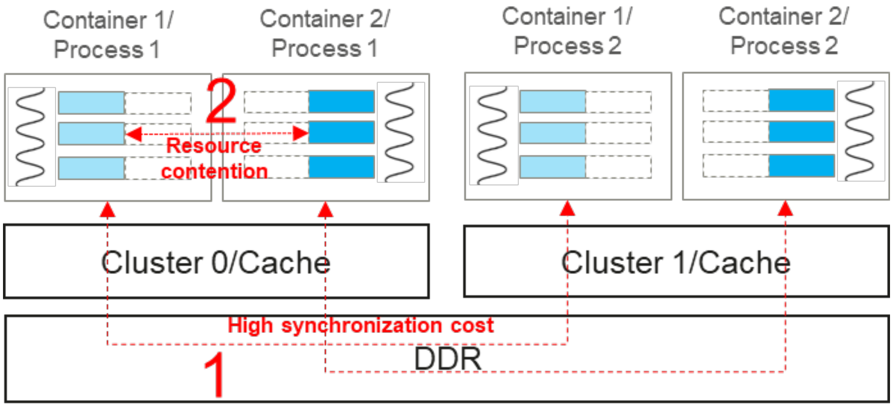
Parallel access to shared software resources (vs. serial)



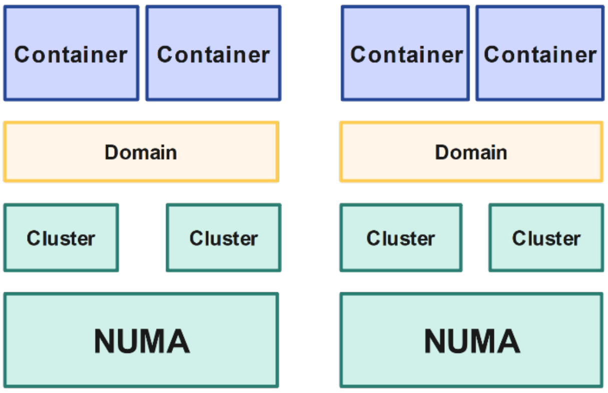
Orderly access to shared hardware resources (vs. chaotic)

openEuler uses lightweight virtualization to divide resources by NUMA domain and enforce container-level resource isolation in each domain. This reduces performance problems resulting from hardware or software resource conflicts and improves container deployment scalability. Its key features are:

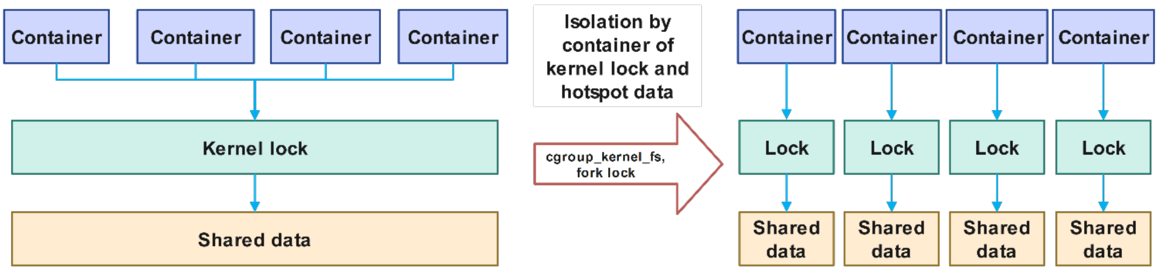
- Lightweight virtualization: Virtual device interrupt offloading allows VirtIO device interrupts to be handled by hardware, reducing storage virtualization overhead. Poll Mode Driver (PMD) queue load balancing spreads VirtQueues from a single reactor to multiple reactors, eliminating CPU bottlenecks.
- CPU scheduling by domain: CPUs are divided into domains by cluster for container deployment. Each container operates within an independent scheduling domain. This design isolates interference between containers, reduces cross-cluster cache synchronizations, and eases contention for hardware resources like cache and NUMA memory. Performance is improved by more than 10% in high-concurrency Redis scenarios.



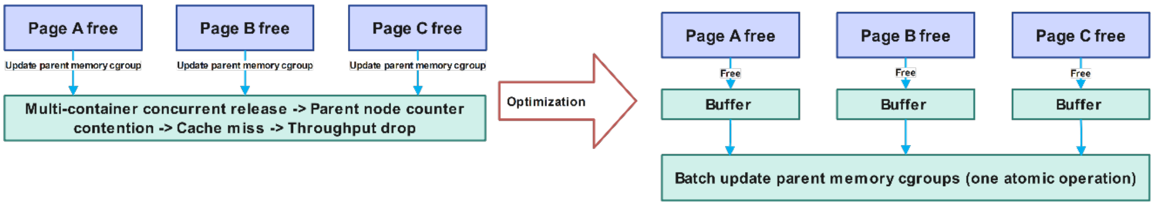
- Inter-cluster data sharing within a container: Services within a container operate on different CPU clusters. Cache synchronization between these clusters and latency of access across them lead to inconsistent service performance.
- Resource contention between containers: Tasks of multiple container services compete for resources like cache and memory bandwidth. As a result, services interfere with each other and resource usage is unbalanced.



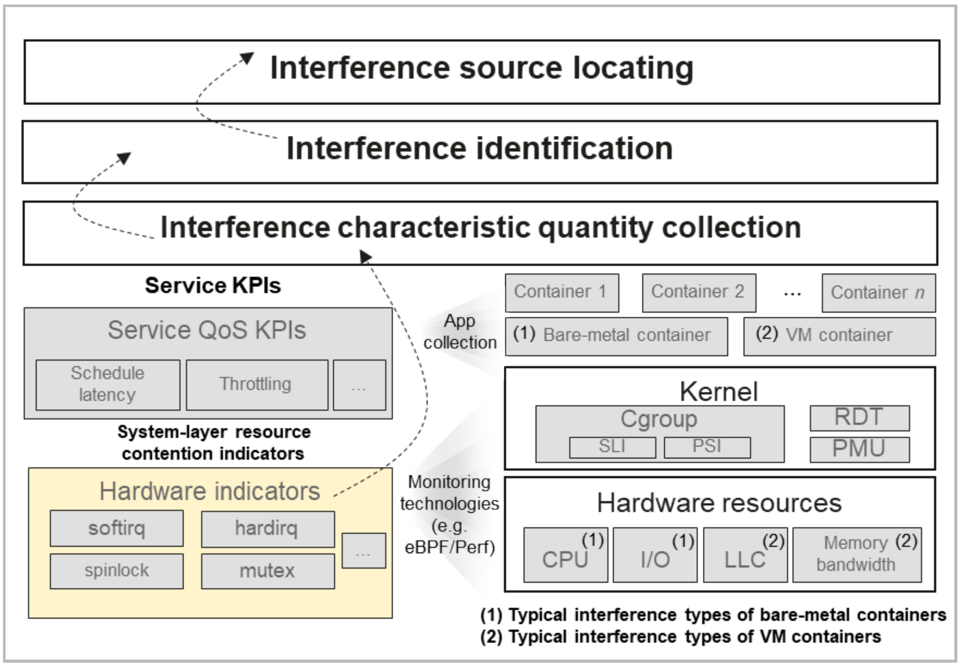
- Ext4 block allocation interference isolation: Scalability for Ext4 block allocation is improved by optimizing two key locks—group lock and s_md_lock. When one block group is busy, the system uses free ones instead. This reduces CPU waste from multiple containers competing for the same block group and makes better use of all available block groups, thereby easing group lock conflicts. Additionally, the global target of streaming allocation is divided into smaller inodes, reducing conflicts on the global lock s_md_lock and enabling more centralized organization of file data. In 64-container deployments, the OPS increases over fivefold in scenarios involving both block allocation and release and over tenfold for block allocation alone.



- Network TCP hash interference isolation: Lock contention occurs between tcp_hashinfo bash and ehash, leading to throughput drops and latency increases in high-concurrency situations due to frequent ehash calculation. To address this, the spinlock for tcp_hashinfo bash and ehash is changed to Read-Copy Update (RCU) and the ehash calculation method to incremental local port (lport). These changes minimize query time and calculation frequency, thereby decreasing lock contention for TCP connect hash.
- Enhanced control group (cgroup) isolation: Original atomic operations are replaced with percpu counters to avoid parent node contention across namespaces and eliminate rlimit count interference between containers. This mechanism addresses the linearity issue in the will-it-scale/signal1 test case and triples concurrent throughput in a 64-container deployment. Memory cgroups are released in batches to avoid contention for the same parent node's count caused by frequent small memory releases, enhancing memory count scalability. In the tlb-flush2 test case, this improves throughput by 1.5 times with 64 containers. Leveraging eBPF's programmable kernel capabilities, a host information isolation and filtering approach is provided to present container-specific resource views. Compared with the peer LXCFS solution, this openEuler solution avoids the overhead of switching between the kernel mode and user mode, and eliminates performance and reliability bottlenecks associated with the LXCFS process. It doubles the resource view throughput in a single container and achieves a 10-fold increase in a 64-container deployment.

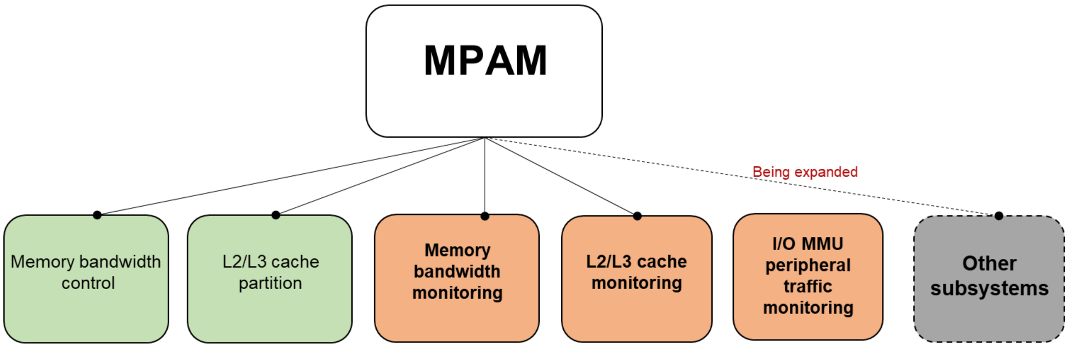

Page A free
Page B free
Page C free
Update parent memory cgroup
Update parent memory cgroup
Update parent memory cgroup
Multi-container concurrent release -> Parent node counter contention -> Cache miss -> Throughput drop
Optimization
Page A free
Page B free
Page C free
Free
Free
Free
Buffer
Buffer
Buffer
Batch update parent memory cgroups (one atomic operation)

- Interference monitoring: Interference falls into three categories by result: instruction execution failure, instruction execution slowdown, and increased instruction executions. Interference is monitored from the kernel perspective, with statistics collected on each typical category during runtime. The system supports online monitoring of schedule latency, throttling, softirq, hardirq, spinlock, mutual exclusion (mutex), and simultaneous multi-threading (SMT) interference while incurring less than 5% performance overhead.


Interference source locating
Interference identification
Interference characteristic quantity collection
Service KPIs
Service QoS KPIs
Schedule latency
Throttling
...
System-layer resource contention indicators
Hardware indicators
softirq
hardirq
spinlock
mutex
...
App collection
Monitoring technologies (e.g. eBPF/Perf)
Container 1
Container 2
...
Container n
(1) Bare-metal container
(2) VM container
Kernel
Cgroup
SLI
PSI
RDT
PMU
Hardware resources
CPU (1)
I/O (1)
LLC (2)
Memory bandwidth (2)
(1) Typical interference types of bare-metal containers
(2) Typical interference types of VM containers

- Memory system resource partitioning and monitoring (MPAM): MPAM controls memory and cache QoS by setting limits, guarantees, or priorities for memory bandwidth and cache usage. It applies different isolation policies per thread depending on the service type. Real-time monitoring tracks service-level and thread-level usage of shared resources and feeds data back to control policies for effective closed-loop management. MPAM also works with the system memory management unit (SMMU) to enhance I/O QoS for peripheral devices and heterogeneous accelerators, enabling isolated bandwidth configuration and device-level resource monitoring.

Figure 4-1 MPAM for cache and memory bandwidth monitoring



MPAM implements both static and dynamic control to improve resource utilization:

1. Static control limits the offline service bandwidth to keep online services unaffected. In real-world scenarios, offline services typically consume 5% to 20% of total system bandwidth under different workloads. MPAM controls the bandwidth of offline services with approximately 1% accuracy.
2. In dynamic mode, MPAM frees up bandwidth for offline services during idle periods of online services to prevent unnecessary resource waste.

Application Scenarios

In a many-core server, service containers are deployed at a high density to minimize interference between them, thereby enhancing resource utilization.

NestOS

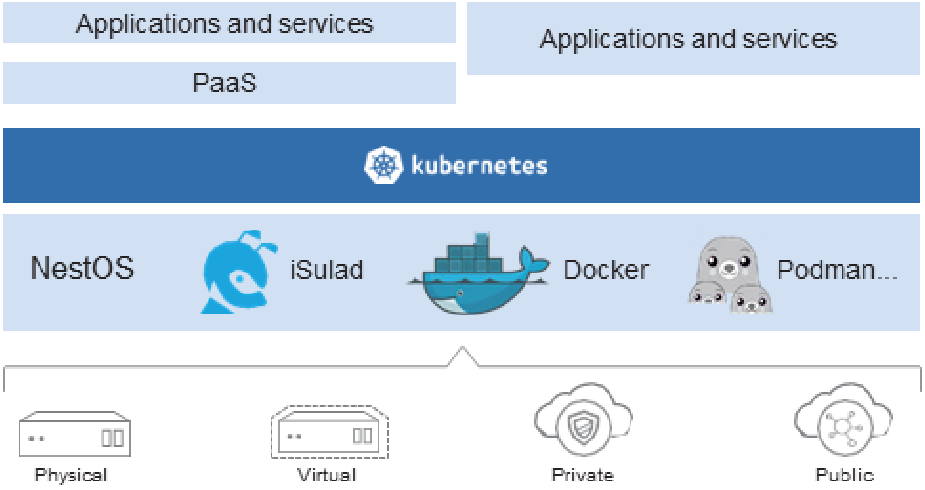
SIG	Cloud Native

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 platforms such as Kubernetes and OpenStack, reducing container overheads and providing extensive cluster components in large-scale containerized environments.

Challenges

Various runtimes and management software have been emerging as containers and Kubernetes are widely adopted in cloud native scenarios. Technologies such as container and orchestration further decouple service rollout and O&M from the underlying environment. Without a unified O&M stack, O&M platforms need to be built repeatedly.

Feature Description



- Out-of-the-box design: Integrates popular container engines like iSulad, Docker, and Podman to provide lightweight and tailored cloud OSs.
- Easy configuration: Uses the Ignition feature 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, 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 failovers to ensure integrity and security during system running.

Application Scenarios

NestOS can meet the demands of containerized cloud applications to solve problems such as inconsistent and repetitive O&M operations of stacks and platforms, which are typically caused by the decoupling of containers and underlying environments when using container and container orchestration technologies for rollout and O&M. NestOS ensures consistency between services and the OS.

Repositories

https://gitee.com/openeuler/NestOS

KubeOS

SIG	Cloud Native

KubeOS is a lightweight, secure container OS designed for cloud native scenarios to enable unified management of both containers and node OSs through Kubernetes, including atomic upgrades and API-based O&M.

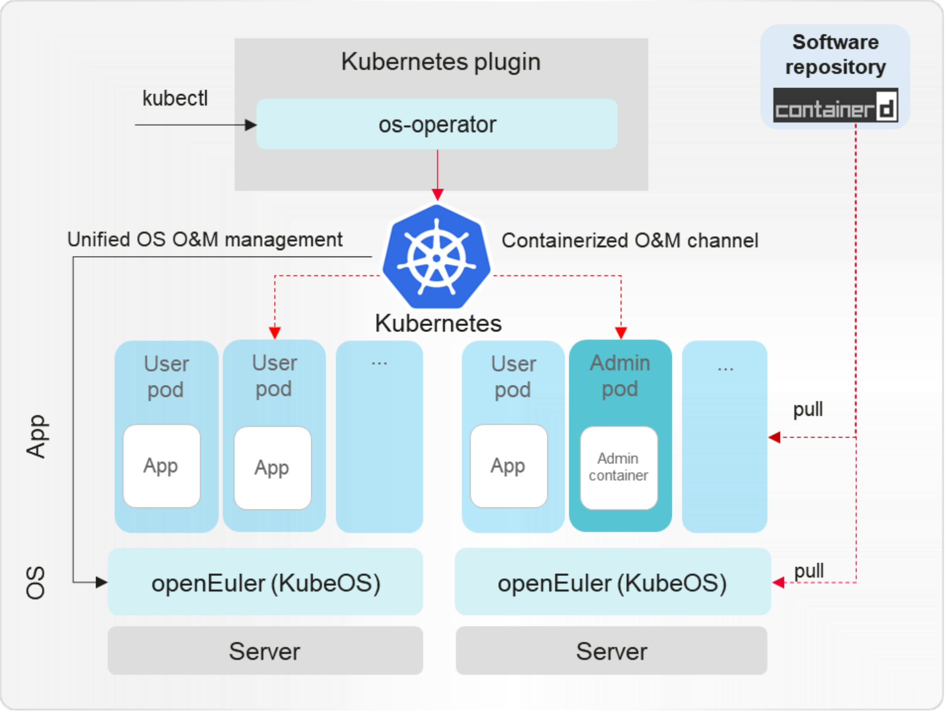
Challenges

Traditional OSs are designed for general-purpose workloads and work poorly in situations like single-purpose containers and Kubernetes orchestration in cloud native environments, leading to OS management complexity and O&M inefficiencies.

- Containerized applications lead to new challenges which traditional OSs are too heavy to address.
- Containers and OSs each use an O&M and management system, which results in management redundancy and inefficient scheduling between two systems.
- Siloed package management leads to inconsistency of container OS version in a cluster, which necessitates a unified container OS management mechanism.

Feature Description

KubeOS is a lightweight container OS developed from openEuler for cloud native scenarios. It works with Kubernetes to enable unified, atomic management of both containers and OSs.



KubeOS has the following features:

- KubeOS connects OSs to clusters as components and uses Kubernetes to implement unified management of node OSs and service containers.
- KubeOS detects cluster status before OSs change to implement collaborative scheduling of service containers and OSs.
- KubeOS uses Kubernetes-native declarative APIs to perform OS O&M and management in a standard manner.
- KubeOS supports atomic OS upgrade and rollback to ensure consistency across cluster nodes.

- KubeOS only entails components required for container running, which reduces the attack surface and vulnerabilities as well as OS overheads and reboot time. Additionally, the read-only rootfs protects the system from attacks and malicious tampering.

Application Scenarios

KubeOS is mainly used as a cloud-native infrastructure to provide a basic operating environment for cloud services and help cloud vendors and telecom customers with OS O&M in cloud-native scenarios.

Repositories

https://gitee.com/openeuler/KubeOS

Kmesh

SIG	eBPF

Kmesh is a high-performance service mesh data plane software. Based on a programmable kernel, Kmesh offloads traffic governance from proxies to the OS, which shortens the traffic path from multiple hops to one hop. This significantly improves application access performance in a service mesh.

Challenges

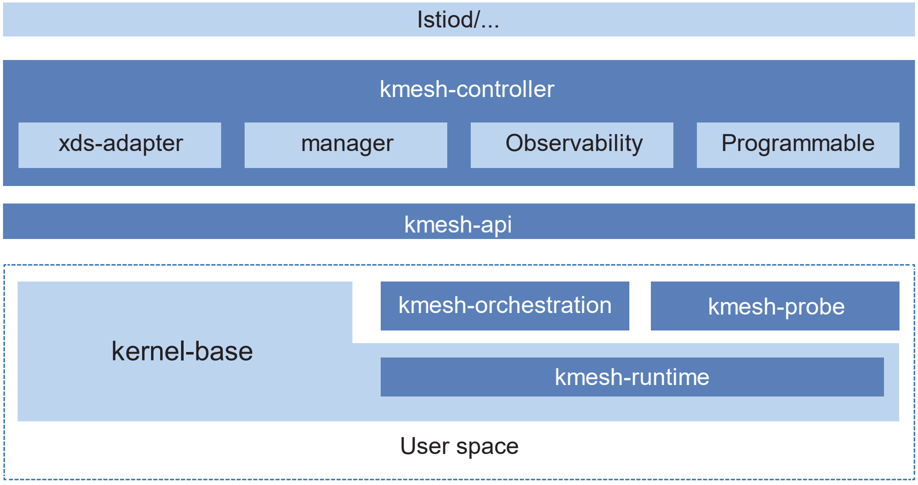
The boom of cloud-native applications places higher requirements on cloud infrastructure in terms of scalability and Service Level Agreement (SLA). Data centers expand to connect with more cluster services and manage soaring volumes of data, posing a big challenge to efficiently implement traffic governance between microservices.

Service mesh is a next-gen microservice technology that offloads traffic governance from services to the mesh infrastructure without affecting applications. However, service meshes use proxies, which introduces extra latency and overheads. In Istio, for example, the single-hop service access latency increases by 2 ms to 3 ms, making the software unable to meet the SLA requirements of latency-sensitive applications.

Feature Description

Kmesh works on a programmable kernel to offload traffic governance to the OS, improving performance on the service mesh data plane. Kmesh supports the following features:

1. Connects to a mesh control plane (such as Istio) that complies with the Dynamic Resource Discovery (xDS) protocol.
2. Orchestrates traffic in three ways:
   (1) Load balancing: Various policies such as polling.
   (2) Routing: L7.
   (3) Gray: Backend service policies available in percentage mode.



As shown in the figure, the Kmesh architecture consists of the following components:

- kmesh-controller: Kmesh management program, which is responsible for Kmesh lifecycle management, xDS protocol interconnection, and O&M monitoring.
- kmesh-api: API layer provided by Kmesh for external systems, including orchestration APIs converted by xDS and O&M monitoring channels.
- kmesh-orchestration: L3 to L7 traffic orchestration implemented based on eBPF, such as routing, gray, and load balancing.
- kmesh-probe: O&M monitoring probe, which provides end-to-end monitoring.
- kmesh-runtime: runtime that supports L3 to L7 traffic orchestration implemented in the kernel.

Application Scenarios

Kmesh can serve latency-sensitive applications such as e-commerce, cloud gaming, online conferencing, and short videos. Kmesh brings a 5-fold forwarding performance increase in HTTP tests, compared to Istio.

Repositories

https://gitee.com/openeuler/Kmesh

iSulad

SIG	iSulad

iSulad is a C/C++ lightweight container engine not restricted by hardware architectures or specifications, featuring low overhead and wide applicability.

Challenges

A container is an isolated environment that streamlines application packaging and distribution. Compared with virtualization, containers accelerate distribution and reduce overhead, effectively improving development and deployment efficiencies. As the Docker container engine, Kubernetes container orchestration and scheduling, and cloud-native deployments become more widespread, the container ecosystem is developing rapidly.

Container technology faces an increasing number of user requirements, including:

- Containers need to be deployed and started quickly.
- Resources consumed by containers must be limited to a reasonable range.
- Containers should adapt to Internet of Things (IoT) and edge computing scenarios.

Based on these user requirements, the iSulad container solution is proposed by openEuler as a lightweight and fast container engine.

