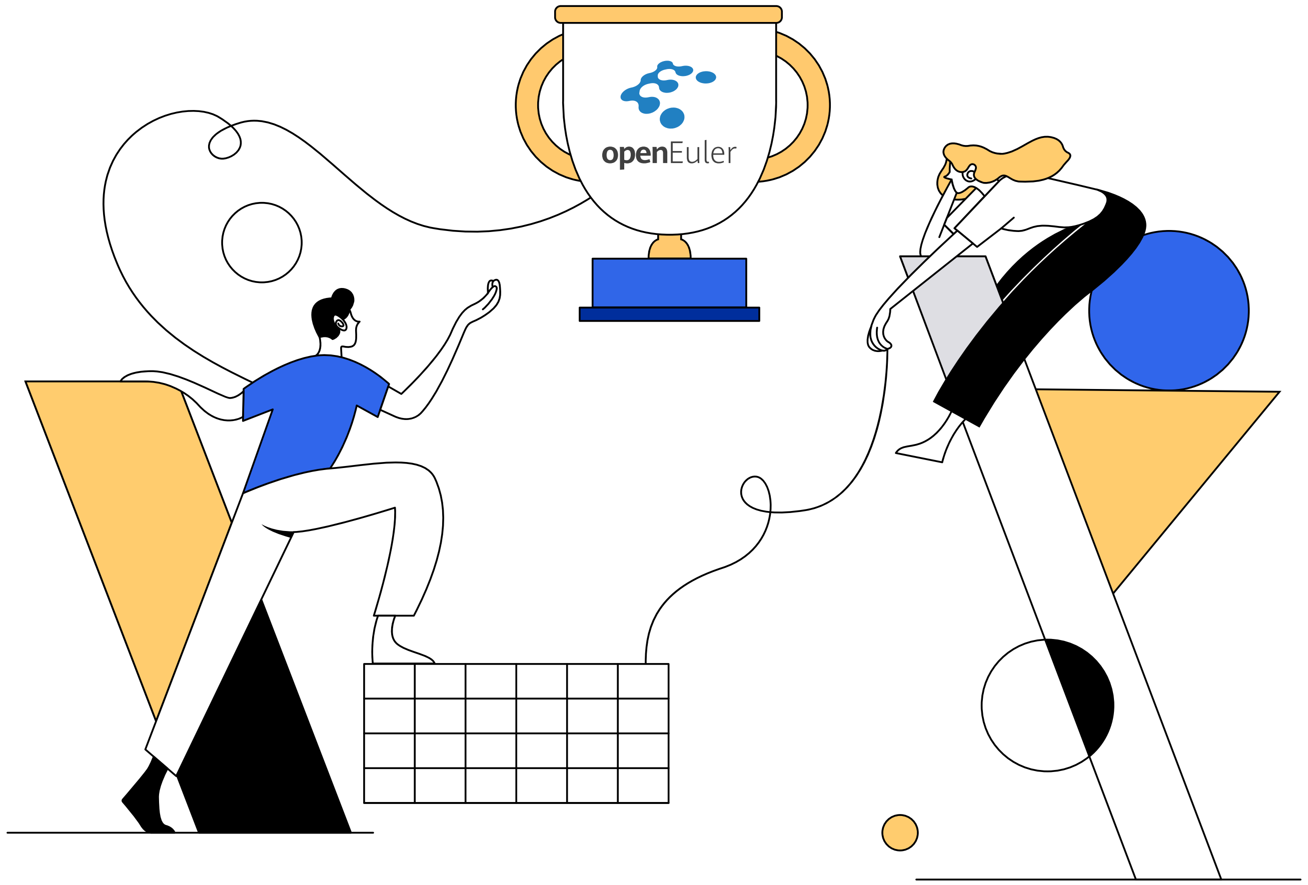