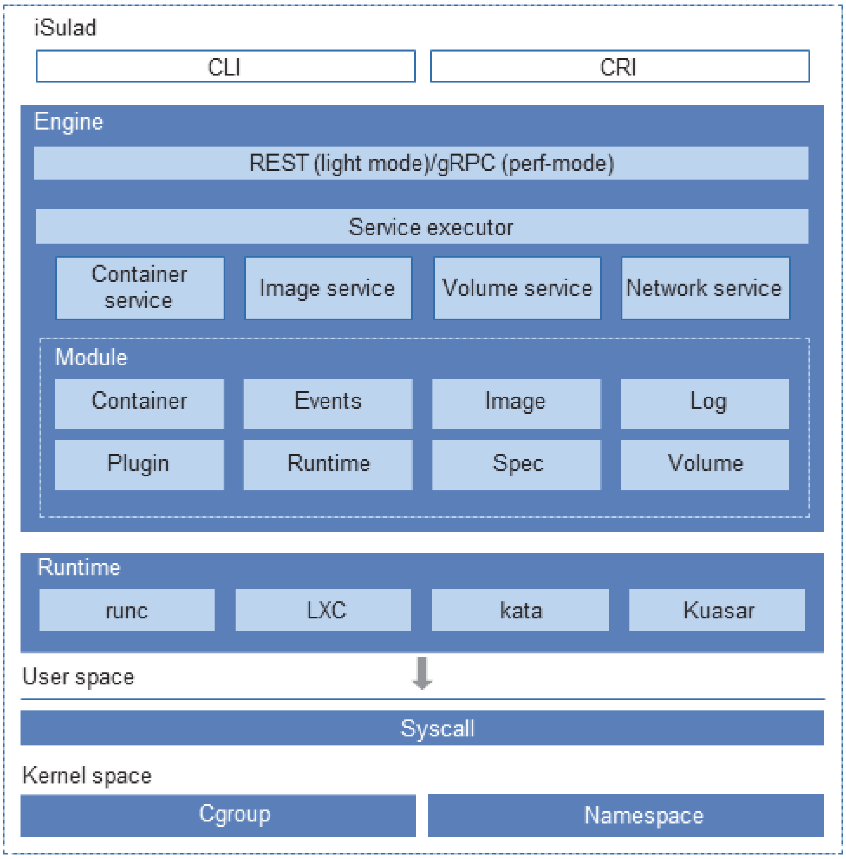
Feature Description

iSulad features a unified architecture tailored for ICT. Compared with a Go Docker container engine, iSulad is more resource efficient, starts containers faster, and can be used in a wider range.

Named after the small isula ant, which is a small but powerful insect, iSulad is a flexible, stable, and secure container base for various applications.

iSulad provides commands similar to those of Docker, for greater usability. It supports the northbound Container Runtime Interface (CRI) and can connect to Kubernetes. iSulad as the container base can be used to orchestrate and schedule containers through Kubernetes. It also supports the southbound Open Container Initiative (OCI) to work with diverse container runtime environments, such as runc, LXC, kata, and Kuasar.

Figure 4-2 iSulad software architecture



Core capabilities of iSulad include container service, image service, volume service, and network service. The container service manages the lifecycle of containers, while the image service enables operations on container images. iSulad complies with the OCI Image Specification and supports mainstream image formats in the industry. In addition, iSulad supports the external rootfs image format of system containers and the embedded image format. The volume service manages data volumes of a container, and the network service works together with network plugins compliant with Container Network Interface (CNI) to provide network capabilities for containers.

As a general-purpose container engine, iSulad supports system containers and secure containers as well as common containers.

- Common containers: They are traditional application containers.
- System containers: They have extended functions based on common containers, including the systemd service capability as well as dynamic addition or release of drives, NICs, routes, and volumes during container runtime. System containers are mainly used for computing-intensive, high-performance, and heavy-concurrency applications and cloud services.
- Secure containers: They are a combination of virtualization and container technologies. Unlike common containers that share the host kernel, secure containers clearly isolate containers through the virtualization layer. Each secure container has its own kernel and a lightweight VM environment, ensuring that different containers on the same host do not affect each other.

Compared with Docker, iSulad features faster container startup and lower resource overhead, because iSulad is developed using C/C++. iSulad optimizes the call chain at the code layer. iSulad calls functions directly through the link library, whereas Docker calls fork and exec functions on binary files for multiple times. The shorter call length enables iSulad to start containers faster. What's more, usage of C language allows iSulad to fully play its role on embedded and edge devices, which allows for a wider range than Docker.

According to tests, iSulad brings only 30% of the Docker-incurred memory overhead, and in Arm and x86 environments, iSulad can start 100 containers concurrently in less than half of the time Docker takes. These advantages enable iSulad to start up containers faster and reduce resource consumption, minimizing the impact on containerized applications.

Application Scenarios

iSulad has been widely used in cloud computing, CT, embedded, and edge scenarios like banking, finance, communication, and cloud. iSulad is best for a more effective full-stack secure container solution when combined with Kuasar and StratoVirt thanks to the openEuler community.

Repositories

https://gitee.com/openeuler/iSulad

StratoVirt

SIG	Virt

StratoVirt is an enterprise-class virtualization platform designed for cloud data centers. It offers a unified architecture that fits into three scenarios: VMs, containers, and serverless computing. StratoVirt is lightweight and causes low memory overhead, supports software and hardware collaboration, and is safe at the Rust language level.

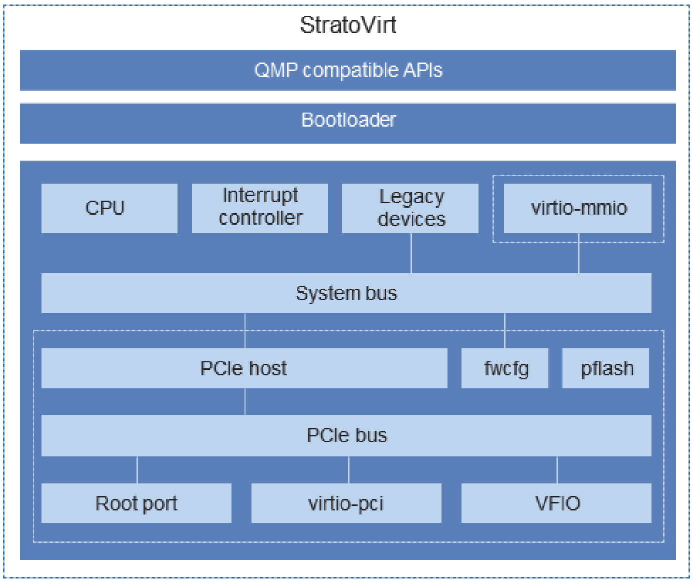
Challenges

As the QEMU virtualization software keeps evolving, its core open source components generate increasingly large volumes of code, among which a large amount of code is outdated. In recent years, CVE security vulnerabilities frequently occur, highlighting problems such as poor security, code redundancy, and low efficiency. As a result, the rust-vmm architecture developed using the memory-safe programming language Rust has become a more practical solution. General-purpose virtualization technologies for all scenarios (data centers, terminals, and edge devices) are the future trend, due to their security, light weight, and performance advantages. StratoVirt is one such technology designed for openEuler.

Feature Description

StratoVirt is an open source, lightweight virtualization technology powered by Linux Kernel-based Virtual Machine (KVM). It reduces memory consumption and accelerates VM startup while maintaining the isolation and security capabilities of traditional virtualization technologies. StratoVirt can be applied in serverless scenarios such as microservices or function computing, and retains virtualization interfaces and designs for quickly importing more features to supplement general-purpose virtualization.

Figure 4-3 StratoVirt software architecture



The core architecture of StratoVirt is divided into three layers from top to bottom:

1. External APIs: StratoVirt uses the QEMU Machine Protocol (QMP) to communicate with external systems, is compatible with OCI, and supports interconnection with libvirt.
2. Bootloader: In lightweight scenarios, StratoVirt uses a simple bootloader to load kernel images, much faster than the traditional BIOS+Grub boot method. In general-purpose virtualization scenarios, StratoVirt supports UEFI boot.
3. Emulated mainboard – microVM: To improve performance and reduce the attack surface, StratoVirt minimizes the emulation of user-mode devices. With the emulation capability, KVM-based devices and paravirtualization devices are available, such as generic interrupt controller (GIC), serial, real-time clock (RTC), and virtio-mmio devices.

4. General-purpose VMs: StratoVirt provides an advanced configuration and power interface (ACPI) table to implement UEFI boot. Virtio-pci and Virtual Function I/O (VFIO) passthrough devices can be added to further improve the VM I/O performance.

Application Scenarios

StratoVirt, iSula, and Kubernetes combine to form a complete container solution, which processes serverless loads efficiently.

Repositories

Kuasar

SIG	Cloud Native

Kuasar is a container runtime that supports unified management of multiple types of sandboxes. It supports multiple mainstream sandbox isolation technologies. Based on the Kuasar container runtime combined with the iSulad container engine and StratoVirt virtualization engine, openEuler builds lightweight full-stack self-developed secure containers for cloud native scenarios, delivering key competitiveness of ultra-low overhead and ultra-fast startup.

Challenges

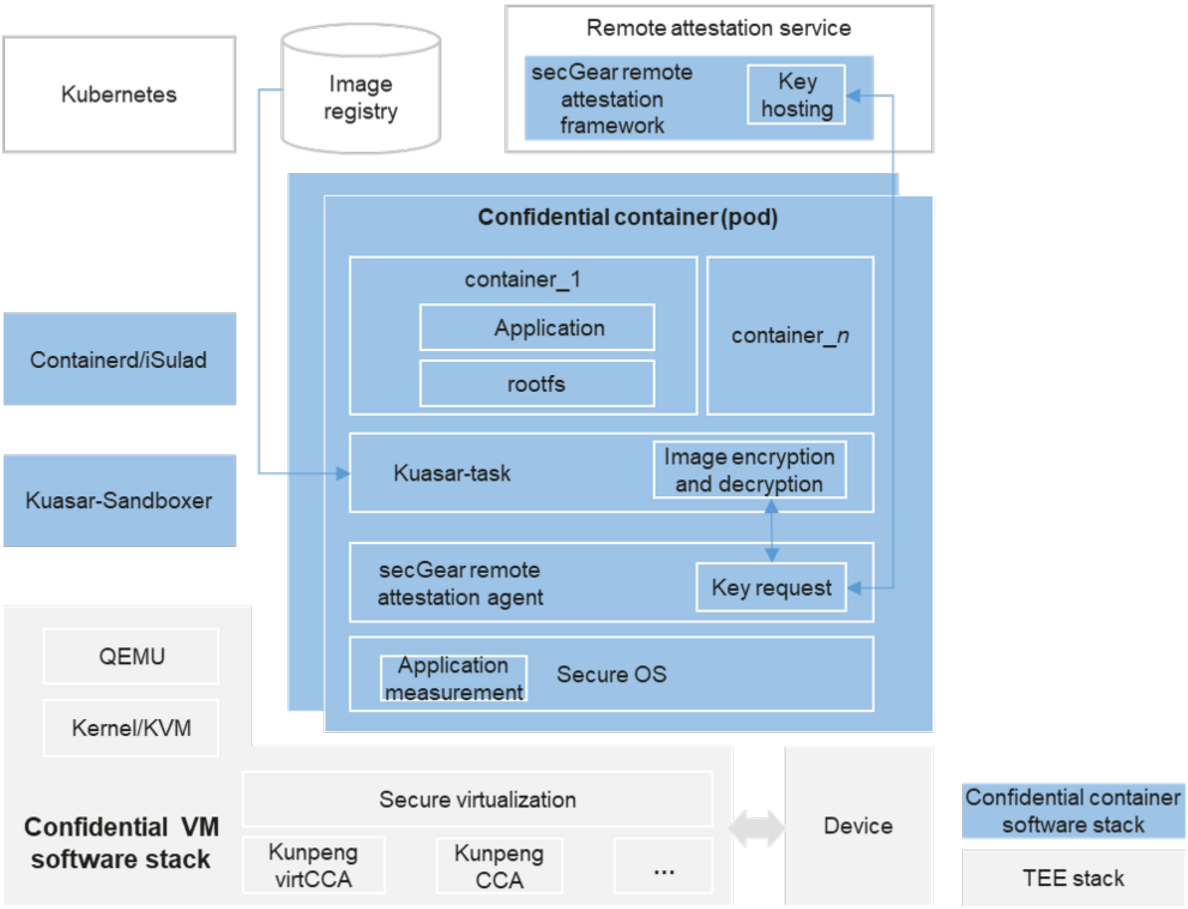
With the widespread adoption of cloud native technologies, migrating cloud native applications to confidential computing environments is essential to protect the confidentiality and integrity of containers and container data, while maintaining the development and deployment experience consistent with that of common containers.

Based on these user requirements, openEuler proposes Kuasar as a container runtime that supports unified management of multiple types of sandboxes.

Feature Description

Kuasar fully utilizes the advantages of the Sandboxer architecture to deliver a high-performance, low-overhead confidential container runtime.

- Native integration with the iSulad container engine preserves Kubernetes ecosystem compatibility.
- Hardware-level protection via virtCCA technology ensures confidential workloads are deployed in trusted environments.
- The secGear remote attestation framework, which complies with the remote attestation procedures (RATS) (RFC9334), allows containers running in a confidential computing environment to prove their trustworthiness to external trusted services.
- Container images can be pulled and decrypted in confidential containers to protect their confidentiality and integrity.



Application Scenarios

Kuasar addresses customer data security needs while seamlessly integrating with cloud native ecosystems. It benefits confidential applications from cloud native advantages including high availability, auto scaling, and rapid delivery. The solution finds broad application in confidential computing scenarios spanning AI security, trusted data circulation, and privacy protection.

Repositories

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

Embedded

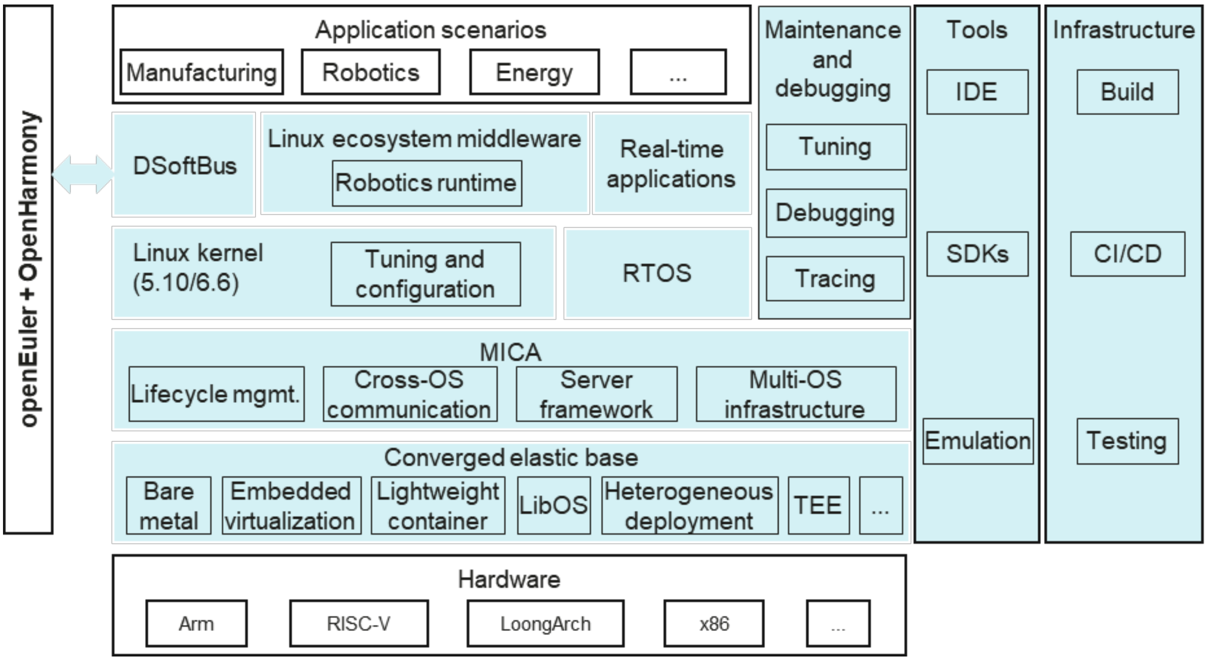
openEuler is suited for embedded applications, offering significant progress in southbound and northbound ecosystems, technical features, infrastructure, and implementation over previous generations.

openEuler Embedded helps build operational technology (OT) applications such as manufacturing and robotics in a closed-loop design, whereby innovations help optimize its embedded system software stack and ecosystem. openEuler Embedded enhances its software package ecosystem by incorporating the oebridge feature, which supports online software installation from an openEuler mirror site. When building Yocto images, oebridge can be used to install openEuler RPM packages for easy image customization. openEuler Embedded also supports the oedeploy feature for quick deployment of AI and cloud-native software stacks.

Kernel support in openEuler is enhanced by optimizing the meta-openeuler kernel configuration and the oeaware real-time tuning feature. These updates help control interference and improve real-time system responsiveness.

Looking ahead, through collaboration with community partners, users, and developers, openEuler Embedded will expand support for new processor architectures like LoongArch, enhance southbound hardware compatibility, and advance key capabilities including industrial middleware, embedded AI, edge computing, and simulation systems, to establish a comprehensive embedded software platform solution.

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, Rockchip, Renesas, TI, Phytium, StarFive, and Allwinner.

Embedded Elastic 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.

1. 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.
2. 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.

3. 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 mixed-criticality (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 makes the universal Linux OS and a dedicated real-time operating system (RTOS) combine 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, while 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:

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

Northbound Ecosystem

1. Northbound software packages: More than 600 common embedded software packages can be built using openEuler.
2. Soft real-time kernel: This capability helps respond to soft real-time interrupts within microseconds.
3. 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.
4. 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 the 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.

Edge Computing

KubeEdge

SIG	Edge

openEuler Edge integrates KubeEdge into a unified management platform, on which edge and cloud applications can be provisioned. It delivers collaboration across edge and cloud to help streamline AI deployments, implement service discovery and traffic forwarding, and improve southbound capabilities.

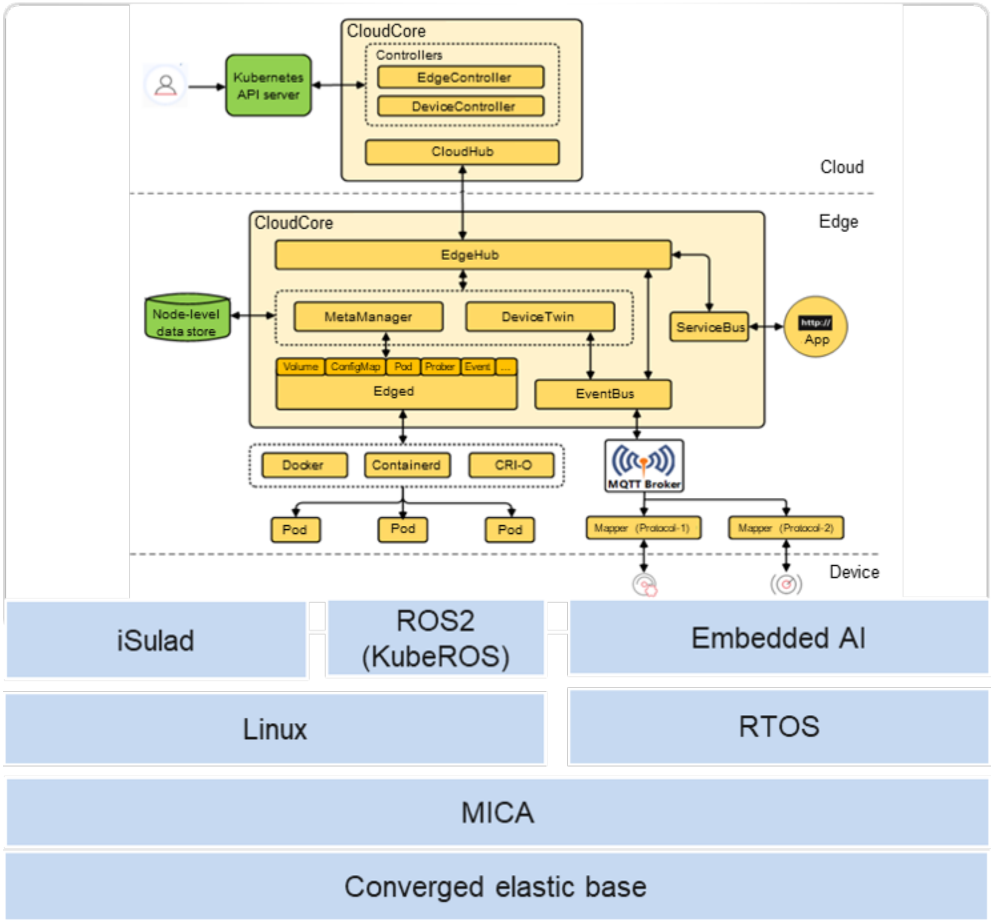
Challenges

Edge computing is one of the top 10 major technology trends, and as such, is dominating current and future business models. Smart city, autonomous driving, and industrial Internet applications are generating massive data volumes that cannot be processed by centralized cloud computing. There is a huge demand for high-speed, low-latency, and cost-efficient edge computing solutions to adapt to frequent data exchanges.

Feature Description

KubeEdge provides the following features:

1. Edge-cloud management collaboration: KubeEdge helps manage and provision edge and cloud applications, implement edge-cloud communication, and manage southbound peripherals across edge and cloud.
2. Edge-cloud service collaboration: EdgeMesh Agent and EdgeMesh Server are deployed on the edge and the cloud, respectively, to improve service discovery and routing.
3. Optimized southbound edge services: Device Mapper is used for southbound access to provide the peripheral profile and parsing mechanisms, helping manage and control southbound peripherals and service streams. The southbound edge services are compatible with the EdgeX Foundry open source ecosystem.
4. Edge data services: The edge data services can implement on-demand persistence of messages, data, and media streams, and provide data analysis and data export capabilities.
5. Collaborative innovation: iSulad interconnects with MICA to achieve elastic base updates and support for RTOS container images, enabling collaborative innovation between KubeEdge and the MICA system.
6. Extensive applications: KubeEdge, KubeROS, embedded AI deployment, and DSoftBus are combined.



Key benefits:

- Reduced cloud-edge bandwidth usage and latency: A self-closed-loop service logic at the edge helps reduce edge and cloud network resource usage, improve the response speed, and protect customer data privacy.
- Cloud-based AI application management and deployment: A variety of applications such as machine learning and image recognition applications can be easily deployed at the edge.

Application Scenarios

In cloud-edge collaboration scenarios such as remote intelligent monitoring and industrial IoT, the native containerized application orchestration function is extended to edge nodes based on KubeEdge, making it easy to manage and deploy edge applications from the cloud.

Repositories

https://mirror.sjtu.edu.cn/openeuler/openEuler-23.03/edge_img/x86_64/openEuler-23.03-edge-x86_64-dvd.iso

https://github.com/kubeedge/kubeedge

DSoftBus

SIG	Distributed middleware

To develop a digital infrastructure OS and improve collaboration between devices and edges, openEuler marks an industry feat by applying DSoftBus technology to the server, edge, and embedded. DSoftBus provides a unified platform to enable collaboration and communication for distributed devices, achieving fast device discovery and efficient data transmission.