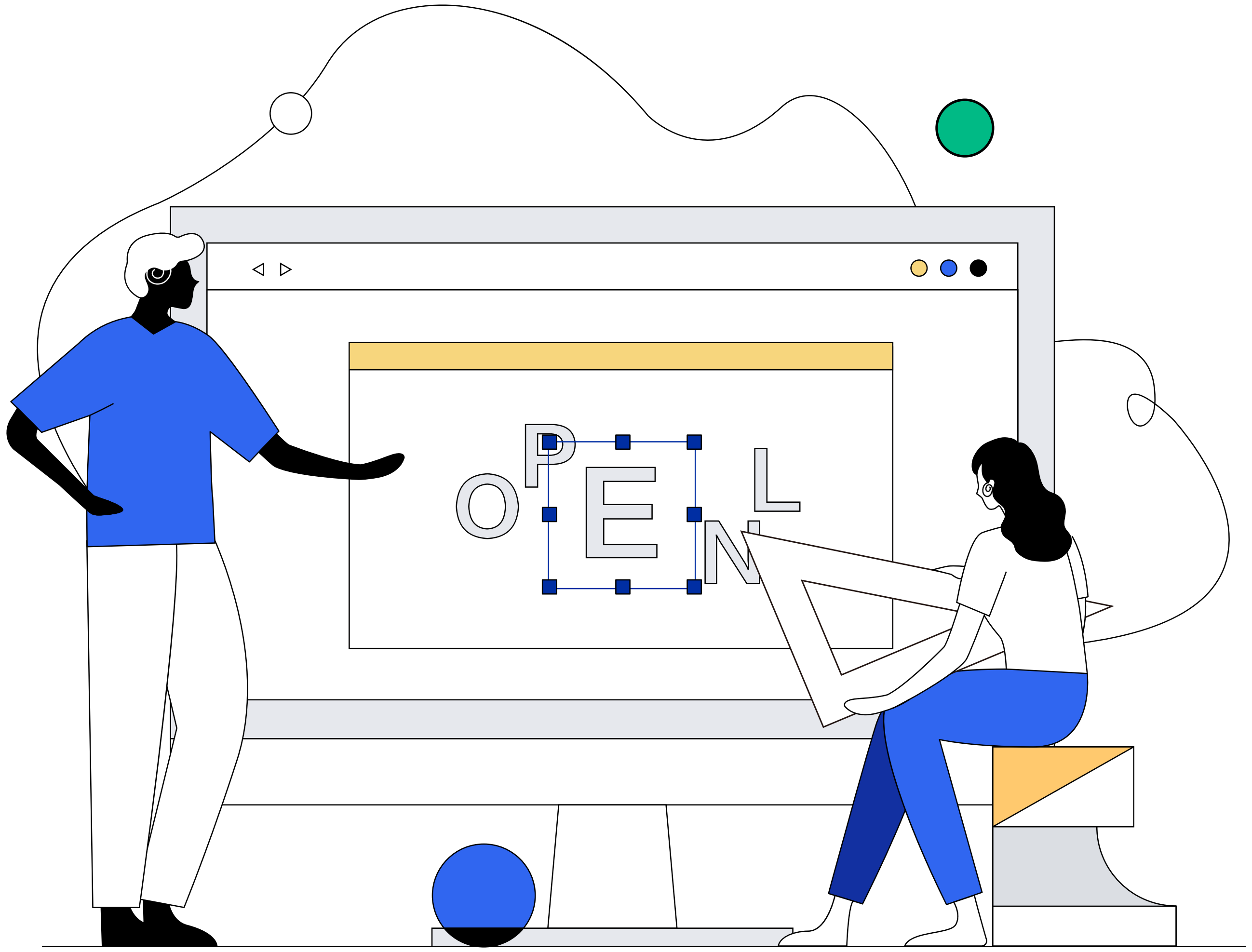
openEuler 21.03