Challenges

Collaboration between edge devices includes the discovery, connection, networking, and transmission phases, though interconnection between edge devices has the following difficulties:

1. Different types of devices: There is no unified solution to cover various hardware capabilities and supported connection modes, such as Wi-Fi, Bluetooth, and near field communication (NFC).
2. Unstable and slow networking: It is difficult to automatically create and allocate network management roles between edge devices and maintain network stability after devices exit, power off, or experience a fault.
3. Poor transmission: Data transmission performance cannot be guaranteed between edge devices, especially for devices that have certain restrictions on power consumption.
4. Difficult API adaptation: There are no unified APIs that mask differences in underlying hardware and networking, to allow one service process to be reused by different edge devices.

Feature Description

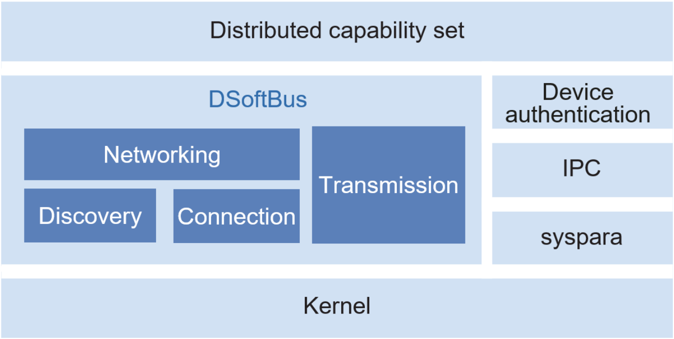
Key Features

1. Device discovery and connection based on Wi-Fi, wired network, and Bluetooth
2. Unified networking and topology of device and transmission information
3. Transmission channel for data bytes, streams, and files

Architecture

DSoftBus consists of four basic modules: discovery, networking, connection, and transmission.

Figure 4-4 Relationship between DSoftBus and external modules



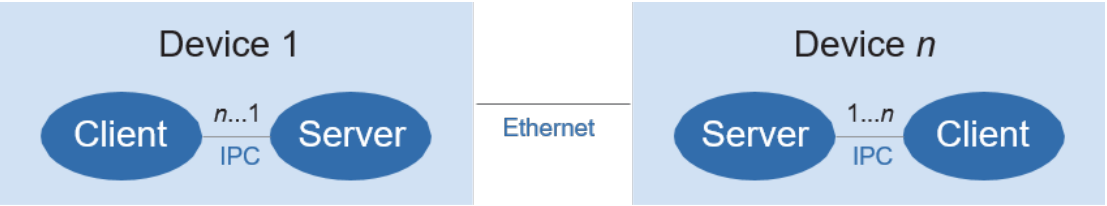
For southbound devices, DSoftBus supports Wi-Fi, wired network, and Bluetooth connection, whereas for northbound distributed applications, it provides unified APIs that mask the underlying communication mechanism.

DSoftBus depends on peripheral modules such as device authentication, inter-process communication (IPC), log, and system parameters (SNs). In embedded environments, such modules can be tailored or replaced to provide basic functions of DSoftBus, meet actual service scenarios, and extend DSoftBus capabilities.

Deployment

1. DSoftBus can be deployed on multiple devices in a LAN, but the devices communicate with each other through the Ethernet.
2. A device consists of a server and a client that communicate through the IPC module.
3. Multiple clients can concurrently access a server on a device.

Figure 4-5 Deployment



DSoftBus provides services for external systems through an independent process by executing the main program of the server.

Application Scenarios

DSoftBus is best suited for device discovery and interconnection between openEuler edge servers and common and OpenHarmony embedded devices.

It provides unified APIs and protocol standards to enable self-connection, self-networking, and plug-and-play of multi-vendor, multi-type devices, implementing the peripheral access needed in industrial production lines and campus management.

Repositories

https://gitee.com/openeuler/dsoftbus_standard

5 Basic Capability Innovations

Kernel Innovations

SIG	Kernel

openEuler runs on Linux kernel and inherits the competitive advantages of community versions and innovative features released in the openEuler community, including but not limited to the following.

- Filesystem in USErspace (FUSE) pass-through: FUSE is widely used in distributed storage and AI applications. In pass-through scenarios, FUSE skips additional processing for read and write I/Os. It only records metadata and forwards the I/O requests to the back-end file system. As a result, FUSE processing turns into the main bottleneck for I/O performance. The FUSE pass-through feature is designed to eliminate the overhead caused by context switches, wakeups, and data copying on the data plane when FUSE directly interfaces with the back-end file system. It allows applications to directly send read and write I/Os to the back-end file system within the kernel. In lab environments, FUSE pass-through has demonstrated notable performance gains. Specifically, fio tests show that read and write performance more than doubles for sizes between 4 KB and 1 MB. FUSE pass-through has also passed fault injection and stability tests, and is available for use as needed.
- Enhanced Memory System Resource Partitioning and Monitoring (MPAM) features

  (1) An improved QoS feature is introduced to optimize memory bandwidth and L3 cache control. In hybrid deployment scenarios, shared resources can be allocated based on the upper limit, lower limit, or priority-based policy.

  (2) The new I/O QoS management feature collaborates with the system memory management unit (SMMU) to isolate I/O bandwidth traffic across hardware peripherals and heterogeneous accelerators. It supports monitoring by iommu_group, providing a new approach to I/O QoS management in heterogeneous environments.

  (3) The L2 cache isolation feature enables monitoring of L2C usage and bandwidth traffic, offering core-level optimization and performance analysis in hybrid deployment scenarios.

  These MPAM features deliver significant performance improvements in test scenarios. In hybrid deployments, the interference rate of SPECjbb as an online service drops from 25.5% to below 5%.
- 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.

- Folio-based memory management: Folio-based Linux memory management is used instead of page. A folio consists of one or more pages and is declared in struct folio. Folio-based memory management is performed on one or more complete pages, rather than on PAGE_SIZE bytes. This alleviates compound page conversion and tail page misoperations, while decreasing the number of least recently used (LRU) linked lists and optimizing memory reclamation. It allocates more continuous memory on a per-operation basis to reduce the number of page faults and mitigate memory fragmentation. Folio-based management accelerates large I/Os and improves throughput, and large folios consisting of anonymous pages or file pages are available. For AArch64 systems, a contiguous bit (16 contiguous page table entries are cached in a single entry within a translation lookaside buffer, or TLB) is provided to reduce system TLB misses and improve system performance. In openEuler 24.03 LTS SP1, multi-size transparent hugepage (mTHP) allocation by anonymous shmem and mTHP lazyfreeing are available. The memory subsystem supports large folios, with a new sysfs control interface for allocating mTHPs by page cache and a system-level switch for feature toggling.
- Large folio for ext4 file systems: The IOzone performance can be improved by 80%, and the writeback process of the iomap framework supports batch block mapping. Blocks can be requested in batches in default ext4, optimizing ext4 performance in various benchmarks. For ext4 buffer I/O and page cache writeback operations, the buffer_head framework is replaced with the iomap framework that adds large folio support for ext4. In version 24.03 LTS SP1, the performance of small buffered I/Os (≤ 4 KB) is optimized when the block size is smaller than the folio size, typically seeing a 20% performance increase.
- xcall and xint: The Linux kernel is becoming increasingly complex, and system calls, especially the simple ones, can be a performance bottleneck. System calls on the AArch64 platform share the same exception entry point, which includes redundant processes such as security checks. Common ways to reduce system call overhead include service offloading and batch processing, but both require service adaptation. xcall provides a solution that does not require service code modification. It streamlines system calls by optimizing their processing logic, trading off some maintenance and security capabilities to reduce overhead.

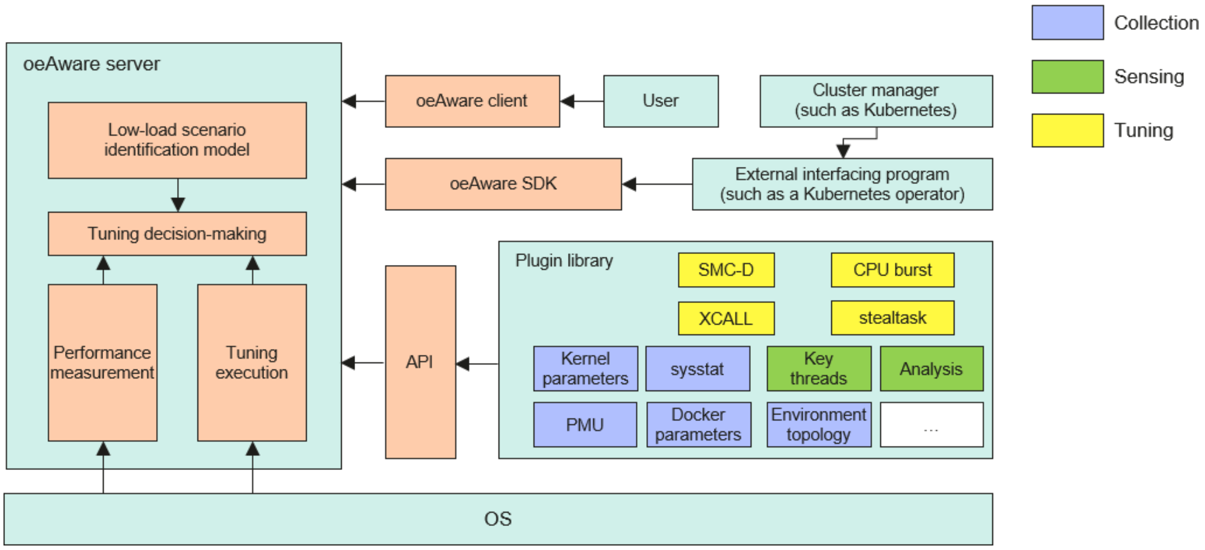
  To unify interrupt handling, the kernel integrates all interrupt handling processes into its general interrupt handling framework. As the kernel evolves, the general interrupt handling framework has gradually accumulated many security hardening and maintenance features that are not closely related to interrupt handling, increasing latency unpredictability. xint simplifies interrupt handling to reduce the latency and system overhead.

Efficient Concurrency and Ultimate Performance

oeAware

SIG	A-Tune

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 features that run independently and are statically enabled or disabled.



Challenges

Traditional tuning features are scattered, and are developed and promoted separately. This makes it difficult to gather all tuning capabilities. In addition, some tuning processes are isolated, wasting development resources. In contrast, oeAware supports intelligent openEuler tuning based on existing inputs, which will grow over time, preventing repeated development and allowing users to focus on modeling and tuning.

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.

- 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-node 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 examines system NUMA and network information during service operations, suggesting optimal tuning methods.
- The system tuning plugins comprise stealtask for enhanced CPU tuning, smc_tune (SMC-D) which leverages shared memory communication in the kernel space to boost network throughput and reduce latency, xcall_tune (XCALL) which bypasses non-essential code paths to minimize system call processing overhead, seep_tune which adjusts the core frequency in real time to reduce power consumption, and dynamic_smt_tune which schedules SMT queues during low-load periods to reduce interference between SMT0 and SMT1 and consequently alleviate interference among applications.

- The Docker tuning plugin addresses CPU performance issues during sudden load spikes by utilizing the CPU burst feature.
- The NUMA tuning plugin supports automatic core binding and memory migration, overcoming cross-NUMA memory access bottlenecks.
- The NIC tuning plugin monitors the affinity between network traffic and CPUs in real time and binds NIC queue interrupts in real time to improve the performance of network I/O-intensive applications.
- Transparent hugepage (THP) tuning applies to applications that experience frequent last-level cache (LLC) misses.
- sysBoost reduces indirect instruction jumps by combining and optimizing ELF files.
- Computing power coordination temporarily allocates idle CPU resources from a host to container instances in need.

Constraints

- smc_tune: SMC acceleration must be enabled before the server-client connection is established. This plugin is most effective in scenarios with numerous persistent connections.
- Docker tuning: This plugin 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 instance 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_tune excels in environments demanding high throughput and low latency, including high-performance computing (HPC), big data processing, and cloud platforms. By leveraging direct memory access (DMA), smc_tune 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.

Repositories

https://gitee.com/openeuler/oeAware-manager

A-Tune

SIG	A-Tune

A-Tune is an AI-powered OS performance tuning engine. It uses AI technologies to enable the OS to learn services statuses, simplify IT system tuning, and deliver excellent performance.

Challenges

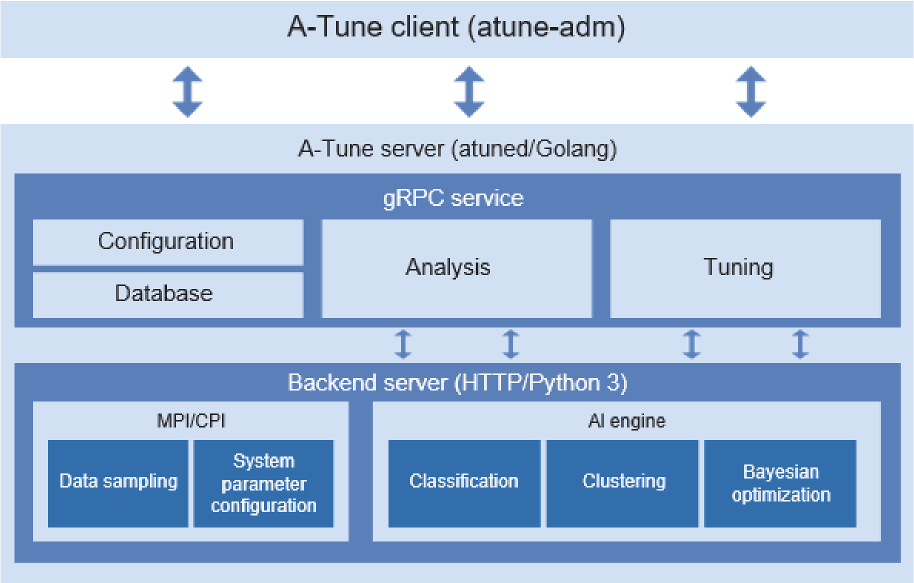
The development of hardware and software applications over the past few decades has coincided with a larger, more comprehensive Linux kernel. In openEuler, the **sysctl** command is used to configure kernel parameters and has over 1,000 parameters (**sysctl -a | wc -l**). A typical IT system covers the CPU, accelerator, NIC, compiler, OS, middleware framework, and upper-layer applications, and uses 7,000 parameters, most of which are default settings, which cannot tap into the full system performance. Parameter tuning faces the following difficulties:

1. A large number of parameters depend on each other.
2. Various types of upper-layer application systems require varying tuning of parameters.
3. Complex and diversified loads require the parameters to vary accordingly.

Feature Description

A-Tune is in a client/server (C/S) architecture. The atune-adm on the client is a command-line tool that communicates with the atuned process on the server through the gRPC protocol. The atuned process contains a frontend gRPC service layer (implemented by Golang) and a backend service layer, the former of which manages optimization configurations and data and provides tuning services for external systems, including intelligent decision-making (Analysis) and automated tuning (Tuning). By contrast, the backend service layer is an HTTP service layer executed by Python that consists of the Model Plugin Interface (MPI)/Configurator Plugin Interface (CPI) and AI engine. The MPI/CPI discovers system configurations, and the AI engine provides machine learning capabilities for the upper layer, including classification and clustering for model identification and Bayesian optimization for parameter search.

Figure 5-1 A-Tune software architecture



A-Tune delivers intelligent decision-making and automated tuning.

Intelligent decision-making is to sample system data and identify the corresponding loads using the clustering and classification algorithms of the AI engine, obtain the types of service loads, extract optimization configurations from the database, and select the optimal parameter values to fit each service load.

1. Automatically selects key features and removes redundant ones for precise user profiling.
2. Two-layer classification model accurately identifies service loads using the classification algorithm.

3. Load awareness proactively identifies application load changes and implements adaptive tuning.

Automated tuning uses configuration parameters and performance metrics of the system or application as reference, to repeatedly perform iteration using the parameter search algorithm of the AI engine. This can obtain parameter configurations that deliver optimal performance.

1. Automatically tunes core parameters to reduce search space and improve training efficiency.
2. Allows users to select the optimal tuning algorithm based on the application scenario, parameter type, and performance requirements.
3. Adds load features and optimal parameters to the knowledge base to improve future tuning operations.

Application Scenarios

A-Tune is widely used in Linux environments that handle big data, databases, middleware, and HPC workloads. Commonly used in industries such as finance and telecom, the software improves performance of applications such as MySQL, Redis, and BES middleware by 12% to 140%, respectively.

Repositories

https://gitee.com/openeuler/A-Tune

Gazelle

SIG	High-performance-network

Gazelle is a high-performance user-mode protocol stack. It directly reads and writes NIC packets in user mode based on DPDK and transmits the packets through shared hugepage memory, and uses the lwIP protocol stack. Gazelle greatly improves the network I/O throughput of applications and accelerates the network for the databases such as MySQL and Redis.

Challenges

1. **Design principles for an ideal protocol stack**

   The protocol stack should prevent packet copy, context switching, and cache miss during packet transmission between apps and NICs. It must fully utilize hardware scale-out capabilities, such as NIC multi-queue and CPU multicore, while preventing data access contention between cores. In addition, it should be applicable to diverse network thread models of applications.

2. **Features of an ideal protocol stack**

   To enhance performance, the protocol stack should enable zero copy, eliminate context switching, and prevent cache miss. In terms of linearity, it should feature distributed multicore deployment, lock-free, and independent protocol stack context. It should also be compatible with various application network models, provide a unified abstraction layer, and prevent protocols or hardware from complicating applications.

3. **Current situation**

   The kernel protocol stack focuses on universal compatibility while gradually implementing acceleration technologies. However, its performance remains consistently limited.

The user-mode protocol stack prioritizes performance at the cost of universality and adaptability.

Feature Description

1. **Features**

- High performance:

  Zero copy: DPDK-based packet passthrough

  Lock-free protocol stack: Based on lwIP, supporting linear scale-out

  Adaptive scheduling: Balanced traffic scheduling, maximizing computing power

  Dynamic core binding: Enabling application/protocol stack threads to run on adjacent cores

  Hardware offload: Improving protocol stack processing performance by offloading functions such as CSUM

- Universality:

  POSIX compatibility

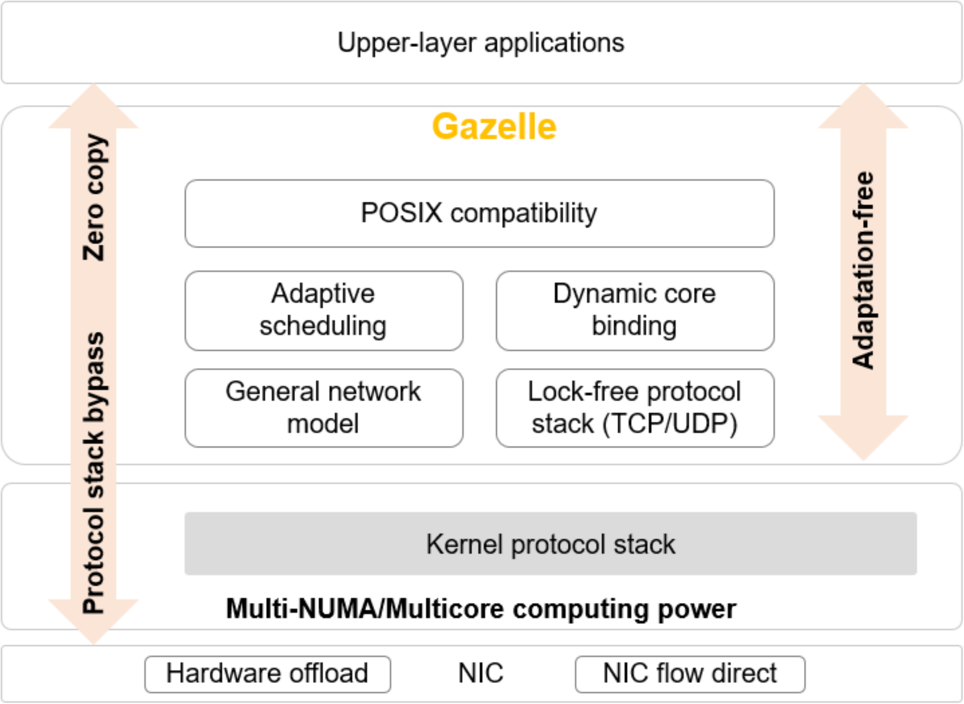
  General network model

  Compatible kernel protocol stack

- Adaptation-free:

  Applicable to applications without any modification

Figure 5-2 Gazelle architecture

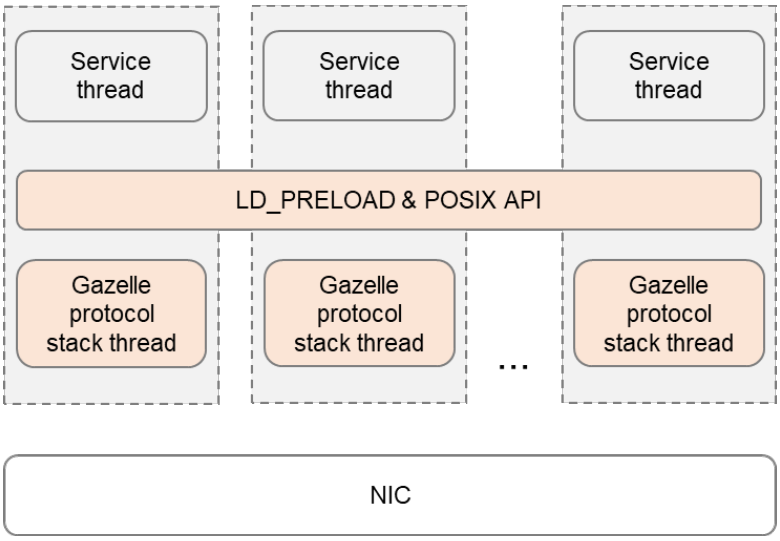


2. **RTC mode**

The number of service network threads is small and fixed. If the service thread FD is not shared across threads and no blocking call is made, Gazelle runs in RTC mode (co-thread mode). In this case, services and the protocol stack run in the same context. When the services execute poll/epoll, the protocol stack performs polling and packet receiving. The

CPU is exclusively used, and no wakeup scheduling is required, achieving ultimate performance.

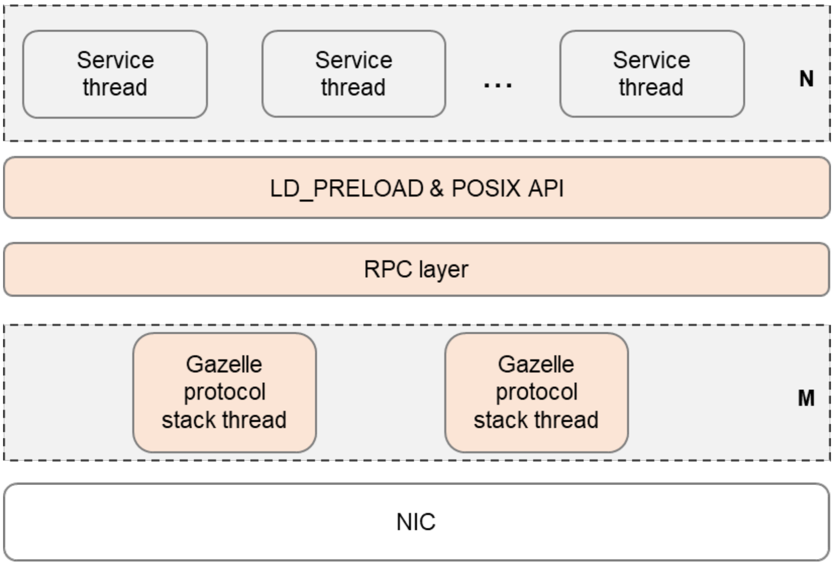
Figure 5-3 Thread model in RTC mode


Service thread
Service thread
Service thread
LD_PRELOAD & POSIX API
Gazelle protocol stack thread
Gazelle protocol stack thread
...
Gazelle protocol stack thread
NIC

3. **RTW mode**

When there are a large number of service threads, FD is shared across threads and Gazelle runs in RTW mode (independent thread mode). This mode separates the protocol stack thread from the service thread. Upon receiving a request, the protocol stack triggers and wakes up the corresponding service thread, which is similar to the software interrupt of the Linux kernel protocol stack. The service thread reads and writes packets through lock-free queues, preventing lock contention with the protocol stack. Other control-plane socket requests are sent to the protocol stack thread through RPC.

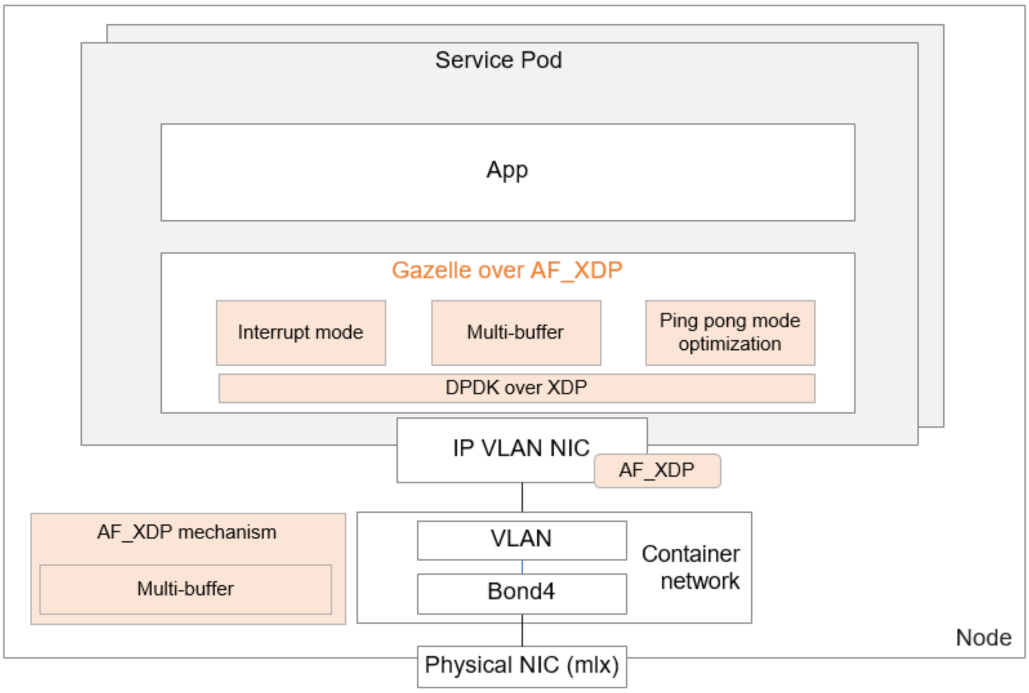
Figure 5-4 Thread model in RTW mode



4. **over AF_XDP mode**

In cloud-native container networks, this mode helps overcome network limits and decouple resources. Specifically, this mode (1) does not exclusively occupy NICs and uses AF_XDP to implement NIC queue passthrough to the user mode; (2) does not exclusively occupy CPUs and supports interrupts, which lowers CPU usage when the workload is light.

Figure 5-5 Thread model in over AF_XDP mode



Application Scenarios

Gazelle can immediately boost database, middleware, containerized app, and overall system performance. In redis-benchmark tests on system performance, Gazelle improves SET and GET performance by 1.6 to 3.5 times compared to the kernel mode. In MySQL TPCC tests, Gazelle delivers 10% to 20% performance gains.

Gazelle breaks through the limitations of traditional user-mode protocol stacks on performance, compatibility, and deployment flexibility. It significantly enhances network processing efficiency and opens up opportunities for system optimization in throughput- and latency-demanding scenarios. This breakthrough allows Gazelle to support a wider range of service demands, including cloud-native platforms, edge computing, high-performance databases, and financial transaction systems. Yet its full potential remains untapped, with many new applications still being explored and tested. The Gazelle community is actively evolving, continuously drawing feedback from developers and enterprise users, improving the functional ecosystem, and promoting standardization and best practice. It is expected to become a core component of the next-generation high-performance network infrastructure.

Repositories

https://gitee.com/openeuler/gazelle

BiSheng JDK

SIG	Compiler

BiSheng JDK is an open source edition of the Huawei JDK developed on OpenJDK. It is a high-performance OpenJDK distribution that can run in the production environment thanks to its extensive development by the dedicated team, who solved many issues caused by the native OpenJDK defects in actual deployments. BiSheng JDK offers a stable, reliable, high-performance, and easy-to-debug JDK, and can even be supercharged when run on the Kunpeng AArch64 architecture.

Challenges

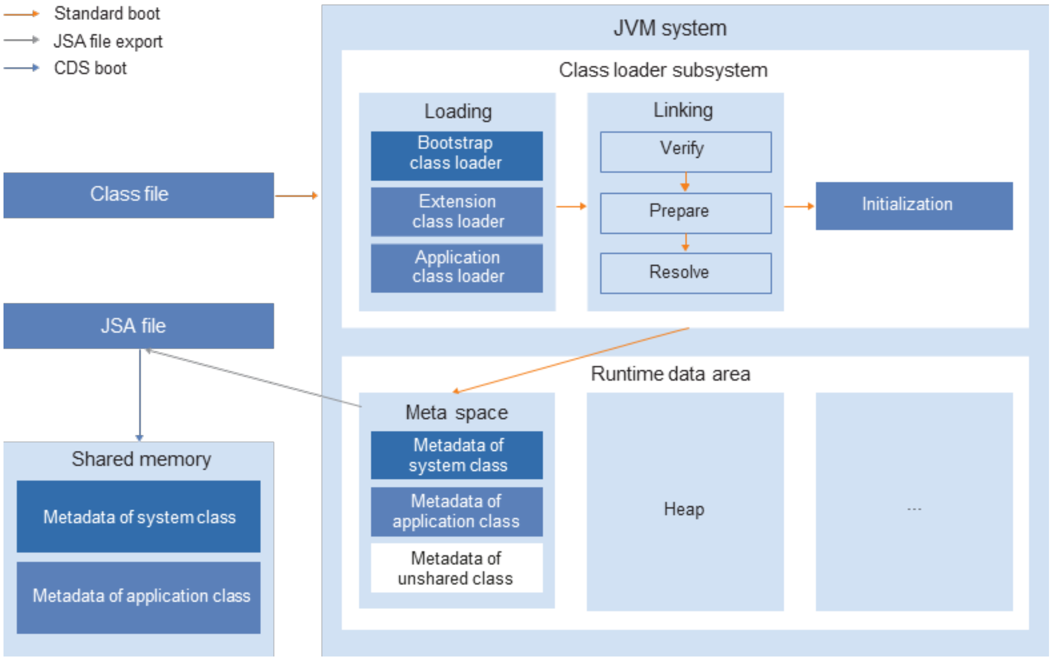
JDK is the core component for Java environments that has been plagued by issues of performance and stability. Despite difficulties in introducing new major features to OpenJDK 8 during the maintenance period, optimization features have been added to BiSheng JDK to resolve Java startup, garbage collection (GC), latency, encryption and decryption, and communication issues.

Feature Description

Feature 1: AppCDS

In the initial phase of running a Java program, class loading is time-consuming and needs to be performed each time the program is running. Class data sharing (CDS) saves the loaded class data to a file, so that the next time a user runs a Java program, the loaded class data is directly restored from the file to the memory. BiSheng JDK 8 provides application class data sharing (AppCDS), an extension of OpenJDK's CDS, to support application classes, improving performance by more than 7% on average in Hive SQL scenarios.

Figure 5-6 AppCDS service process



Feature 2: Promptly Return Unused Committed Memory from G1

The Garbage-First garbage collector (G1 GC) may not return committed Java heap memory to the OS in a timely manner, and may perform it only after a full GC operation. Since G1 tries hard to completely avoid full GCs, it will not return Java heap memory in many cases unless forced to do so externally.

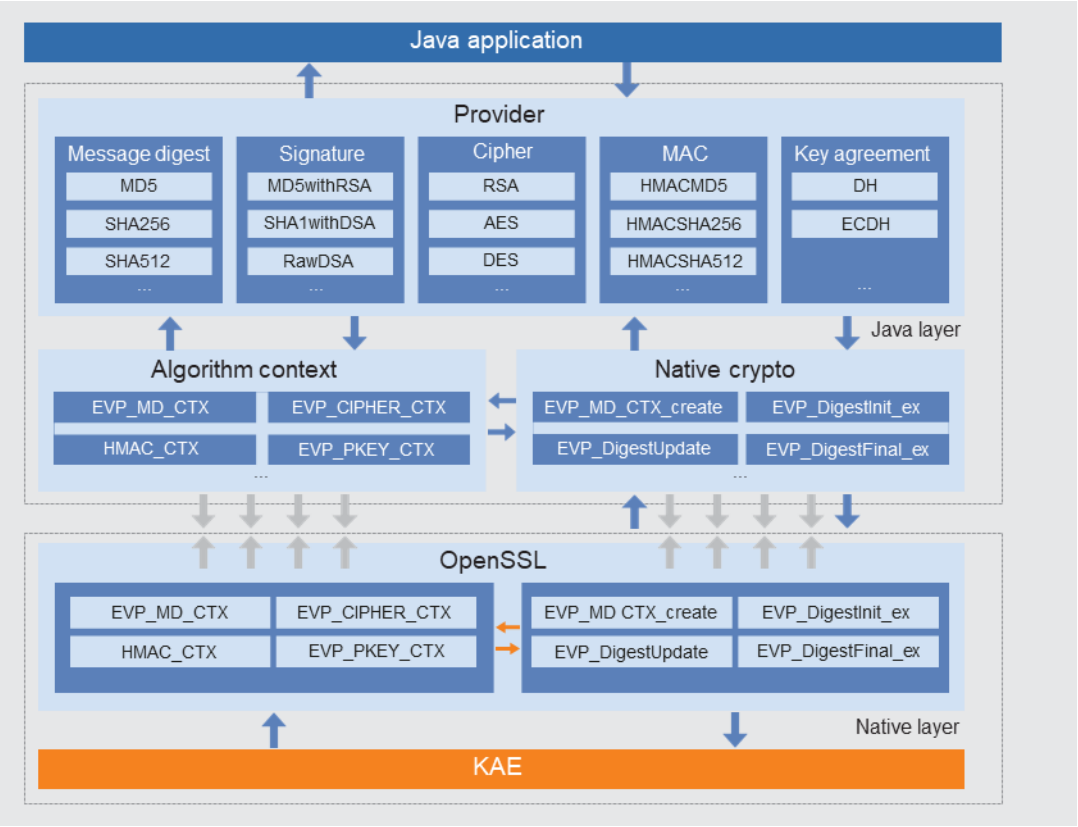
This behavior is particularly disadvantageous in container environments where resources are paid by use. Even when the JVM only uses a fraction of its assigned memory resources due to inactivity, G1 will retain all of the Java heap, causing users to pay full price for the resources or cloud providers unable to fully utilize their hardware. One solution is to ensure the JVM can detect phases of Java heap under-utilization, and automatically reduce its heap usage. Assuming 49 microservices are running and this feature is enabled, G1 can return unused committed memory to reduce the physical memory committed by G1 by 40% compared with that of the default settings.

Feature 3: KAE Provider

The Kunpeng Accelerator Engine (KAE) is an encryption and decryption module that supports the RSA, SM3, SM4, DH, MD5, and AES algorithms, and provides high-performance symmetric and asymmetric encryption/decryption algorithms. The KAE is compatible with OpenSSL 1.1.1a and later versions, and can work in synchronous or asynchronous mode.

BiSheng JDK adopts the Provider mechanism to support KAE encryption and decryption of Kunpeng servers, improving the security and related services on Kunpeng AArch64 servers. In HTTPS scenarios, the performance is doubled.

Figure 5-7 KAE Provider architecture



The KAE Provider supports the following algorithms.

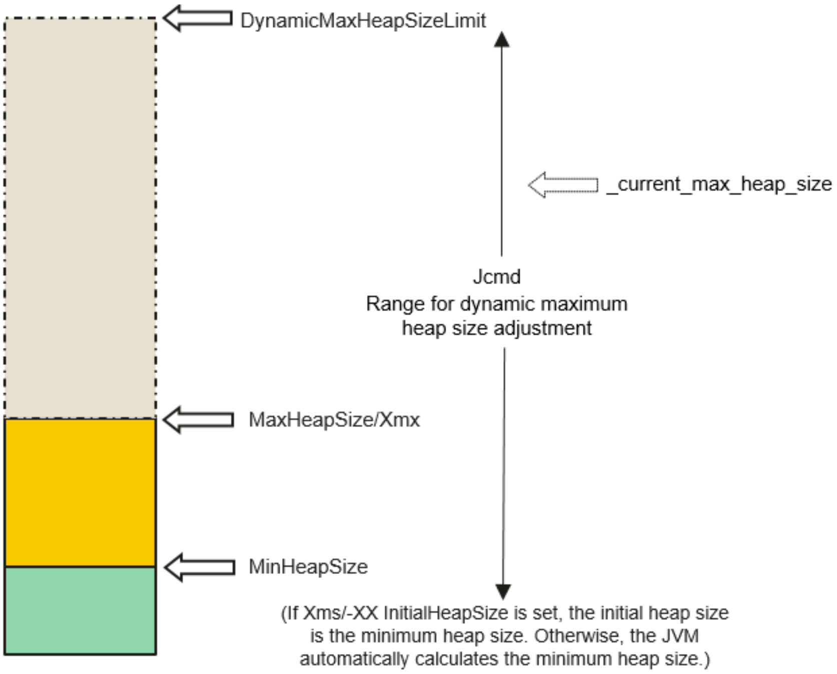
Algorithm	Description
Digest algorithms	MD5, SHA256, SHA384, SM3
AES	ECB, CBC, CTR, GCM
SM4	ECB, CBC, CTR, OFB
HMAC	HMACMD5, HMACSHA1, HMACSHA224, HMACSHA256, HMACSHA384, HMACSHA512
RSA	512-bit, 1024-bit, 2048-bit, 3072-bit, and 4096-bit keys
DH	DHKeyPairGenerator and DHKeyAgreement; 512-bit, 1024-bit, 2048-bit, 3072-bit, and 4096-bit keys
ECDH	ECKeyPairGenerator and ECDHKeyAgreement; secp224r1, prime256v1, secp384r1, and secp521r1
RSA signatures	RSASignature and RSAPSSSignature; only RSAPrivateCrtKey supported for private keys

Feature 4: Dynamic JVM Heap Scaling

Containerized deployment is popular today, allowing for vertical resource scaling. OpenJDK sets the maximum heap memory only at startup and cannot dynamically adjust it during operation. This prevents Java applications from accessing extra memory added by containers.

BiSheng JDK supports dynamic memory scaling, allowing users to update the Java heap memory upper limit during application running without restarting the JVM or causing service interruption. This feature enables dynamic vertical scaling in container environments and also supports lowering the heap upper limit.

Figure 5-8 Heap upper limit scaling


DynamicMaxHeapSizeLimit
_current_max_heap_size
Jcmd
Range for dynamic maximum heap size adjustment
MaxHeapSize/Xmx
MinHeapSize
(If Xms/-XX InitialHeapSize is set, the initial heap size is the minimum heap size. Otherwise, the JVM automatically calculates the minimum heap size.)

Application Scenarios

BiSheng JDK is common Java software developed and distributed on OpenJDK. It is widely used in Linux environments to process big data, middleware, and encryption and decryption tasks, in industries like finance, middleware, carrier, and Internet. On average, it improves the Spark performance by 10%, while for encryption and decryption improves by over 100%.

Repositories

BiSheng JDK 8, BiSheng JDK 11, BiSheng JDK 17, and BiSheng JDK 21 are open sourced, with quarterly version and feature updates. Java developers can obtain the latest information and communicate with each other in the BiSheng JDK open source community.

Software	Delivery Type	URL
BiSheng JDK 8	Open source code repository	https://gitee.com/openeuler/bishengjdk-8
BiSheng JDK 11	Open source code repository	https://gitee.com/openeuler/bishengjdk-11
BiSheng JDK 17	Open source code repository	https://gitee.com/openeuler/bishengjdk-17
BiSheng JDK 21	Open source code repository	https://gitee.com/openeuler/bishengjdk-21

GCC for openEuler

SIG	Compiler

The GCC for openEuler compiler is developed based on the open source GNU Compiler Collection (GCC). The open source GCC is the de facto standard of cross-platform compilers, and it complies with the GPLv3 license, becoming the most widely used C/C++ compiler on Linux. GCC for openEuler inherits capabilities of the open source GCC. It also has optimizations on C, C++, and Fortran languages and delivers enhanced features such as automatic feedback-directed optimization (FDO), software-hardware collaboration, memory optimization, and automatic vectorization. GCC for openEuler is compatible with a wide range of hardware platforms such as Kunpeng, Phytium, and Loongson, fully unleashing the computing power of these hardware platforms.

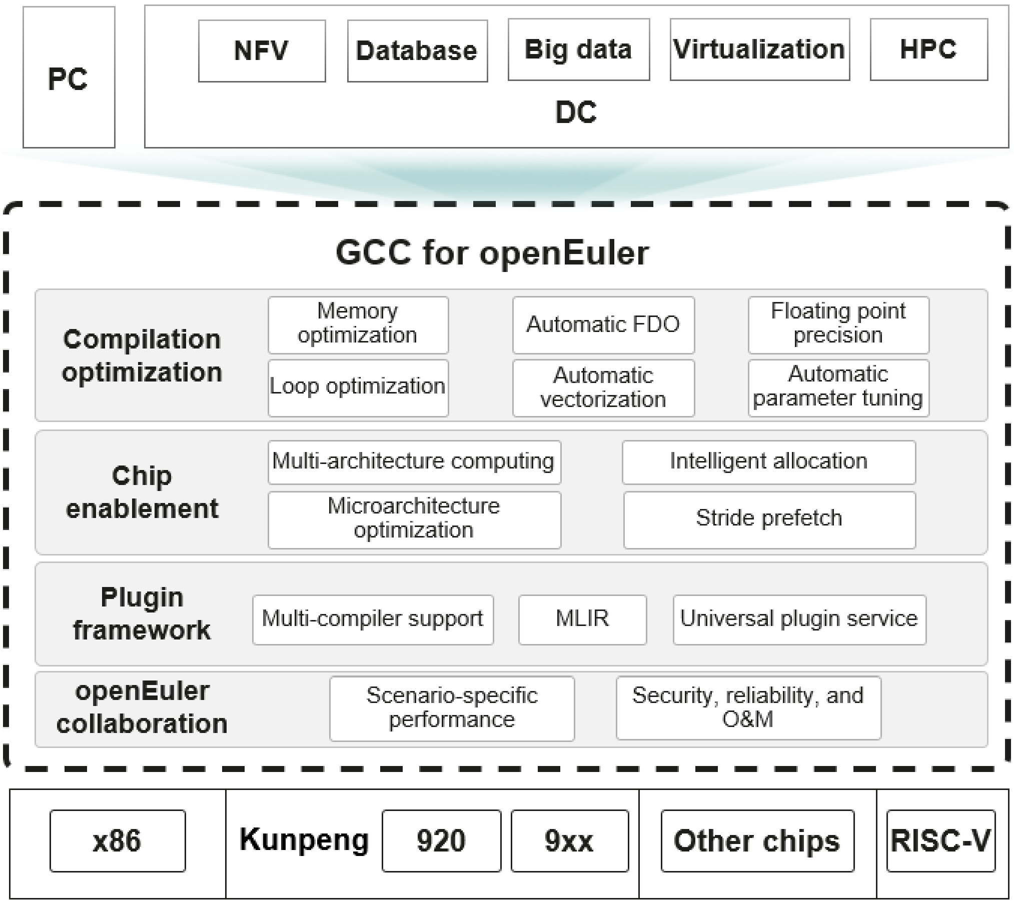
Challenges

GCC is fundamental to the OS as it is the default compiler of the Linux kernel and the de facto standard of cross-platform compilers. Any change made to GCC may heavily impact upper-layer applications. Therefore, GCC developers must familiarize themselves with the knowledge needed, such as compilation principles while improving feature security and robustness. GCC for openEuler aims to boost the performance of upper-layer applications by providing more advanced features than the open source GCC. The GCC is a fixed topic at the Compiler SIG meetings conducted every two weeks, which all are welcome to attend.

Feature Description

GCC for openEuler is compatible with mainstream hardware platforms including Kunpeng and x86, for use in openEuler performance, security, reliability, and O&M projects, for example, working in the compiler plugin framework to provide universal plugins. GCC for openEuler supports multi-architecture computing and microarchitecture optimization to implement intelligent memory allocation, memory optimization, and automatic vectorization. Besides that, GCC for openEuler incorporates industry-leading FDO technologies to implement automatic FDO, improving application performance for databases. GCC for openEuler has made major breakthroughs in the following aspects:

1. Basic performance: Improves computing performance in general scenarios and also applies to multi-architecture computing.
2. FDO: Integrates industry-leading FDO technologies to implement multi-modal FDO throughout the process, improving key applications such as databases in cloud-native applications.
3. Chip enablement: Supports multi-architecture computing instruction sets and leverages computing advantages based on hardware systems such as memory to improve scenario-specific performance such as HPC.
4. Plugin framework: Offers one set of plugins that is compatible with different compilation frameworks, streamlining the GCC and LLVM ecosystems.



Application Scenarios

GCC for openEuler is developed based on the open source GCC. It is widely used in Linux environments such as openEuler and is perfect for databases, virtualization, and HPC. On the Arm platform, GCC for openEuler delivers 20% higher basic performance of SPEC CPU 2017 than the open source GCC, boosting the performance of the MySQL database by more than 15%.

Repositories

The GCC for openEuler code has been open-sourced and is upgraded together with official openEuler releases. GCC developers can find the latest updates in the openEuler community.

Software Product	Delivery Type	URL
GCC for openEuler	Code repository	https://gitee.com/openeuler/gcc
	Package repository	https://gitee.com/src-openEuler/gcc

LLVM for openEuler

SIG	Compiler

The open source LLVM project provides a collection of tools and libraries for application compilation. This project has attracted extensive attention from developers, and companies have launched their commercial compilers based on the LLVM project. The Huawei BiSheng Compiler team is the primary contributor to this project. LLVM for openEuler is innovative in terms of compatibility, performance, and development-state secure coding. It adapts to multiple hardware platforms, such as Kunpeng, Phytium, and Loongson, to fully unleash diversified computing power.