Technical White Paper



CONTENTS

Introduction



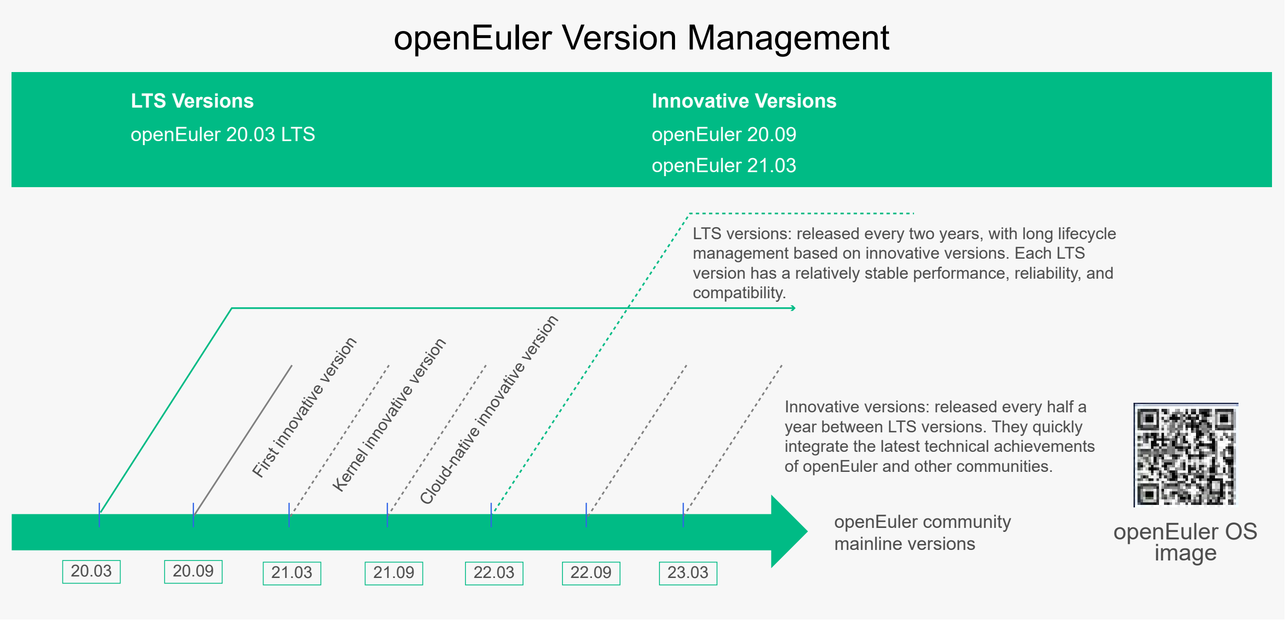
openEuler is an open source Linux release platform. In the openEuler community, global developers join hands to build an open, diversified, and architecture-inclusive software innovation ecosystem. openEuler is an innovative platform driven by community collaboration. It aims to build a unified and open OS that supports multiple processor architectures, and to advance the hardware and software application ecosystem.

On December 31, 2019, the openEuler open source OS community was officially established. It was initiated for diversified computing architectures.

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

On September 30, 2020, the innovative version openEuler 20.09 was released. This version was the product of collaboration between multiple companies, teams, and independent developers in the openEuler community. The release of openEuler 20.09 is a milestone in the growth of the openEuler community and the history of open source in China.

On March 31, 2021, the innovative version openEuler 21.03 was released. This version not only switched to Linux kernel 5.10 but also incorporated multiple new features such as live kernel upgrade and tiered memory expansion. These highlights improve multi-core performance and deliver thousand-core computing power.



openEuler Version Management

LTS Versions
openEuler 20.03 LTS

Innovative Versions
openEuler 20.09
openEuler 21.03

LTS versions: released every two years, with long lifecycle management based on innovative versions. Each LTS version has a relatively stable performance, reliability, and compatibility.

Innovative versions: released every half a year between LTS versions. They quickly integrate the latest technical achievements of openEuler and other communities.

First innovative version
Kernel innovative version
Cloud-native innovative version
20.03
20.09
21.03
21.09
22.03
22.09
23.03
openEuler community mainline versions
openEuler OS image

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

openEuler is a technology incubator. It releases an innovative version every 6 months to quickly integrate the latest technical achievements of openEuler and other communities. After being verified in the openEuler open source community, the latest features are added to each new release. Each of the features exists in the openEuler community as a single open source project, enabling openEuler developers to obtain the source code and other open source communities to integrate.

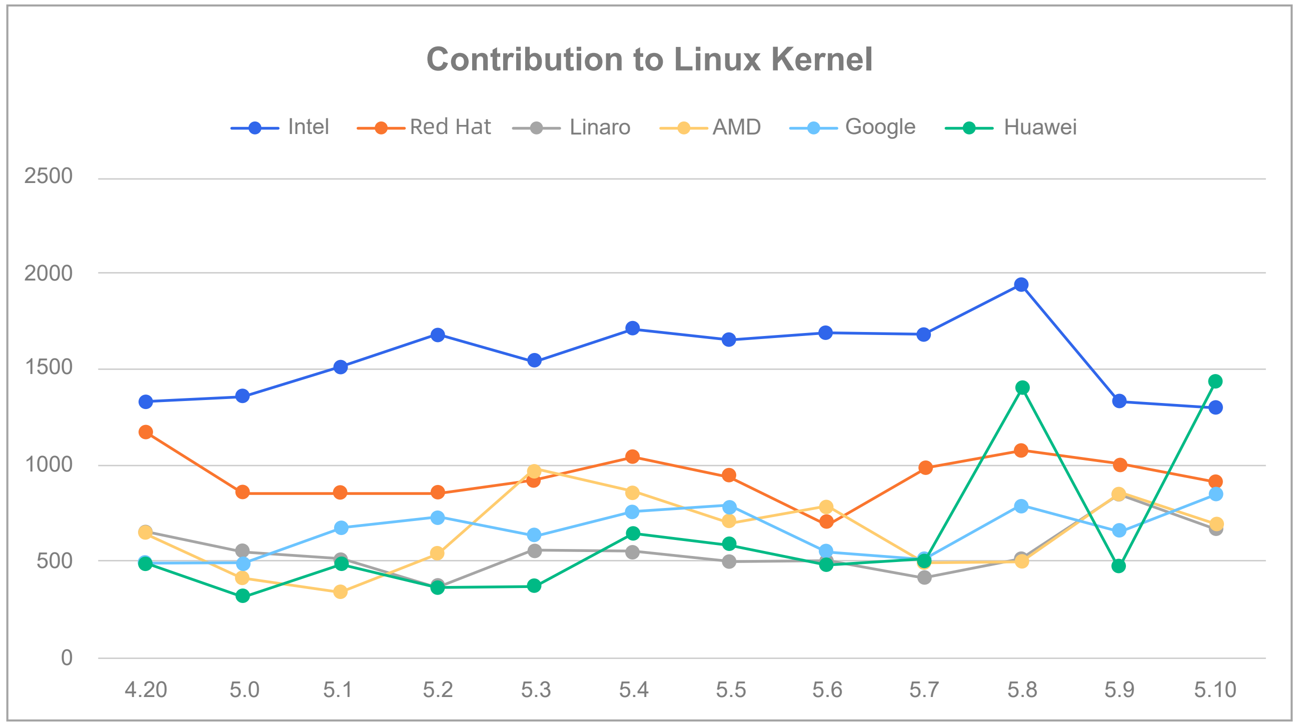
The latest technical achievements in the open source communities are continuously incorporated into each of the subsequent openEuler releases. Users give feedback on these releases to stimulate the innovation vitality in the openEuler community and incubate more technologies. openEuler plays the roles of a release platform and a technology incubator, which server to advance each other and drive the continuous version evolution of new versions.

Innovation for Diversified Computing Power

openEuler supports multiple processor architectures including x86, ARM, and RISC-V. It is an OS that drives continuous innovation in diversified computing architectures. openEuler supports the x86 architecture represented by Intel and Zhaoxin and the ARM architecture represented by Kunpeng and Phytium. In addition, an RISC-V trial edition of openEuler has been released. openEuler is intended to work with users and developers to improve the performance experience of diversified computing power as well as advance the ecosystem development.

openEuler tries to match each type of workload with the most appropriate computing power unit, improve the parallel processing capability through software optimization, and unleash the full power of diversified computing architectures.