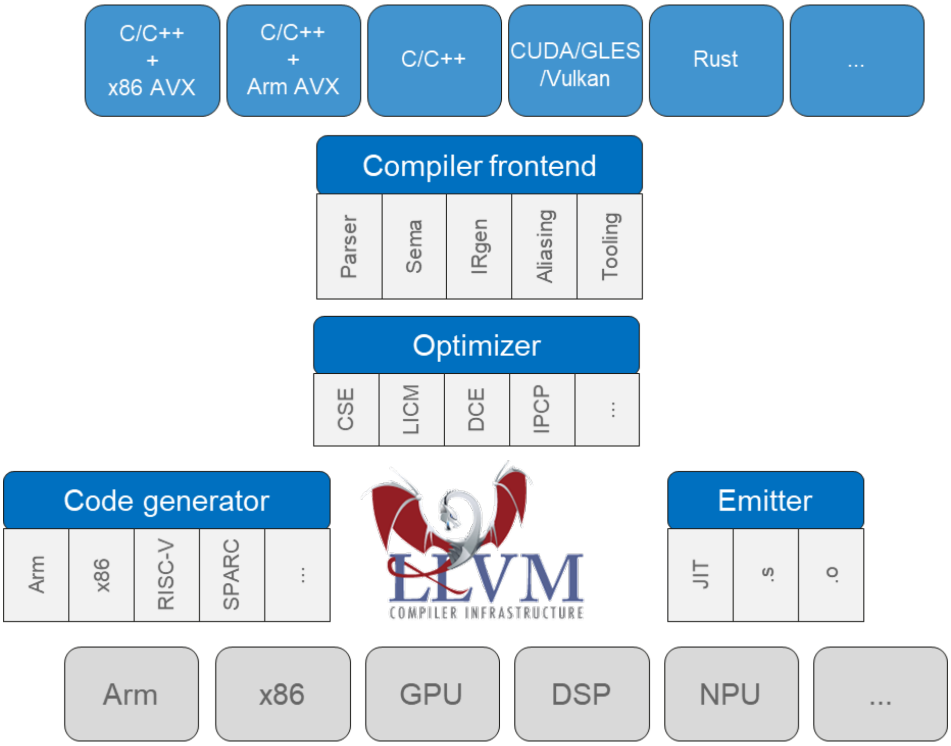
Challenges

LLVM for openEuler is an alternative compiler on openEuler and needs to bring more competitive capabilities than GCC. Specifically, it needs to provide powerful and scalable optimization capabilities to bring more performance gains in major computing scenarios, such as databases, big data, distributed storage, and virtualization. In addition, LLVM for openEuler should develop a comprehensive ecosystem to expand the user base. Specifically, it must be compatible with existing software packages, and also support the efficient compilation of new software packages.

Feature Description

LLVM has a modular architecture design and divides the compilation process into multiple independent phases, such as the frontend, optimization, and backend. This design makes LLVM more flexible and scalable, facilitates the independent evolution and innovation of modules in each phase, and combines different modules through the unified intermediate representation (IR). The LLVM project contains multiple subprojects, such as Clang, Flang, LLVM, MLIR, and LLD.

- **Architecture**



Feature 1: Sanitizer

The LLVM Sanitizer is a toolset used for dynamic code analysis and bug detection. It helps developers detect and debug common memory errors and security problems. These tools are closely integrated with the LLVM compiler and runtime library to facilitate code bug detection and diagnosis.

Function	How to Use	Issues to Detect
Fast memory error detection	-fsanitize=address	• Out-of-bounds accesses to heap/stack/globals • Use-after-free • Use-after-return • Use-after-scope • Double-free • invalid free • Memory leaks
Data race detection	-fsanitize=thread	• Data races
Memory detector	-fsanitize=memory	• Uninitialized reads • Use-after-destruction
Undefined behavior detection	-fsanitize=undefined	• integer-divide-by-zero • Bitwise shifts that are out of bounds for their data type • Dereferencing misaligned or null

Function	How to Use	Issues to Detect
		pointers ● Signed integer overflow
Hardware-assisted memory error detection	-fsanitize=hwaddress	● Same and AddressSanitizer
Stack protection	-fsanitize=safe-stack	● Protects programs from stack buffer overflow attacks.

Feature 2: Clang Extra Tools

Clang Extra Tools are a group of additional tools provided by the LLVM project. They are used together with the Clang C/C++ compiler to support static code analysis, code refactoring, and code style check.

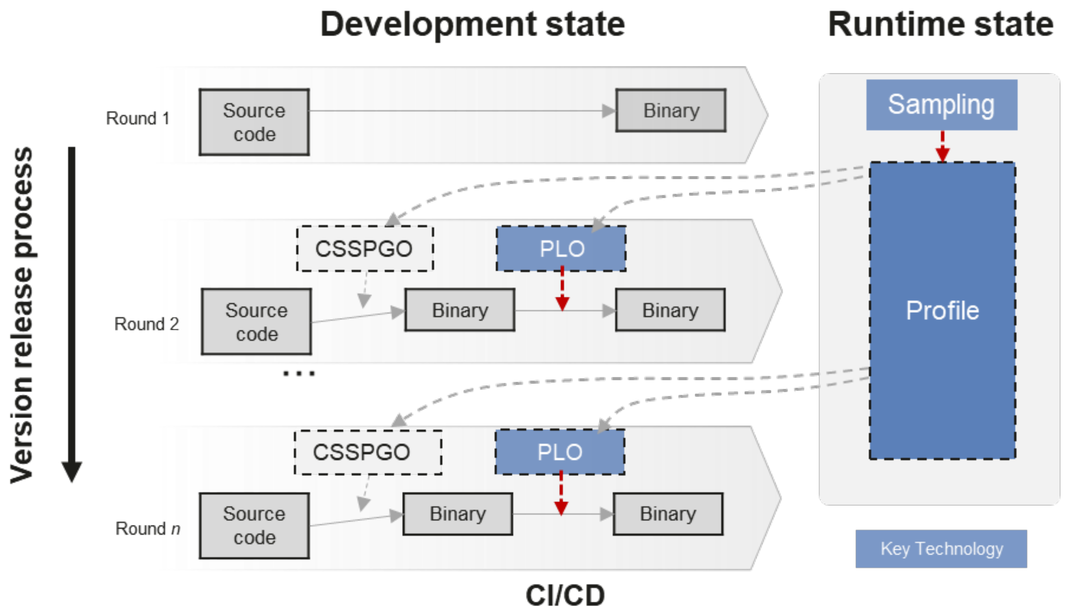
Clang-Tidy: A powerful static code analyzer used to check for common errors, potential problems, and code style violations in C, C++, and Objective-C code. It automatically detects and fixes problems in code, helping developers write higher-quality code.

Clang-Format: A code formatter that automatically formats C, C++, and Objective-C code. It can automatically adjust the indentations, line feeds, and spaces of the code according to the configured rules. It helps maintain a consistent code style among team members, improving code readability and maintainability.

Clang-Check: A tool for compiling a custom static analyzer. It lets developers customize static analysis rules for detecting specific problems or potential errors in code. It provides powerful APIs and frameworks, enabling developers to create tailored code check tools based on their requirements.

Feature 3: CFGO

Continuous feedback-guided optimization (CFGO) is recommended for large-scale data center applications experiencing significant frontend performance bottlenecks due to high i-cache and TLB miss rates. CFGO periodically collects runtime environmental information for continuous feedback optimization, ensuring both low overhead and strong performance generalization.



Application Scenarios

As a C/C++/Fortran/Rust compiler, LLVM for openEuler can be used to build server, cloud computing, edge computing, and embedded applications. Compared with the mainstream GCC versions, it delivers more than 30% higher basic performance of SPEC 2017 on the Arm platform, and 5% to 10% higher performance in mainstream computing scenarios. LLVM for openEuler is suitable for mainstream computing and cluster computing scenarios, such as big data, databases, distributed storage, and virtualization.

Repositories

Source code repository:

https://gitee.com/openeuler/llvm-project

Artifact repository:

https://gitee.com/src-openEuler/clang

https://gitee.com/src-openEuler/llvm

https://gitee.com/src-openEuler/lld

Go for openEuler

SIG	Golang

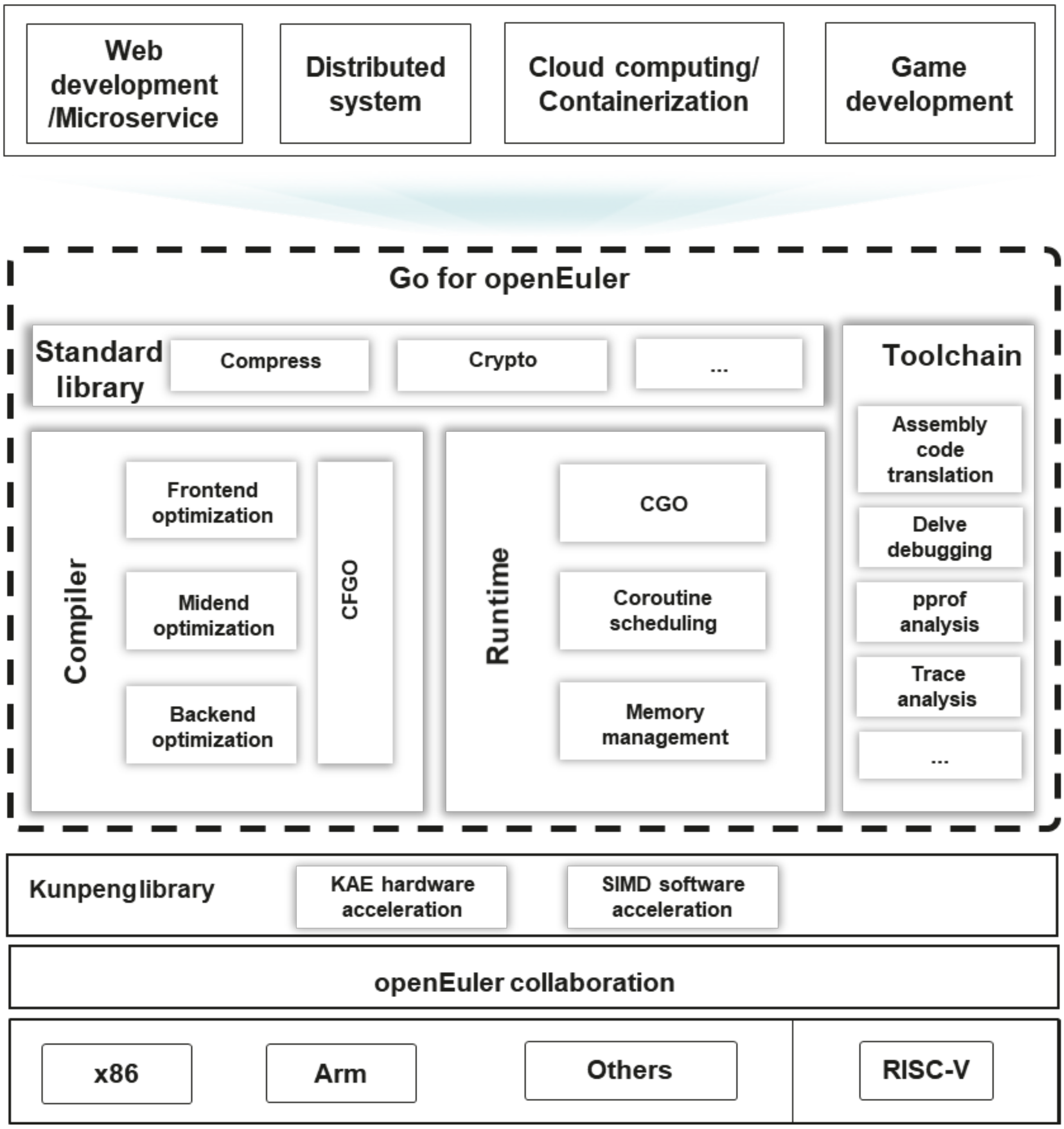
Go for openEuler is a high-performance, reliable, and easy-to-develop Golang release based on open source Golang. It focuses on building an openEuler-based compiler that ensures strong ecosystem compatibility and delivers ultimate user experience.

Go for openEuler is optimized for mainstream Go service loads in container cloud scenarios that require agile development and high running performance, such as cloud native and microservice applications. It solves performance issues caused by insufficient native Golang capabilities and adapts to hardware platforms such as Loongson and Kunpeng, fully unleashing the hardware computing power.

Challenges

Go is used to develop microservice applications in many internet service scenarios, such as Douyin, JD.com. Core applications such as etcd, CubeFS, and Sonic are also built on Go. However, the native capabilities of Go's compiler, standard library, and runtime fall short in meeting the performance demands of these large-scale internet operations.

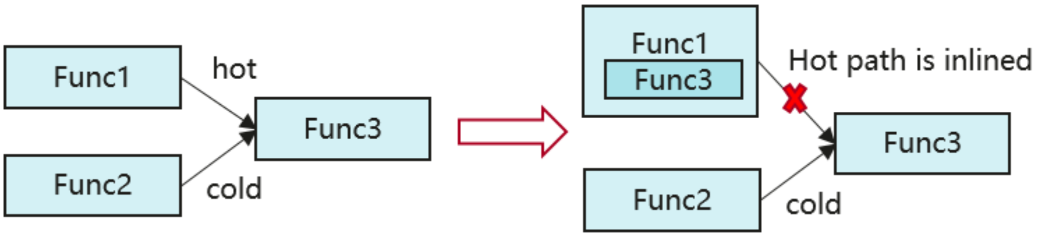
Feature Description



Feature 1: CFGO

CFGO collects runtime information while preserving program functionality to help the compiler make more accurate optimization decisions, resulting in a more efficient target program. Based on the principle of program locality, hot instructions are arranged more closely to improve cache and TLB hit rates, effectively reducing front-end bottlenecks and enhancing overall program performance.

Figure 5-9 Function inlining



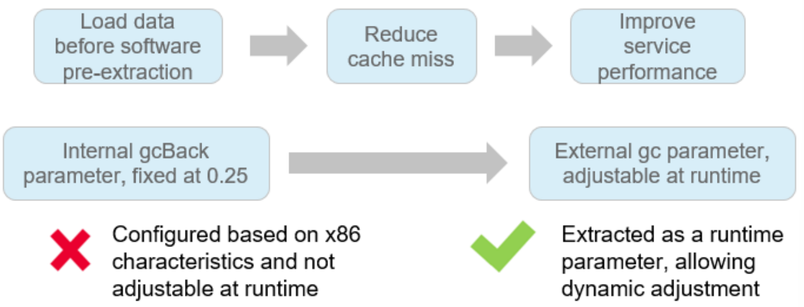
Feature 2: Arm Atomic Instruction Optimization

In certain service scenarios, the use of CAS locks and LD/ST instructions in the Golang runtime incurs considerable overhead. Replacing them with Arm-based instruction sequences can lead to noticeable performance improvements.

Feature 3: Runtime GC Optimization

Based on the rumtime characteristics, software prefetching is inserted to improve data access efficiency. The GC coroutine overhead parameters are extracted as runtime parameters, allowing dynamic adjustment based on service characteristics.

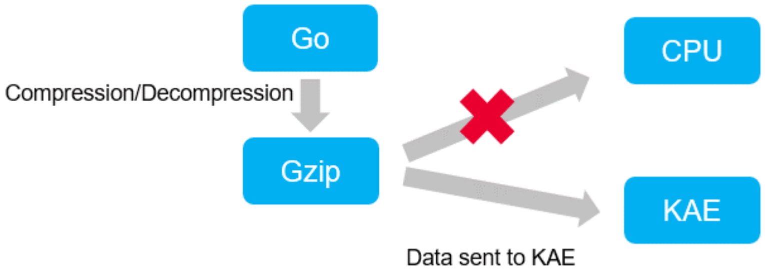
Figure 5-10 Runtime GC optimization


Load data before software pre-extraction
Reduce cache miss
Improve service performance
Internal gcBack parameter, fixed at 0.25
External gc parameter, adjustable at runtime
Configured based on x86 characteristics and not adjustable at runtime
Extracted as a runtime parameter, allowing dynamic adjustment

Feature 4: Combining KAE to Enable Underlying Hardware Acceleration

The compression/decompression logic of Golang's Gzip Compress library is modified to enable underlying hardware acceleration.

Figure 5-11 KAE enablement



Application Scenarios

Go for openEuler is a software foundation developed using Golang, widely used in Linux environments such as openEuler. It covers application scenarios including cloud native, distributed storage, and cloud gaming, helping internet customers improve performance by 20% in mainstream service scenarios.

Repositories

https://gitee.com/openeuler/golang

Compiler Plugin Framework

SIG	Compiler

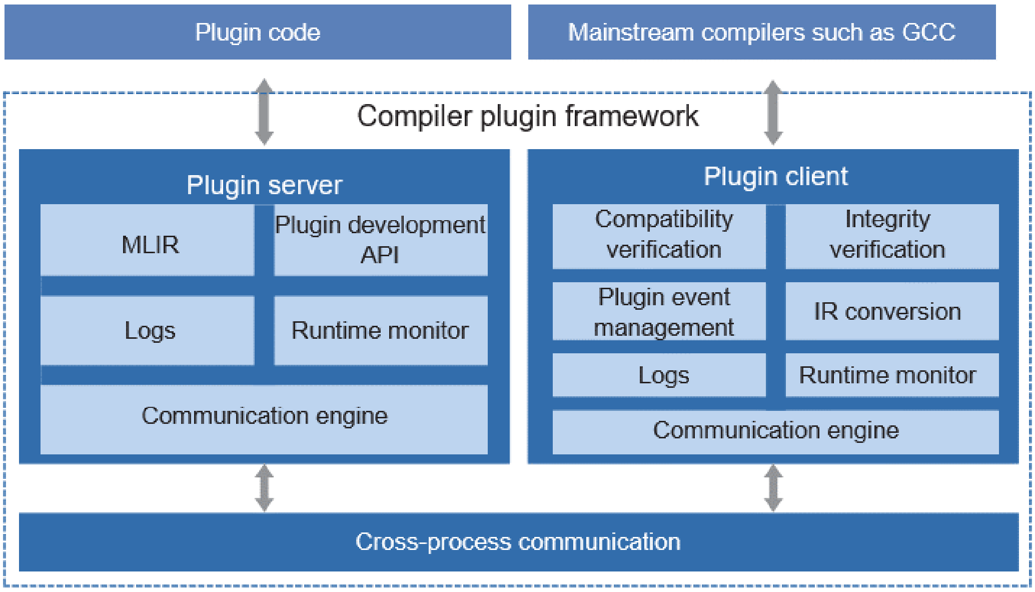
The compiler plugin framework is a plugin development platform that provides MLIR-oriented interfaces, helping develop a plugin while applying it on multiple compilers. The framework supports and maintains common capabilities such as plugin compatibility and integrity checks.

Challenges

There are two major compiler frameworks: GCC and LLVM. A large number of compilation tools and extended compilation capabilities are developed based on the two compiler frameworks. The code for one compiler framework cannot be reused on another. Therefore, compilation tool development faces the following difficulties:

- The compiler needs in-depth modification, which also complicates compiler maintenance.
- Repetitive coding occurs when you develop the compilation tool based on the two compilation frameworks.
- Lack of base capabilities such as compatibility increases tool development and maintenance costs.

Feature Description



1. MLIR-based plugin development and easy conversion of IRs such as GIMPLE
2. The compiler plugin framework supporting 19 classes of GIMPLE statements
3. Common capabilities such as compatibility and binary integrity checks
4. Monitoring and verifying plugin status, such as for compiler security and operations
5. Executing plugin clients as GCC plugins, so functions can run without modifying the GCC compiler code
6. Link time optimization (LTO)

Application Scenarios

Scenario 1: Compilation tool build and integrity verification

The compiler plugin framework can work as the development platform to build compilation tools while applying them on multiple compilers such as GCC. The framework supports and maintains common capabilities such as compatibility and binary integrity checks.

Scenario 2: Quick enabling and verification of compilation tools

The compiler plugin framework helps develop compilation tools as plugins to run on mainstream compilers such as GCC. There is no need to modify the source code of the compilers, streamlining development efficiency.

Repositories

https://gitee.com/openeuler/pin-gcc-client

https://gitee.com/openeuler/pin-server

High Security and Reliability

secGear

SIG	Confidential computing

secGear, an open source confidential computing component of openEuler, provides a simple and easy-to-use confidential computing software stack and solution. By lowering the threshold of adoption, it drives the development of the confidential computing ecosystem.

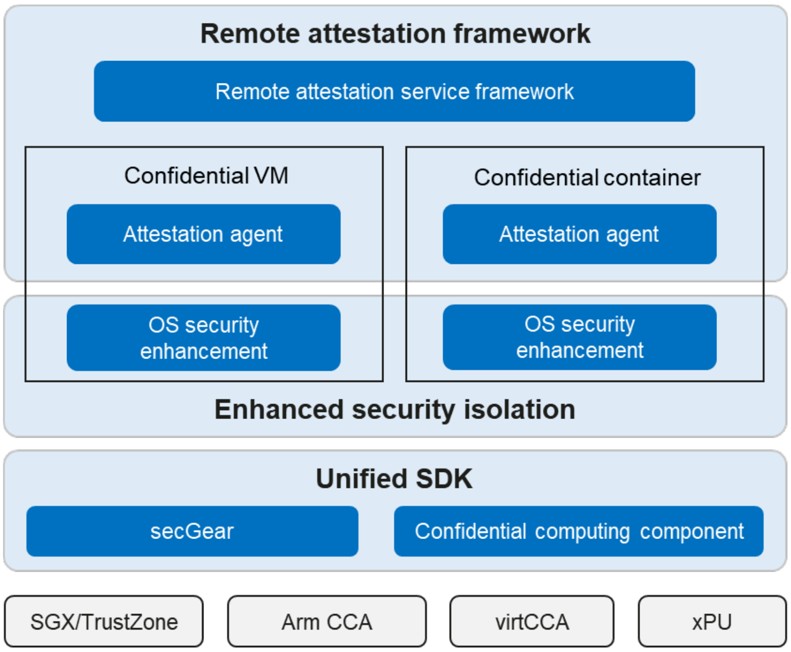
Challenges

With the continuous advancement of confidential computing and cloud-native technologies, solutions such as confidential virtual machines (VMs) and confidential containers are gaining momentum among users. While these technologies provide OS-level secure and isolated runtime environments, they also bring new challenges:

- Secure isolation: Key components in the startup and runtime chain—such as virtualization and container runtimes—often fall outside the protection boundary, making them vulnerable. OS-level isolation still exposes services and ports that may be exploited as entry points. Furthermore, confidential VMs and containers consist of numerous components, any of which could introduce vulnerabilities or backdoors that threaten secure execution.
- Secure use: Differences in underlying implementations across TEE technology vendors hinder upper-layer applications from fully leveraging hardware capabilities and make it difficult to migrate confidential VMs or containers.
- Secure interconnection: While confidential VMs and containers simplify the development and deployment of secure applications, challenges remain in establishing trusted and secure connections—between users and confidential cloud applications or services, and among confidential VMs or containers themselves—to ensure data transmission security. Additionally, when confidential VMs or containers use heterogeneous computing devices such as xPUs, maintaining secure trust among these devices becomes a critical issue.