Continuous Contribution to the Linux Kernel



Contribution to Linux Kernel

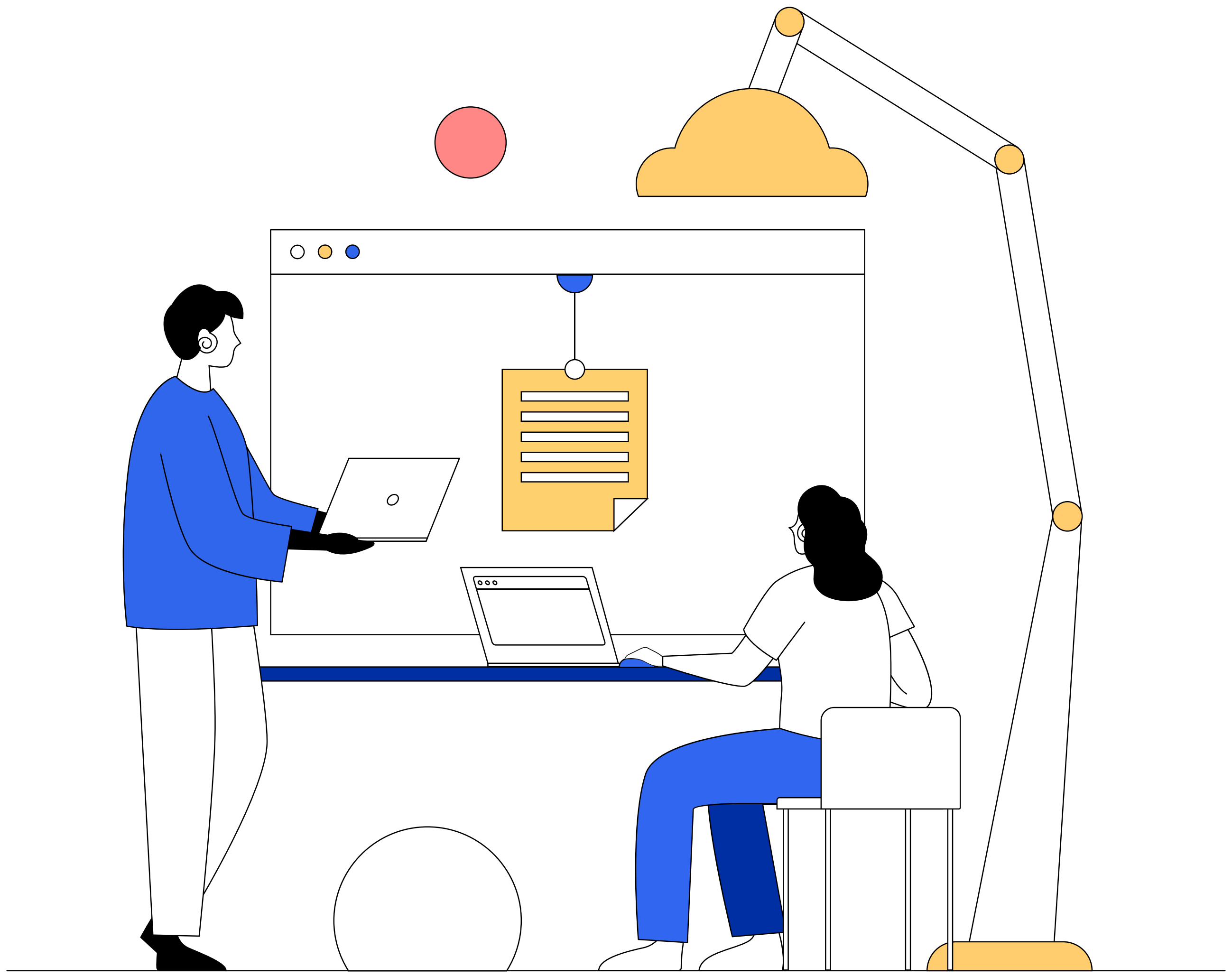
As a major contributor to the Linux kernel, Huawei has put a great deal of effort into advancing the ARM64 architecture, Advanced Configuration and Power Interface (ACPI), memory management, file systems, media, kernel documents, bug fixes, and code rebuild. Over the past decade, Huawei has contributed more than 13,000 patches to the Linux kernel.

In Linux kernel 5.10, Huawei's code contribution ranks No. 1 in the world. Huawei is committed to kernel innovation and has been continuously contributing to upstream communities.

Open and Transparent Management of the Open Source Software Supply Chain

The process of building an open source OS is also a process of supply chain aggregation and optimization. A reliable open source software supply chain is fundamental to a large-scale commercial OS. openEuler reviews its software dependencies based on real user scenarios, sorts out the upstream community addresses of all the software packages, and verifies its source code by comparing it to that of the upstream communities. This is a complete lifecycle management that covers building, 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.