Feature Description

In the era of AI and cloud-native technologies, secGear, openEuler's confidential computing component, focuses on building a distributed, cloud-native confidential computing foundation—managing heterogeneous hardware TEEs at the southbound layer and supporting cloud-native deployment of service applications at the northbound layer.



secGear provides the following core capabilities:

- Remote attestation service framework: Includes attestation service and agent, enabling rapid deployment of a complete remote attestation system.
- Security isolation enhancement: For confidential VM and container scenarios, provides configuration hardening and secure OS image build capabilities to reduce the attack surface during startup and runtime.
- Unified SDK: Unifies interfaces across different SDKs to enable cross-architecture source compatibility. Supports zero-switching for frequent REE-TEE interactions and provides confidential computing components such as local attestation, cross-TEE RA-TLS, and secure channels for inter-TEE communication.

The new security features are as follows:

- Kuasar confidential container image encryption/decryption and secure key transmission based on remote attestation
- NPU firmware measurement and attestation for AI inference scenarios
- OS security configuration hardening guide and secure OS image tailoring and creation

Application Scenarios

secGear is widely used in openEuler AI full-stack security, encrypted database, hardware HSM replacement, and big data scenarios. It helps customers in industries such as finance and telecom quickly migrate services to the confidential computing environment and protect data security during running.

Repositories

https://gitee.com/openeuler/secGear

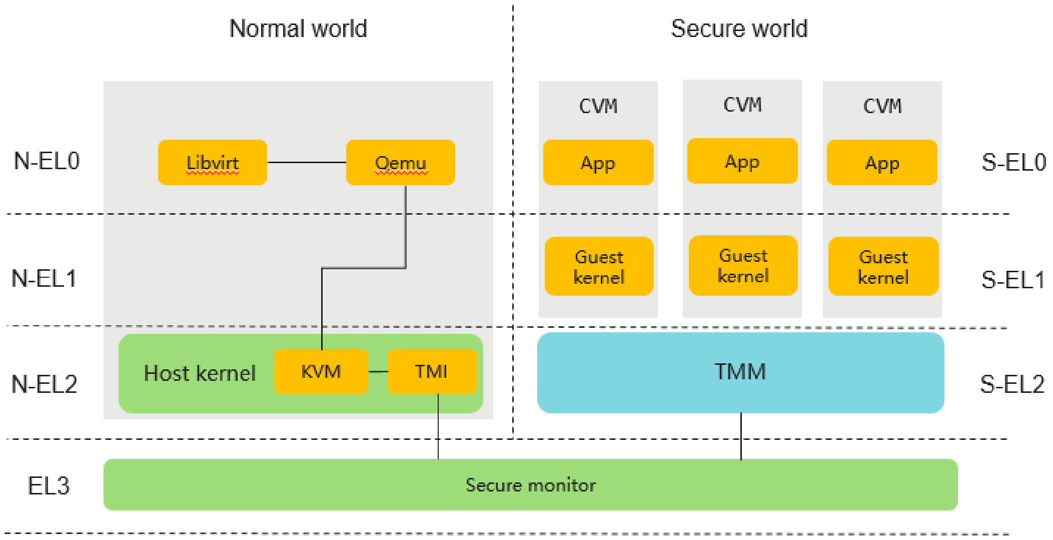
virtCCA Confidential Computing

SIG	Confidential computing

virtCCA implements security domain virtualization and supports confidential VMs and containers. Based on the Kunpeng platform, the confidential VM capability is implemented on the TEE side to seamlessly migrate the software stack in the existing common VMs to the confidential environment, a wider range of applications can be quickly migrated to a trusted environment, achieving trusted data circulation.

Challenges

Data transmission and storage have mature encryption measures, but the protection of data in the computing state is insufficient, making data protection throughout the lifecycle impossible. Confidential computing is a key technology to solve this problem. Traditional dedicated high-security TEEs require re-development or migration of applications. The application ecosystem is bound to the secure OS, which has a high usage threshold and cannot meet the requirements of cloud native general ecosystems, such as VMs and containers. Based on the standard interface of the Arm Confidential Compute Architecture (Arm CCA), openEuler builds a TEE virtualization management module upon the TrustZone firmware. This module supports memory isolation and the management of context, lifecycle, and page tables among confidential VMs, thereby enabling seamless application migration to TEEs.



Feature Description

1. Device passthrough

   Device passthrough utilizes the PCIe protection component embedded in the PCIe root complex of the Kunpeng processor. A selector is added to the PCIe bus to regulate communication between the CPU and peripherals. Operating through the system memory management unit (SMMU), this selector manages both the control logic and data transfer of both inbound and outbound traffic, safeguarding the entire data link.

   The device passthrough capability of virtCCA PCIPC offers secure isolation and performance enhancements for PCIe devices, with the following benefits:

(1) Secure isolation

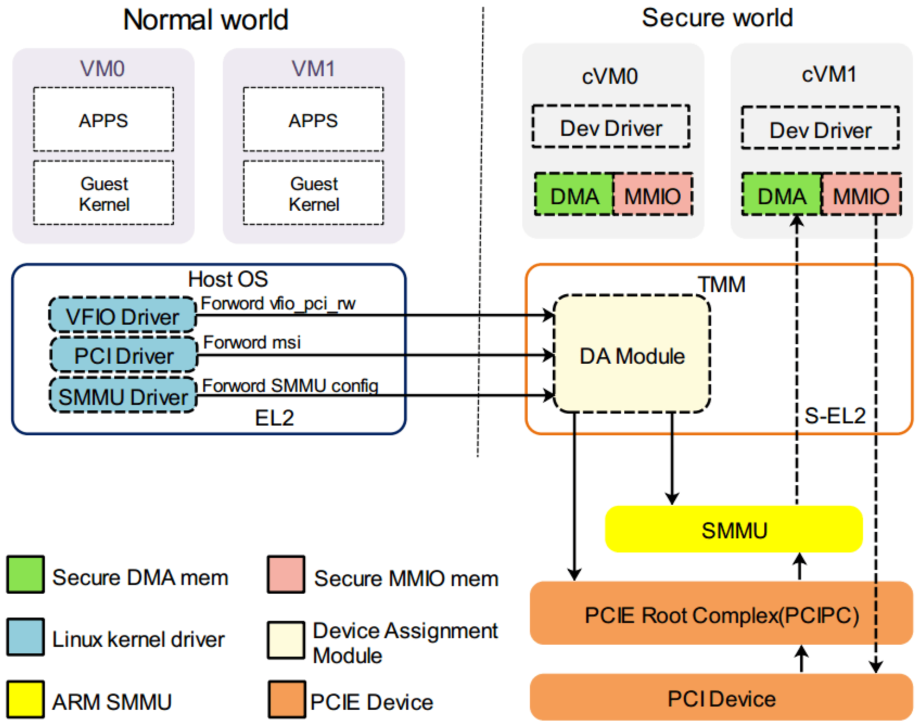
The TEE manages device access permissions, preventing host software from accessing TEE devices.

(2) High performance

Confidential device passthrough eliminates performance loss on the data plane compared to conventional encryption and decryption solutions.

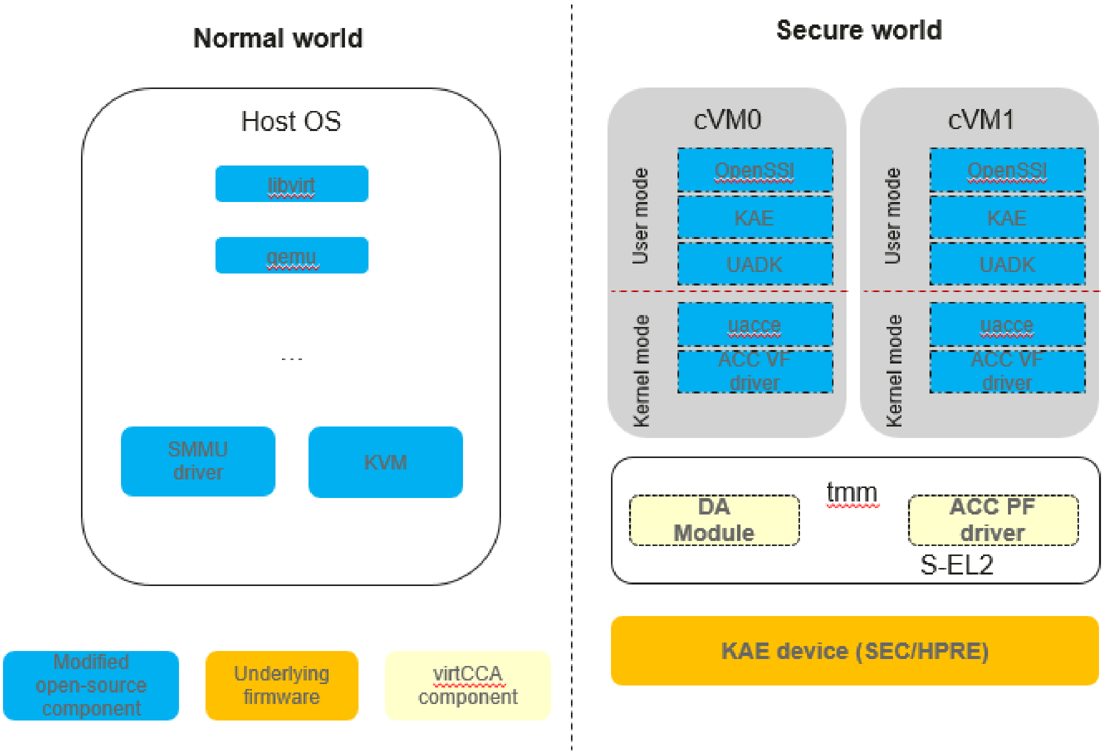
(3) High usability

Compatibility with existing open source OSs removes the requirement for kernel driver modifications.



2. Hardware-based acceleration for SM algorithms

Hardware-based acceleration for SM algorithms is powered by the Kunpeng processor, utilizing KAE capabilities in the TEE. It employs the UADK user-mode accelerator framework to enhance SM algorithm performance and enable algorithm offloading within confidential VMs.



Application Scenarios

virtCCA-based VMs are suitable for encrypted databases, secure cloud servers, multi-party confidential computing, trusted data circulation, and AI model data protection.

Repositories

https://gitee.com/openeuler/virtCCA_sdk

https://gitee.com/openeuler/virtCCA_driver

CCA

SIG	Virt

Arm CCA is a new confidential computing architecture specification introduced in Arm v9-A. It aims to define a standard confidential computing solution for next-gen computing devices.

openEuler implements CCA support for OS-related components (KVM, QEMU, libvirt, and guest kernel) based on the Arm CCA specifications. openEuler's community distribution provides native support for realm confidential VMs, meeting the security requirements for protecting data in use. In addition, openEuler ensures compatibility with traditional application ecosystems and VM management software.

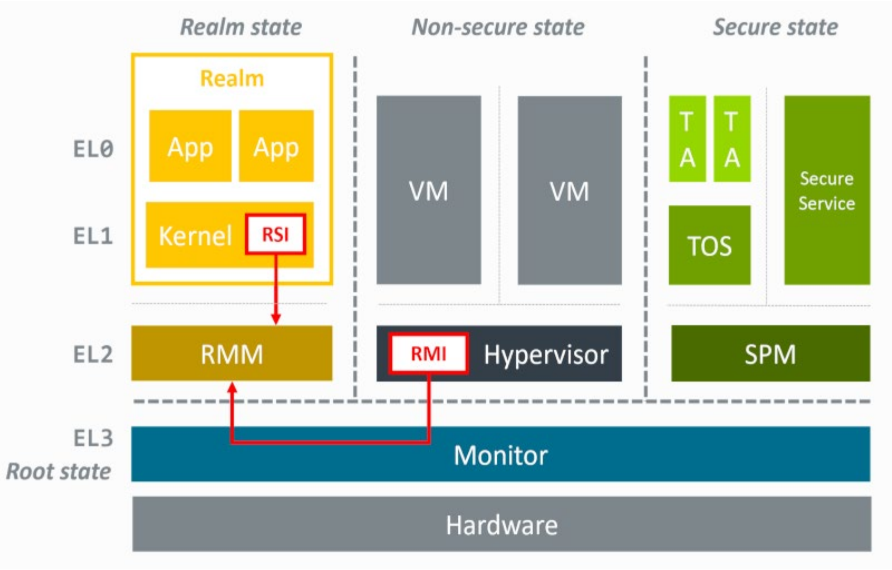
Challenges

Arm CCA faces multiple technical challenges to implement universal and efficient confidential computing.

- **Trust boundary establishment:** In cloud scenarios, the underlying software stack, such as the Hypervisor and OS, may be extremely complex, have vulnerabilities, or even be malicious. CCA's core design principle is to "reduce trust assumptions," explicitly excluding the Hypervisor and host OS from the trust boundary. They can schedule resources of realms, but cannot access the memory content.
- **Security of dynamic memory management:** Memory needs to be frequently allocated and released between different workloads. The challenge is how to allow the system manager (such as Hypervisor) to securely manage memory without exposing the realm memory, and implement memory clearing and reuse.
- **Remote attestation:** How do users remotely verify that their code is running in a real, hardware-enhanced Arm CCA environment, rather than a simulated malicious environment? This requires a set of hardware-based cryptographic proof mechanisms (remote attestation) so that the hardware itself can issue a "proof report" for users to verify.

Feature Description

Arm CCA works with the following core components to build a protected execution space—realm, which is completely isolated from the normal world in terms of code execution and data access.



- **Realm:** Realm is the core abstraction of CCA. It is a new type of execution environment parallel to the non-secure world and secure world (TrustZone). A realm is hardware-isolated and designed to host sensitive code and data. It is independent of the host OS and Hypervisor. The host OS and Hypervisor can manage the realm but cannot access the content within.
- **Dynamic management:** Hypervisor can dynamically create realms and allocate memory and CPU resources to them as required. However, after the realm is initialized, Hypervisor transfers its control to a protected secure virtualization module—realm management monitor (RMM), and can no longer access the data within the realm.

- **Memory management:** CCA extends the system memory management unit (MMU) to identify and isolate realm memory. Any access attempt from outside the realm (including Hypervisor) is blocked by the hardware, ensuring data confidentiality.
- **Remote attestation:** Each processor supporting CCA has a unique hardware-based identity. When the realm is started, it can generate an attestation token signed by hardware cryptography. Users can obtain the report and verify the signature and component measurement value to ensure that their workloads are running in a real and unaltered Arm CCA environment.

Application Scenarios

Confidential VMs are mainly used in cloud computing environments to provide high-level TEE for core workloads. They enable customers to process sensitive data (such as financial records, healthcare information, and intellectual property) on untrusted public clouds while ensuring the confidentiality and integrity of the data, which even cloud service providers cannot access. This effectively meets the key requirements of data sovereignty, regulatory compliance, and cross-agency privacy data collaboration (such as joint modeling and analysis).

Repositories

https://gitee.com/openeuler/kernel/tree/OLK-6.6

https://gitee.com/openeuler/qemu/tree/qemu-8.2.0

https://gitee.com/openeuler/libvirt/tree/libvirt-9.10.0

Simplified O&M

oeDeploy

SIG	Ops

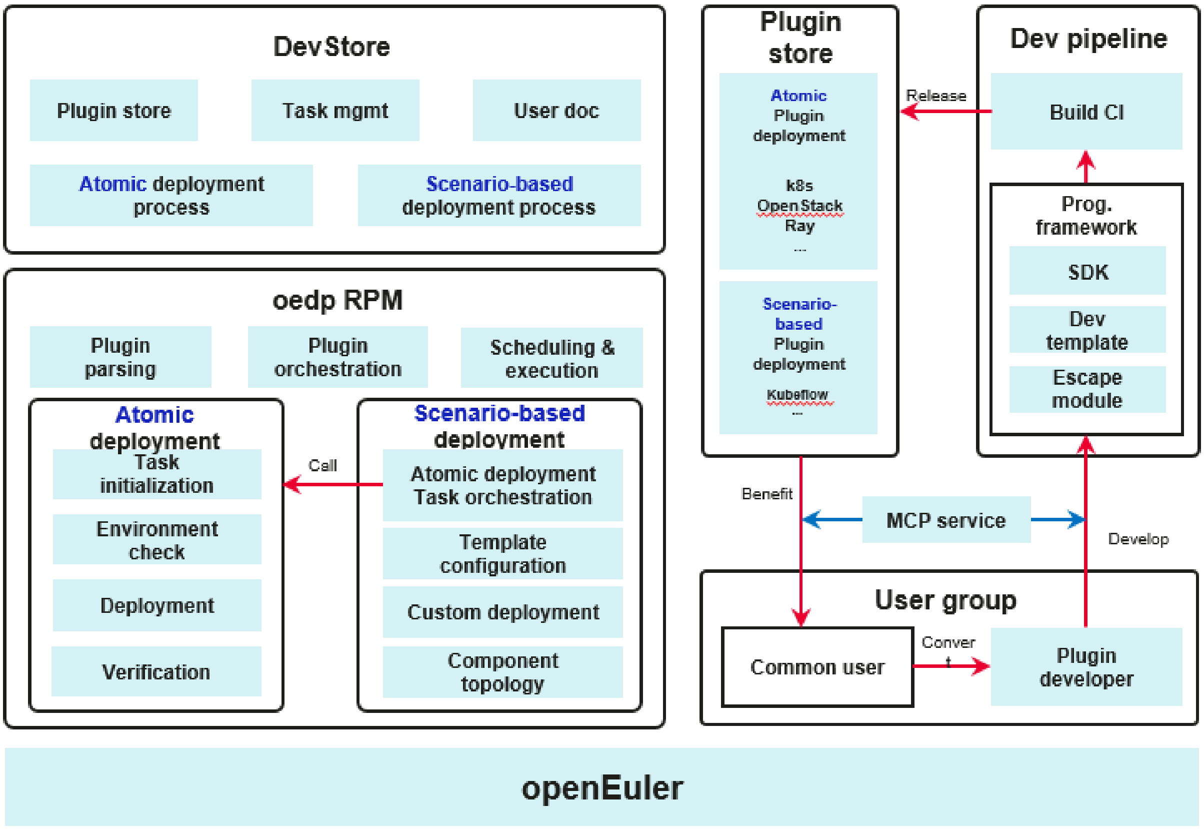
oeDeploy is a lightweight yet powerful software deployment tool that accelerates environment setup across single-node and distributed systems with high efficiency.

Feature Description

Multi-scenario support and quick deployment of mainstream software: oeDeploy facilitates quick deployment for both single-node applications and cluster software environments. It now includes quick deployment capabilities for Kubernetes environments with multiple master nodes. It also extends support to community toolchains like openEuler Intelligence and DevKit-pipeline, as well as popular Retrieval Augmented Generation (RAG) software such as RAGFlow, AnythingLLM, and Dify.

Flexible plugin management and excellent deployment experience: oeDeploy provides an extensible plugin architecture for flexible management of diverse deployment capabilities, empowering developers to quickly publish custom deployment plugins. It now supports plugin source management, enabling easy plugin updates and plugin initialization. While oeDeploy currently offers a streamlined CLI, a GUI and plugin store will soon launch, promising an even more efficient software deployment experience with less code.

Efficient deployment and intelligent development: oeDeploy introduces the MCP service, offering an out-of-the-box experience within DevStation. Leveraging LLM inference capabilities, it supports quick deployment of various software using natural language, boosting deployment efficiency by 2x. It can also convert user documents into executable oeDeploy plugins, increasing development efficiency by 5x.



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.

For developers and maintenance personnel, oeDeploy enables quick setup of complex environments, deploying mainstream AI training and inference frameworks in just minutes. This significantly streamlines software development and eliminates tedious configuration work. Developers can also extend oeDeploy by creating and sharing custom deployment templates, democratizing quick deployment for broader user communities. By leveraging foundation models and MCP capabilities, oeDeploy makes deployment more efficient and development more intelligent.

Repositories

https://gitee.com/openeuler/oeDeploy

A-Ops Intelligent O&M

SIG	Ops

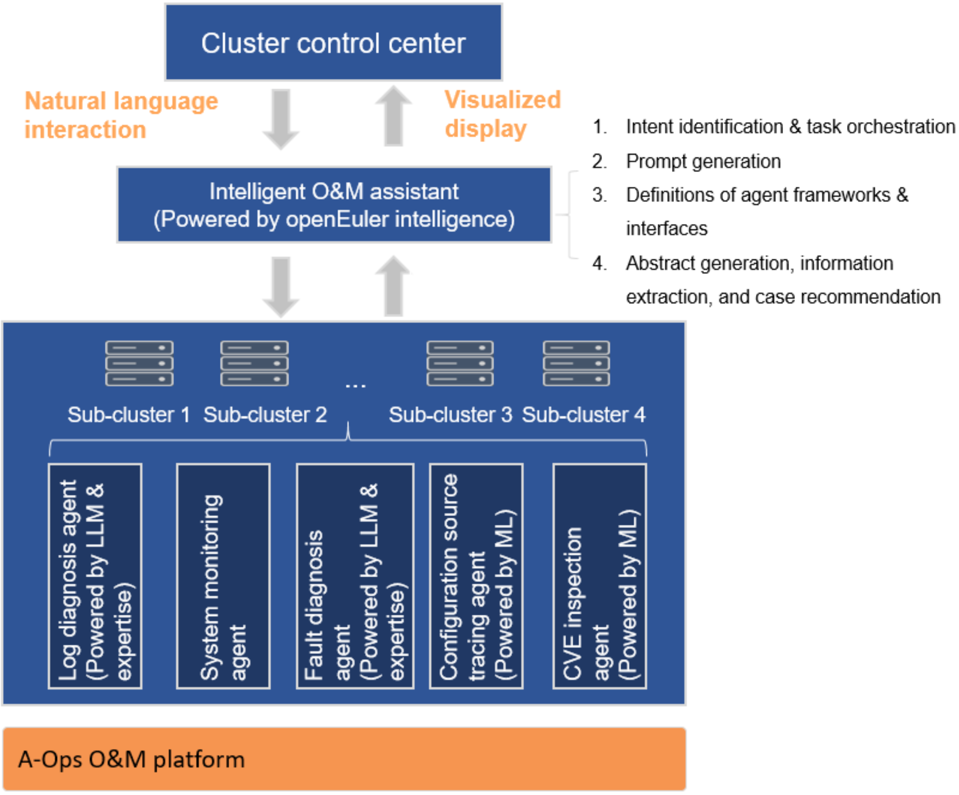
A-Ops is an OS-based fault O&M platform that provides intelligent O&M solutions for data collection, health check, fault diagnosis, and fault rectification. A-Ops enables intelligent O&M through interactive dialogs and wizard-based operations.

The intelligent interactive dialogs, 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.

Challenges

As cloud-native, serverless, and related technologies have proliferated in recent years, cloud infrastructure operations have become increasingly complex. A-Ops modernizes the O&M process with intelligent assistants that perform routine tasks through intelligent interaction, making O&M easier.

Feature Description


Cluster control center
Natural language interaction
Visualized display
Intelligent O&M assistant (Powered by openEuler intelligence)
1. Intent identification & task orchestration
2. Prompt generation
3. Definitions of agent frameworks & interfaces
4. Abstract generation, information extraction, and case recommendation
Sub-cluster 1
Sub-cluster 2
...
Sub-cluster 3
Sub-cluster 4
Log diagnosis agent (Powered by LLM & expertise)
System monitoring agent
Fault diagnosis agent (Powered by LLM & expertise)
Configuration source tracing agent (Powered by ML)
CVE inspection agent (Powered by ML)
A-Ops O&M platform

A-Ops integrates the intelligent O&M assistant based on the openEuler Intelligence for 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 assistant 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 interactive dialogs, 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.

Repositories

https://gitee.com/openeuler/aops-zeus

https://gitee.com/openeuler/aops-apollo

https://gitee.com/openeuler/aops-vulcanus

https://gitee.com/openeuler/aops-hermes

https://gitee.com/openeuler/aops-ceres

https://gitee.com/openeuler/authHub

6 Developer Support

Infrastructure

CVE Manager Vulnerability Management

SIG	Infrastructure/Security-committee

Vulnerability management is a general term for the processes, tools, and mechanisms used by the openEuler community to detect, collect, handle, and disclose security vulnerabilities.

Challenges

Vulnerability management is a general term for the processes, tools, and mechanisms used by the openEuler community to detect, collect, handle, and disclose security vulnerabilities.

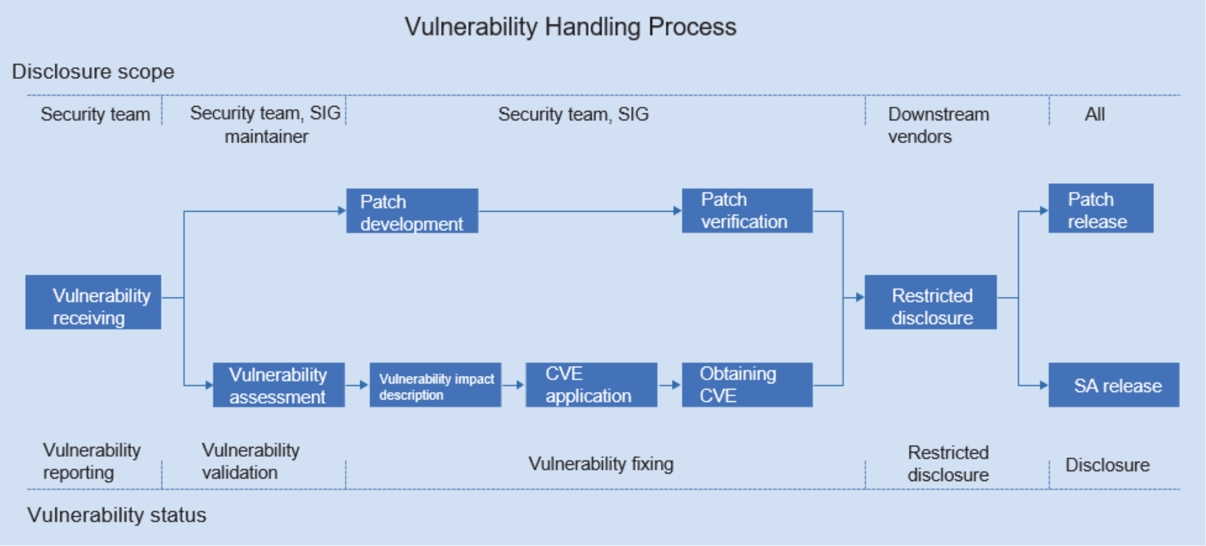
Objective:

- Timely detection
- Efficient analysis
- Quick fix
- Controlled disclosure

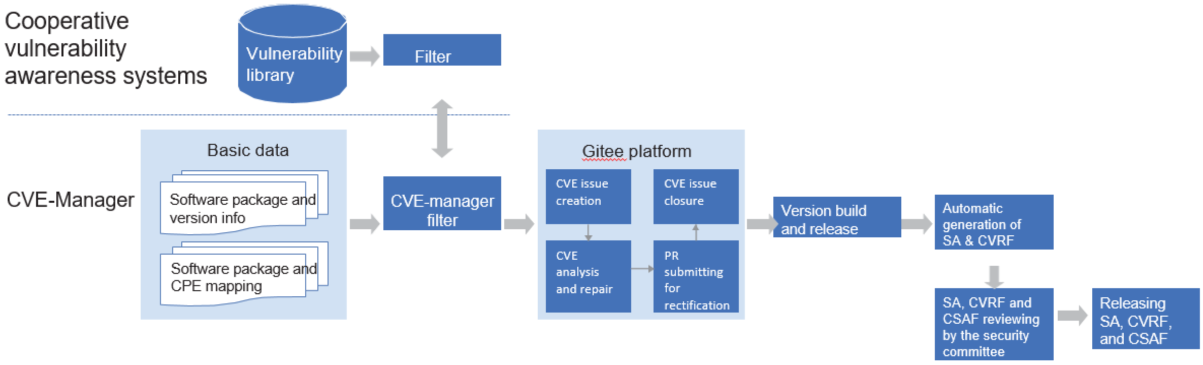
Feature Description

The openEuler community places a high priority on the security of the community edition. The openEuler Security Committee is responsible for receiving, investigating, and disclosing security vulnerabilities affecting the openEuler community. We encourage vulnerability researchers and industry organizations to report suspected security issues to the openEuler Security Committee. We will respond promptly, analyze thoroughly, and remediate reported vulnerabilities.

The bug response process is designed for LTS versions and of the openEuler community and their services packs. The following flowchart shows the E2E vulnerability handling process.



The openEuler security team encourages you to report suspected vulnerabilities in openEuler products to the openEuler community and work with us to remediate and responsibly disclose them. Please email potential security issues to the openEuler security team at openEuler-security@openEuler.org. The security team will respond to reports submitted via email within 48 hours and provide the reporter with updates on handling progress.


Cooperative vulnerability awareness systems
Vulnerability library
Filter
CVE-Manager
Basic data
Software package and version info
Software package and CPE mapping
CVE-manager filter
Gitee platform
CVE issue creation
CVE issue closure
CVE analysis and repair
PR submitting for rectification
Version build and release
Automatic generation of SA & CVRF
SA, CVRF and CSAF reviewing by the security committee
Releasing SA, CVRF, and CSAF

CVE Manager obtains public vulnerability awareness information from cooperative vulnerability awareness systems, and uses the robot to create and maintain vulnerability records in software package repositories on Gitee. After vulnerabilities are fixed, it starts the general version build and release process and then the security notice release process.

openEuler uses CVSS v3 for vulnerability scoring.

For the security of openEuler users, the openEuler community will not discuss, confirm, or disclose the security issues of an openEuler product until the vulnerability is investigated and resolved and the security notice is issued. After a security vulnerability is resolved, the openEuler community will release a security announcement, with information including the technical details, CVE identifier, CVSS security score, and severity level of the vulnerability, as well as the affected and fixed versions. The community also provides security notices in CVRF and CSAF formats that can be subscribed via email.

Application Scenarios

Identify, analyze, fix, and disclose public vulnerabilities in openEuler long-term maintenance releases.

Collect, analyze, fix, and disclose 0-day vulnerabilities in openEuler long-term maintenance releases.

Repositories

https://www.openEuler.org/zh/security/vulnerability-reporting/

https://gitee.com/openeuler/security-committee/blob/master/security-process.md

https://gitee.com/openeuler/cve-manager

Compass-CI

SIG	CICD

Compass-CI is an open-source software platform that can be continuously integrated. It provides developers with test services, login services, auxiliary fault demarcation services, and analysis services based on historical data for upstream open-source software, such as GitHub, Gitee, and GitLab. Compass-CI performs automatic tests (including the build tests and the use case tests included in software packages) based on the open-source software PR to build an open and complete task execution system.

Challenges

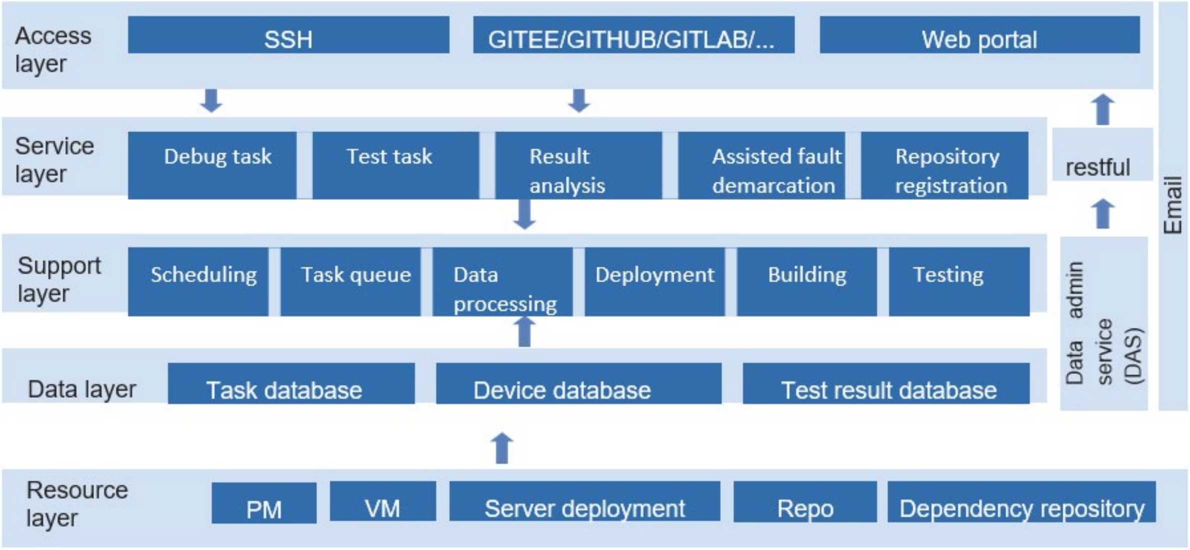
As Linux grows increasingly complex, open-source software developers, constrained by limited resources, often validate only a single scenario. Faced with the many Linux distributions, the primary challenge is how to rapidly integrate, test, and validate open-source software.

Resources are limited, but real-world usage spans a matrix of {OS, architecture, hardware, …} combinations. As a result, many issues are not discovered until later in use, driving up the cost of fixes.

Problems are hard to reproduce, and substantial effort is spent preparing environments, making rapid issue localization difficult.

Feature Description

Compass-CI is a general-purpose, full-stack software testing platform that integrates a build-and-test system, interactive login and debugging, test analysis and comparison, and assisted fault localization. By proactively testing tens of thousands of open-source projects, Compass-CI surfaces issues on processors and operating systems, promptly and automatically pinpoints the root causes, and reports them to third-party developers. This enables timely fixes and helps ensure software quality. Compass-CI provides a friendly development experience for community developers, and works with community developers to flourish the open source software ecosystem and improve the open source software quality.



- Test service: Developers build locally and push code to GitHub. Compass-CI automatically fetches the code, runs tests, and reports results back to the developers.
- Environment login: Compass-CI provides SSH access. If an issue is detected during testing, developers can log in to the test environment to debug as needed.
- Test result analysis: Compass-CI records historical test results and offers both web and CLI interfaces. Developers can analyze existing results and identify factors that impact outcomes.
- Assisted fault locating: During testing, Compass-CI automatically detects error information, triggers git tree–based testing, and pinpoints the change in the affected module that introduced the issue.

Application Scenarios

Aggregating developer test cases: When developers submit code, test cases, and test tools to a code hosting platform, Compass-CI automatically pulls the code for build and test, retrieves test cases embedded in open-source packages for automated execution, and reports the results.

On-demand login and debugging: If a bug is found during testing, Compass-CI can provide debugging resources at any time. Developers can log in to the environment to reproduce and debug the issue.

Snapshot data analysis and comparison: During tests, the system comprehensively monitors runtime metrics (CPU, memory, I/O, network), archives snapshot data, and enables cross-run snapshot comparisons to help analyze results and identify factors impacting outcomes.

Assisted demarcation: When a bug is detected, Compass-CI automatically triggers a regression mechanism to locate the first commit where the issue was introduced.

Repositories

https://gitee.com/openeuler/compass-ci

https://gitee.com/compass-ci/lkp-tests

oepkgs

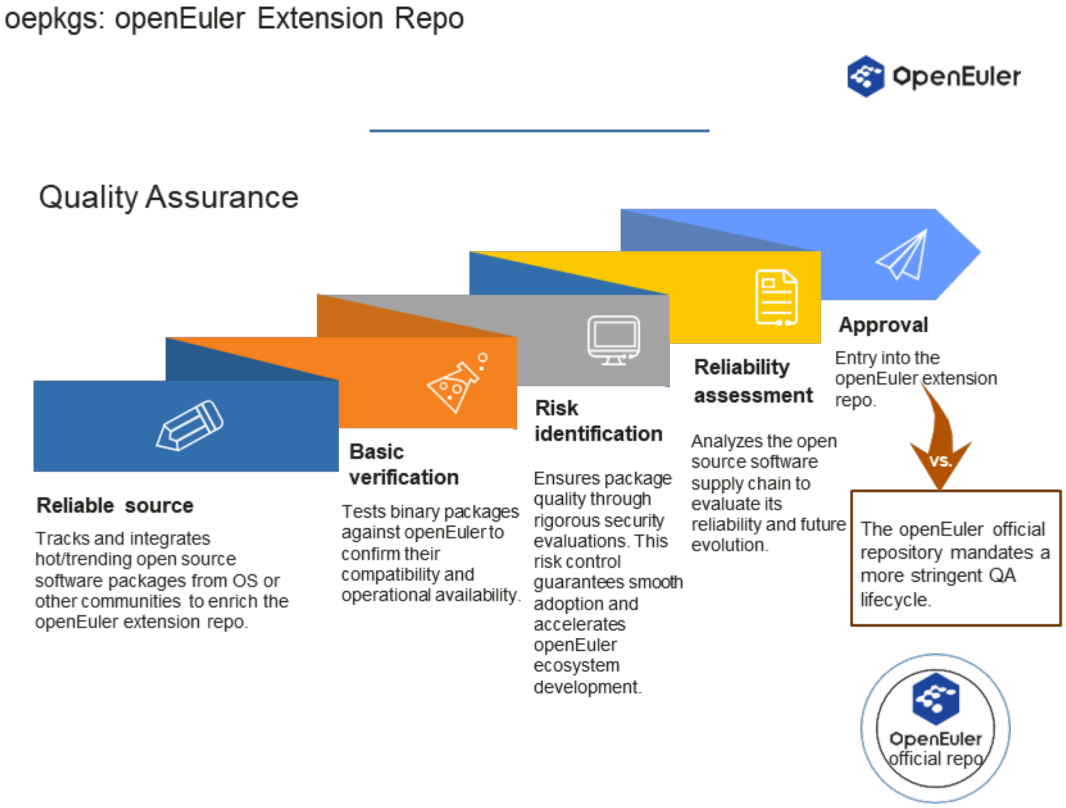
The open external packages service (oepkgs) provides over 30,000 source code packages and millions of binary software packages that are compatible with the openEuler ecosystem. It provides one-stop software package compatibility, file query, and download, as well as open source software package risk detection services for developers, OSVs, and enterprises who are porting from CentOS, Fedora, or other OSs to openEuler.

Challenges

- Accurate and fuzzy retrieval of files and software packages
- Access control for software packages in CI/CD, metadata dependency analysis and management, and flexible verification of software packages
- Risk detection, security, and compliance analysis of open source software

Feature Description

Developed collaboratively by the Institute of Software, Chinese Academy of Sciences (ISCAS), Nanjing Institute of Software Technology (NIST), and openEuler community, oepkgs provides extensive software packages. To ensure software repository quality and continuous evolution, its mature CI/CD system offers a comprehensive lifecycle from source tracing analysis, source code building, and binary scanning, to basic function verification, vulnerability and compliance risk detection, and patch and version update. Furthermore, it delivers powerful capabilities like RPM package retrieval, metadata analysis, SBOM and supply chain analysis, and security and compliance assessment. These features are coupled with one-stop services like file/software package queries, risk queries, and download services to elevate user experience.



oepkgs: openEuler Extension Repo

Quality Assurance

Reliable source

Tracks and integrates hot/trending open source software packages from OS or other communities to enrich the openEuler extension repo.

Basic verification

Tests binary packages against openEuler to confirm their compatibility and operational availability.

Risk identification

Ensures package quality through rigorous security evaluations. This risk control guarantees smooth adoption and accelerates openEuler ecosystem development.

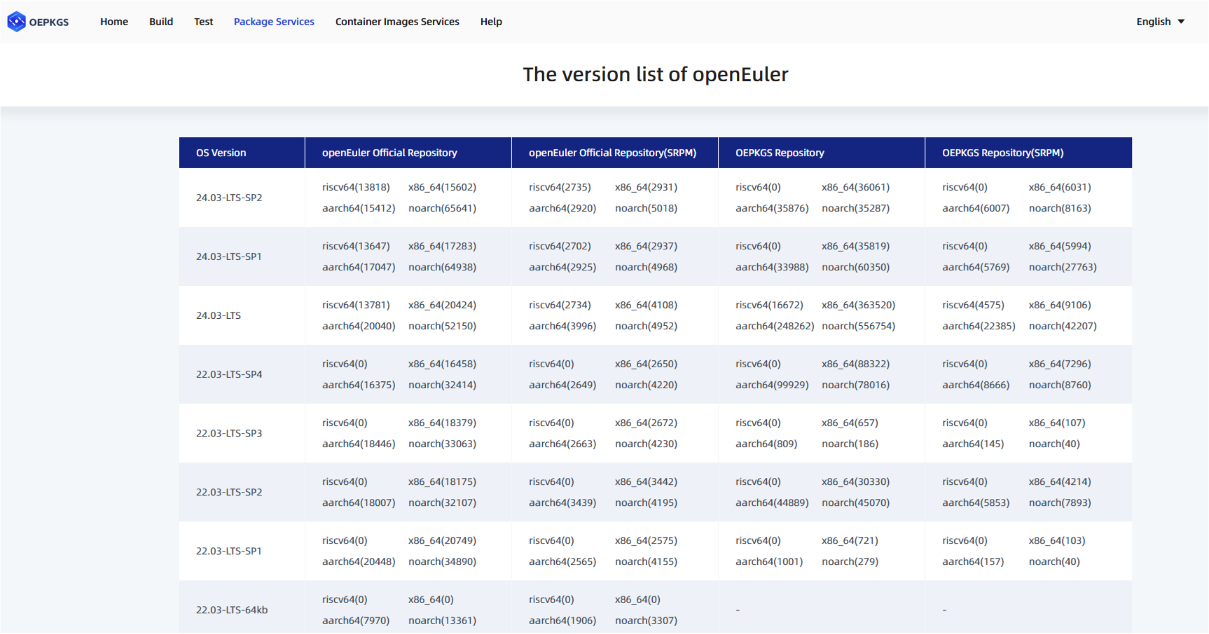
Reliability assessment

Analyzes the open source software supply chain to evaluate its reliability and future evolution.

Approval

Entry into the openEuler extension repo.

The openEuler official repository mandates a more stringent QA lifecycle.



The version list of openEuler

OS Version	openEuler Official Repository	openEuler Official Repository(SRPM)	OEPKGS Repository	OEPKGS Repository(SRPM)

24.03-LTS-SP2	riscv64(13818) x86_64(15602)
	aarch64(15412) noarch(65641)
		riscv64(2735) x86_64(2931)
		aarch64(2920) noarch(5018)
			riscv64(0) x86_64(36061)
			aarch64(35876) noarch(35287)
				riscv64(0) x86_64(6031)
				aarch64(6007) noarch(8163)

24.03-LTS-SP1	riscv64(13647) x86_64(17283)
	aarch64(17047) noarch(64938)
		riscv64(2702) x86_64(2937)
		aarch64(2925) noarch(4968)
			riscv64(0) x86_64(35819)
			aarch64(33988) noarch(60350)
				riscv64(0) x86_64(5994)
				aarch64(5769) noarch(27763)

24.03-LTS	riscv64(13781) x86_64(20424)
	aarch64(20040) noarch(52150)
		riscv64(2734) x86_64(4108)
		aarch64(3996) noarch(4952)
			riscv64(16672) x86_64(363520)
			aarch64(248262) noarch(556754)
				riscv64(4575) x86_64(9106)
				aarch64(22385) noarch(42207)

22.03-LTS-SP4	riscv64(0) x86_64(16458)
	aarch64(16375) noarch(32414)
		riscv64(0) x86_64(2650)
		aarch64(2649) noarch(4220)
			riscv64(0) x86_64(88322)
			aarch64(99929) noarch(78016)
				riscv64(0) x86_64(7296)
				aarch64(8666) noarch(8760)

22.03-LTS-SP3	riscv64(0) x86_64(18379)
	aarch64(18446) noarch(33063)
		riscv64(0) x86_64(2672)
		aarch64(2663) noarch(4230)
			riscv64(0) x86_64(657)
			aarch64(809) noarch(186)
				riscv64(0) x86_64(107)
				aarch64(145) noarch(40)

22.03-LTS-SP2	riscv64(0) x86_64(18175)
	aarch64(18007) noarch(32107)
		riscv64(0) x86_64(3442)
		aarch64(3439) noarch(4195)
			riscv64(0) x86_64(30330)
			aarch64(44889) noarch(45070)
				riscv64(0) x86_64(4214)
				aarch64(5853) noarch(7893)

22.03-LTS-SP1	riscv64(0) x86_64(20749)
	aarch64(20448) noarch(34890)
		riscv64(0) x86_64(2575)
		aarch64(2565) noarch(4155)
			riscv64(0) x86_64(721)
			aarch64(1001) noarch(279)
				riscv64(0) x86_64(103)
				aarch64(157) noarch(40)

22.03-LTS-64kb	riscv64(0) x86_64(0)
	aarch64(7970) noarch(13361)
		riscv64(0) x86_64(0)
		aarch64(1906) noarch(3307)
			-	-

Application Scenarios

- Quickly convert upstream projects into RPM packages using the software introduction platform.
- Enable rapid Linux OS replacement through multi-version package import.
- Query package compatibility with openEuler or identify the source package for any given file.
- Allow enterprise users to detect open source software risks on a unified query platform.
- Provide closed-source software distribution channels to enterprise users for direct downloads of software.

Repositories

https://search.oepkgs.net/en-US

https://gitee.com/src-oepkgs

Developer Tools

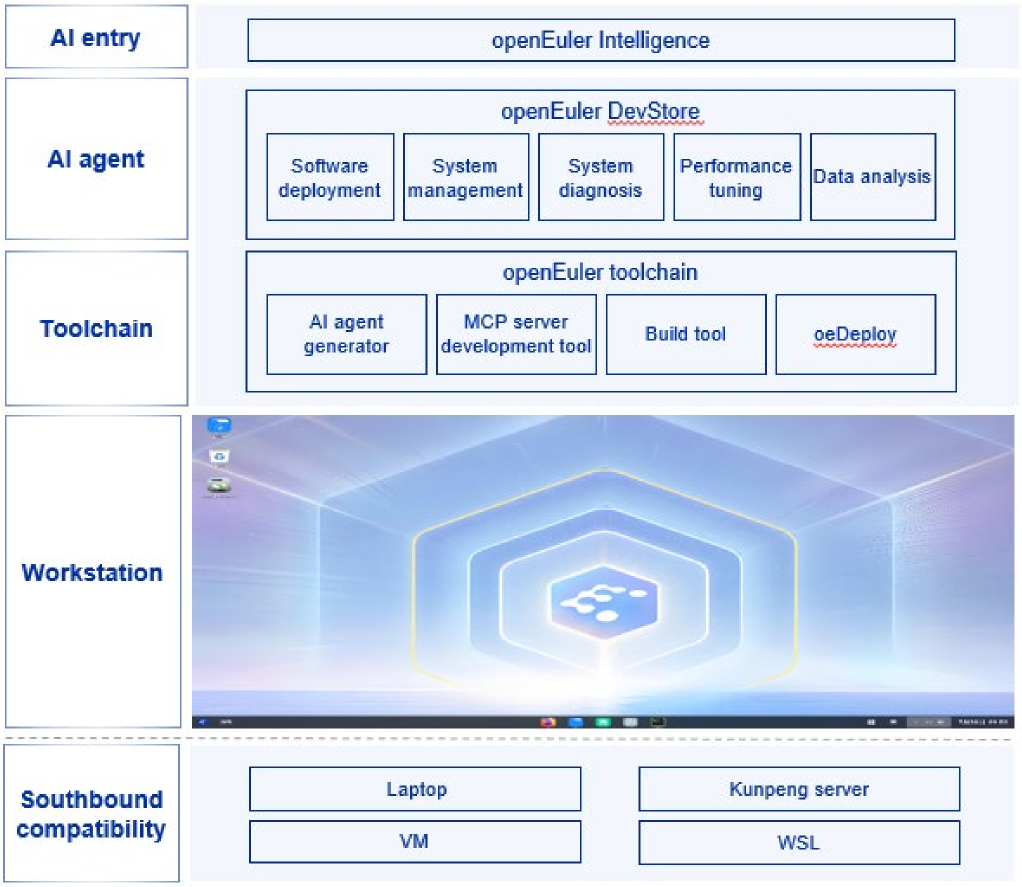
DevStation

SIG	IDE/Intelligence

DevStation is an intelligent developer workstation built on openEuler for geeks and innovators. It provides a secure and ready-to-use development environment that streamlines the entire workflow from deployment and coding to compilation, building, and publishing. With a simple runtime and full-stack development toolchain, it enables a seamless transition from system boot to code execution. Its new MCP AI engine allows users to quickly invoke community toolchains, boosting efficiency from infrastructure setup to application development, without complex installation.

Feature Description

- **Developer-friendly integrated environment**: Pre-installed with a wide range of development tools and IDEs like VSCodium, this distribution supports multiple programming languages to meet the needs of front-end, back-end, and full-stack developers.
- **Native community tool ecosystem**: New tools like oeDeploy (an intuitive deployment tool), epkg (an extended package manager), DevKit, and openEuler Intelligence provide end-to-end support from environment setup to code deployment. oeDevPlugin and oeGitExt are VSCodium plugins designed for the openEuler community, providing visual management for issues and PRs and enabling quick code repository cloning, PR submission, and real-time task status synchronization. openEuler Intelligence uses natural language processing to generate code snippets, create API references, and explain Linux commands.



- **GUI-based programming environment**: DevStation integrates graphical programming tools to streamline coding for beginners while offering powerful visual programming capabilities for advanced developers. It is also pre-installed with productivity tools like Thunderbird.
- **MCP-based intelligent application ecosystem**: DevStation uses the Model Context Protocol (MCP) framework to build a comprehensive intelligent toolchain ecosystem. It includes pre-installed MCP servers like oeGitExt and rpm-builder for managing community tasks and implementing RPM packaging. Crucially, it intelligently wraps conventional development tools like Git and RPM builders using the MCP protocol, offering a natural language interaction interface.
- **Enhanced system deployment and compatibility**: DevStation offers extensive hardware support, especially seamless compatibility with mainstream laptop and PC hardware,

including peripherals like touchpads, Wi-Fi, and Bluetooth. To guarantee an excellent bare metal deployment experience, DevStation features a restructured kernel build script (kernel-extra-modules). Furthermore, it provides flexible deployment options, such as Live CD (instant run without installation), bare metal installation, and VM deployment.

- **New installation tool—heolleo**: heolleo is a modern client tool designed specifically to simplify the DevStation installation process. Built with a modular design, it easily supports feature expansion across various hardware architectures (like x86 and Arm), file systems, and boot loaders (like GRUB). It offers flexible installation modes, supporting system file acquisition from both local ISO images and network sources (HTTP/FTP).
  - Local ISO installation: heolleo provides a local ISO installation mode for users demanding extreme stability, high speed, or deployment in offline or restricted environments. By leveraging existing system image files, it delivers a fast, reliable, and completely offline installation experience with automated partition setup.
  - Network installation: heolleo's network installation mode aligns with modern system deployment trends. It eliminates the need for manual image downloads by allowing you to obtain the latest system files directly from internet servers, which is the most convenient access to the newest DevStation version.

Application Scenarios

- **Multi-language development**: DevStation is ideal for developers working on projects that involve multiple programming languages, such as Python, JavaScript, Java, and C++. With various pre-installed compilers, interpreters, and build tools, it eliminates the need for manual configuration.
- **Quick installation and deployment**: DevStation integrates oeDeploy to deploy distributed software such as Kubeflow and Kubernetes within minutes. oeDeploy provides a unified plugin framework and atomic deployment capabilities, allowing developers to quickly publish custom installation and deployment plugins.
- **Hardware compatibility and bare metal testing**: For testers and developers focused on southbound compatibility, DevStation provides robust hardware support for mainstream laptops and servers, and enables bare metal deployment for driver compatibility testing.
- **Improved developer efficiency**: The MCP RPM-builder toolchain enhances MCP usability by automating the packaging and publishing of RPM packages to the community, ensuring 100% availability for instant MCP installation. It helps build a complete MCP intelligent application ecosystem repository that covers deployment, testing, and performance tuning. This toolchain allows you to query assigned community issues, create PRs to submit code changes, and automate build and verification via CI/CD.

Repositories

https://gitee.com/openeuler/mcp-servers

https://gitee.com/openeuler/heolleo

https://gitee.com/openeuler/oeDevPlugin

https://gitee.com/openeuler/devstation-config

DevStore

SIG	Ops

DevStore is the application store for the openEuler desktop version, acting as a developer-centric software distribution platform. It supports the search and rapid deployment of MCP services and the oeDeploy plugin. DevStore is provided out-of-the-box on the DevStation platform.

Feature Description

- **Rapid installation of MCP services**: Leveraging openEuler community's extensive software ecosystem, DevStore packages the software dependencies required for MCP operation as standard RPM files. Using built-in service management tools, DevStore quickly deploys MCP services in agent applications. It automatically solves software dependency and MCP configuration issues for users, greatly improving user experience. Currently, DevStore supports the deployment of over 80 MCP services.
- **oeDeploy plugin for quick deployment**: DevStore utilizes the oeDeploy tool to enable the rapid deployment of mainstream software, substantially reducing the setup time. The supported software categories include AI software (like Kubernetes, KubeRay, PyTorch, TensorFlow, and DeepSeek), toolchains (like EulerMaker and openEuler Intelligence) and RAG tools (like RAGFlow, Dify, and AnythingLLM).

Application Scenarios

As the application store for the openEuler desktop, DevStore enables developers, including beginners, to swiftly obtain mainstream development tools, AI software, and MCP services. Comprehensive user documents and operation entries are available on the details page. Notably, all complex deployment operations are consolidated into a unified operation interface, significantly reducing the learning cost and improving user experience.

Repositories

https://gitee.com/openeuler/DevStore

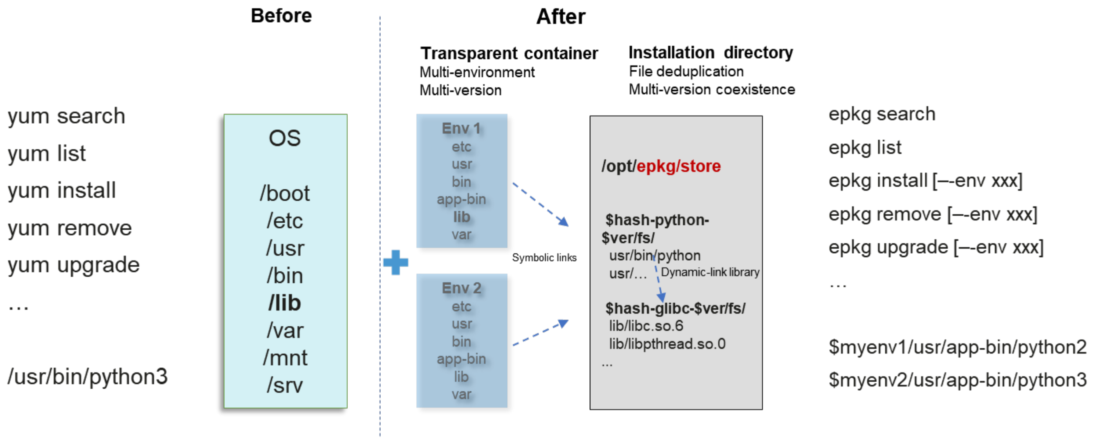
epkg

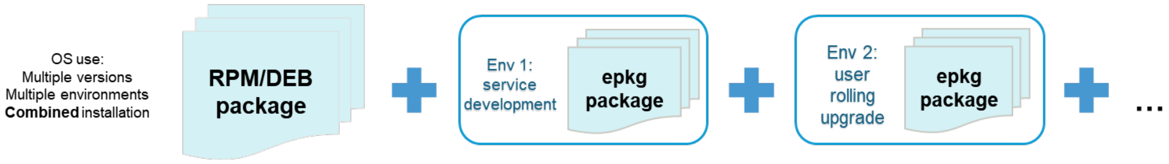
SIG	sig-epkg

epkg is a lightweight and scalable software package management tool provided by openEuler. It offers key features like cross-OS software management, environment isolation, and multi-source collaboration, effectively addressing the challenges of multi-version compatibility. Users can install different software package versions on one OS using simple commands. Additionally, epkg supports comprehensive environment management, such as environment creation, switching, enabling, and rollback. This crucial capability allows users to instantly restore the environment if an error occurs due to an accidental operation or a faulty software installation.

Feature Description

epkg is designed for direct installation on the existing OS. Users can leverage commands such as **epkg env** and **epkg install** to specify the corresponding repository for different environments, install various package versions in each environment, and quickly switch between package versions as needed.



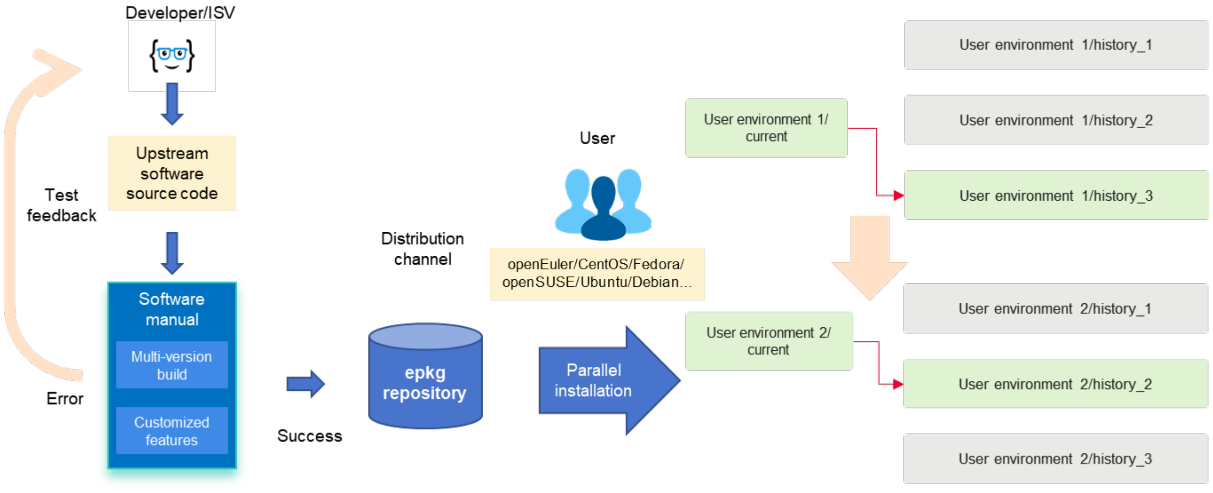


- **Multi-version compatibility**: epkg enables the installation of various software versions without conflict. Specifically, installing different versions of the same software package on the same node does not cause conflicts, allowing users to implement the seamless coexistence of multiple software versions.
- **Environment management**: epkg allows users to implement environment creation, switching, and enabling. By switching between environments, users gain access to distinct package channels, ensuring that the correct package versions are used for each setting. Consequently, when an environment is switched to another, the software package version is also changed.
- **Installation by common users:** epkg allows common users to install software packages, create environments, and manage their environment images, reducing security risks associated with software package installation.

Application Scenarios

For developers: Eliminate the need for OS-specific matrix adaptation, and offer software distribution channels and test feedback.

For users: Provide massive software packages, support multi-environment installation, reproduction, upgrade, and rollback.


Developer/ISV
Upstream software source code
Test feedback
Software manual
Multi-version build
Customized features
Error
Success
Distribution channel
epkg repository
User
openEuler/CentOS/Fedora/ openSUSE/Ubuntu/Debian...
Parallel installation
User environment 1/ current
User environment 2/ current
User environment 1/history_1
User environment 1/history_2
User environment 1/history_3
User environment 2/history_1
User environment 2/history_2
User environment 2/history_3

Repositories

https://gitee.com/openeuler/epkg

QuickIssue

QuickIssue, developed by the openEuler infrastructure team, is an issue tracking system that also excels as a tool for issue classification and submission.

Feature Description

QuickIssue is an efficient, streamlined tool designed for issue submission, boasting the following advantages.

- **Unified submission entry**: Provides a unified entry for issue submission on the openEuler official website, allowing developers to easily locate the correct repository.
- **Flexible user authentication**: Offers alternative methods for submitting issues, accommodating developers regardless of whether they possess a Gitee account.
- **Guided repository selection**: Guides users or developers to the appropriate target repository, and also provides a designated default repository for general submissions.
- **Tailored for openEuler**: Streamlines certain operations on openEuler, including query, search, and filtering.
- **Community integration**: Seamlessly exchanges information with existing community services, including SIG management and contribution statistics.

QuickIssue provides three main functions: creating an issue, querying an issue, and querying a PR.

Creating an issue

- A unified issue submission process ensures all issues in the openEuler community are submitted through one entry.
- Users can submit issues even if they do not have an account on the code hosting platform, typically using an email and verification code method.
- The process is streamlined to help users easily locate the correct target repository, while also providing a designated default repository for general submissions.

Querying an issue

The QuickIssue service provides a comprehensive view of all issues within the openEuler community.

It allows users to filter key information based on their search preferences. To search for issues submitted by email, simply enter the first half of an email address in the **Creator** column to filter the results.

Querying a PR

QuickIssue has access to all PR information in the openEuler community. Users can simply enter the status, creator, and labels to filter the results and the system will display the relevant results. Furthermore, the system uses cached data, which guarantees a fast query response speed.

Application Scenarios

A portal for openEuler developers to submit issues and search for community issues and PRs.

Repositories

https://quickissue.openeuler.org/en/issues/

Compatibility and Technical Assessment

OSV Technical Assessment

The openEuler OSV technical assessment is a standard developed under the guidance of the OpenAtom Foundation to assess OSVs. The assessment is currently being conducted at openEuler Innovation Centers.

Project Introduction

The OSV technical assessment list shows all OS vendors and versions that have been certified according to the community's basic OSV assessment standard.

The OSV technical assessment verifies the consistency of components including OS kernel versions and configurations, KABI, software packages, services, commands, and files. It helps ensure the availability of openEuler-based ecosystem software by evaluating repository reuse (e.g. EPOL/oepkgs) and runtime consistency.

Application Scenarios

This standard assesses the consistency of OSVs' technical roadmaps with openEuler to ensure compatibility with community releases and minimize repeated porting and adaptation.

Further, it ensures that OSV distributions remain forward compatible during development.

To accelerate OS porting and iteration, differential databases are created and integrated into the x2openEuler porting tool.

Repositories

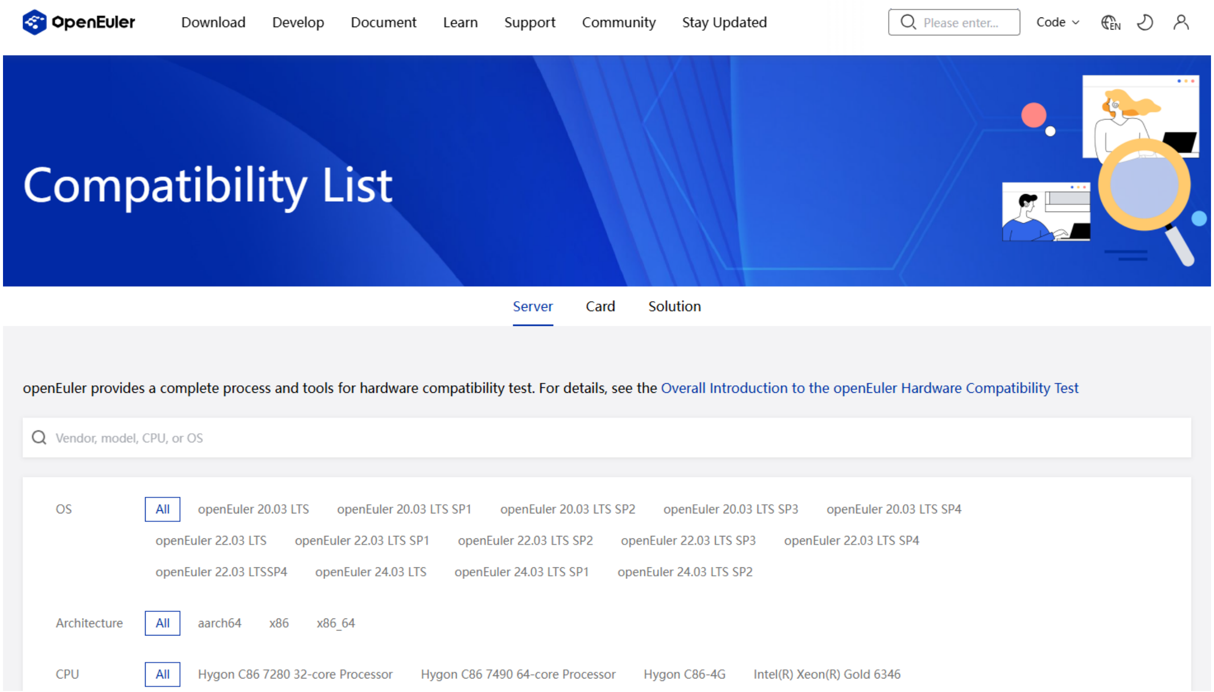
https://gitee.com/openeuler/oecp

openEuler Compatibility List

The openEuler Compatibility List provides a platform for users to query servers, cards, open source software, and commercial software.

Project Introduction

The openEuler Compatibility List provides information about the compatibility of servers, cards, open source software, and commercial software. For more details, visit https://www.openeuler.org/en/compatibility/. You can also query open source software compatibility at https://search.oepkgs.net/en-US.



openEuler provides a complete process and tools for hardware compatibility test. For details, see the Overall Introduction to the openEuler Hardware Compatibility Test

Vendor, model, CPU, or OS

OS: All | openEuler 20.03 LTS | openEuler 20.03 LTS SP1 | openEuler 20.03 LTS SP2 | openEuler 20.03 LTS SP3 | openEuler 20.03 LTS SP4 | openEuler 22.03 LTS | openEuler 22.03 LTS SP1 | openEuler 22.03 LTS SP2 | openEuler 22.03 LTS SP3 | openEuler 22.03 LTS SP4 | openEuler 22.03 LTSSP4 | openEuler 24.03 LTS | openEuler 24.03 LTS SP1 | openEuler 24.03 LTS SP2

Architecture: All | aarch64 | x86 | x86_64

CPU: All | Hygon C86 7280 32-core Processor | Hygon C86 7490 64-core Processor | Hygon C86-4G | Intel(R) Xeon(R) Gold 6346

To facilitate repository creation and continuous evolution for chip and card vendors based on the community infrastructure, openEuler provides a comprehensive set of specifications, processes, and CI/CD pipelines for hardware. This hardware focus encompasses various aspects: CPU architectures, server models, and chip models, along with their respective vendors. openEuler also collaborates with vendors to ensure optimal chip performance, compatibility, compliance, and stability by maintaining functionality, enabling new versions, and adapting to the ecosystem. Companies such as Marvell, NebulaMatrix, and 3SNIC have created repositories in the community and regularly update their versions. For details, visit https://www.openeuler.org/en/compatibility/hardware/.

openEuler integrates open source components according to its package specifications, aligns with upstream projects, and ensures compatibility with mainstream software. In OS porting and upgrade scenarios, openEuler provides a unified platform that allows users to quickly introduce and acquire software packages of the corresponding versions. For details, visit https://www.openeuler.org/en/compatibility/software/.

For ISVs, openEuler provides a testing system for commercial software, including testing specifications, processes, solutions, and toolchains. ISVs can perform tests at the Innovation Centers, with community certificates awarded for standard releases. For details, visit https://certification.openEuler.org/.

Application Scenarios

The openEuler Compatibility List allows users to query compatibility information for CPU architectures, server models, chip models, and their respective vendors, as well as open source and commercial software.

It provides users with access to test schemes, specifications, processes, and tools to ensure that releases are compatible with openEuler standards.

Additionally, IHVs, ISVs, and developers can access specific processes for adding software and hardware to the compatibility list.

Repositories

https://gitee.com/openeuler/oec-hardware

https://gitee.com/openeuler/oec-application

https://gitee.com/openeuler/technical-certification

openEuler Technical Assessment

The openEuler technical assessment is a standard developed under the guidance of the OpenAtom Foundation to assess commercial software, hardware, and OSVs. Currently, this assessment is carried out at openEuler Innovation Centers.

Project Introduction

The openEuler technical assessment runs on the openEuler OS and aims to build a unified ecosystem through standardized device assessment tools and criteria across diverse computing platforms. For hardware and software, components are tested against community specifications to verify compatibility with the openEuler OS, while for OSs, the assessment checks consistency with the openEuler technical roadmap.

Typical OS tests suffer from low efficiency, high costs, and repeated testing due to the absence of an automated, platform- and tool-based standard.

The openEuler ecosystem service platform aggregates diverse computing resources, providing a unified environment for resource management, automated testing, and automatic generation of test reports. It offers the following highlights:

- Unified resource scheduling and automated environment setup, use case testing, and report generation.
- A comprehensive computing hardware resource platform, which is developed with openEuler Ecosystem Innovation Centers to support Kunpeng and x86 architectures, helping reduce computing costs for partners in OS porting, adaptation, and assessment.
- Certifications developed in collaboration with software and server vendors that can certify partners for multiple parties through a single openEuler test, greatly improving the efficiency of building a software partner ecosystem.

Application Scenarios

The compatibility technical assessment defines a unified testing system that verifies solutions and serves as the foundation for a thriving OS ecosystem.

Repositories

https://gitee.com/openeuler/technical-certification

7 Acknowledgments

We extend our sincere gratitude to all contributors of the openEuler community. Through your efforts in coding, documentation, testing, and advocacy, openEuler continues to grow stronger and more open.

Trademark

All trademarks, product, service, and company names mentioned in this document are the property of their respective owners.

Disclaimer

This document may contain predictive information, including but not limited to information about future finance, operations, product series, and new technologies. There are a number of factors or developments that could cause actual results to differ materially from those expressed or implied in the forward-looking statements. Therefore, the information in this document is for reference only and does not constitute any offer or commitment. openEuler is not liable for any behavior that you make based on this document. openEuler may change the information at any time without notice.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of openEuler.