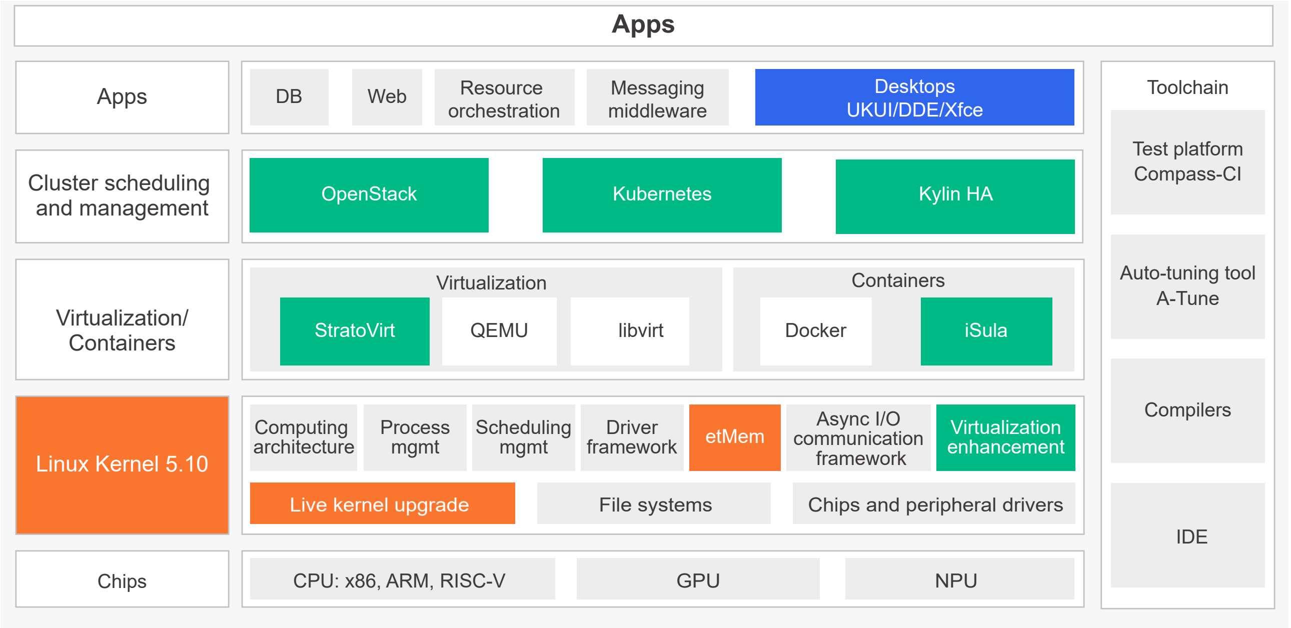
Platform Architecture



System Framework

openEuler is used mainly for servers. It consists of the basic acceleration library, virtualization, kernel, driver, compiler, system tool, OpenJDK, and other components.

As the one OS for all scenarios, openEuler has an innovative architecture and full-stack optimizations to unleash the full computing power of diversified architectures.



Apps

Apps: DB, Web, Resource orchestration, Messaging middleware, Desktops UKUI/DDE/Xfce

Cluster scheduling and management: OpenStack, Kubernetes, Kylin HA

Virtualization/ Containers: Virtualization (StratoVirt, QEMU, libvirt); Containers (Docker, iSula)

Linux Kernel 5.10: Computing architecture, Process mgmt, Scheduling mgmt, Driver framework, etMem, Async I/O communication framework, Virtualization enhancement; Live kernel upgrade, File systems, Chips and peripheral drivers

Chips: CPU: x86, ARM, RISC-V, GPU, NPU

Toolchain: Test platform Compass-CI, Auto-tuning tool A-Tune, Compilers, IDE

Kernel innovations:

- **Linux kernel 5.10:** In-depth optimizations for scheduling, I/O, and memory management
- **Tiered memory expansion etMem:** unified management of various memory and storage media, and smooth expansion of system capacity
- **Live kernel upgrade:** quick, hitless fixes of kernel vulnerabilities

Cloud base:

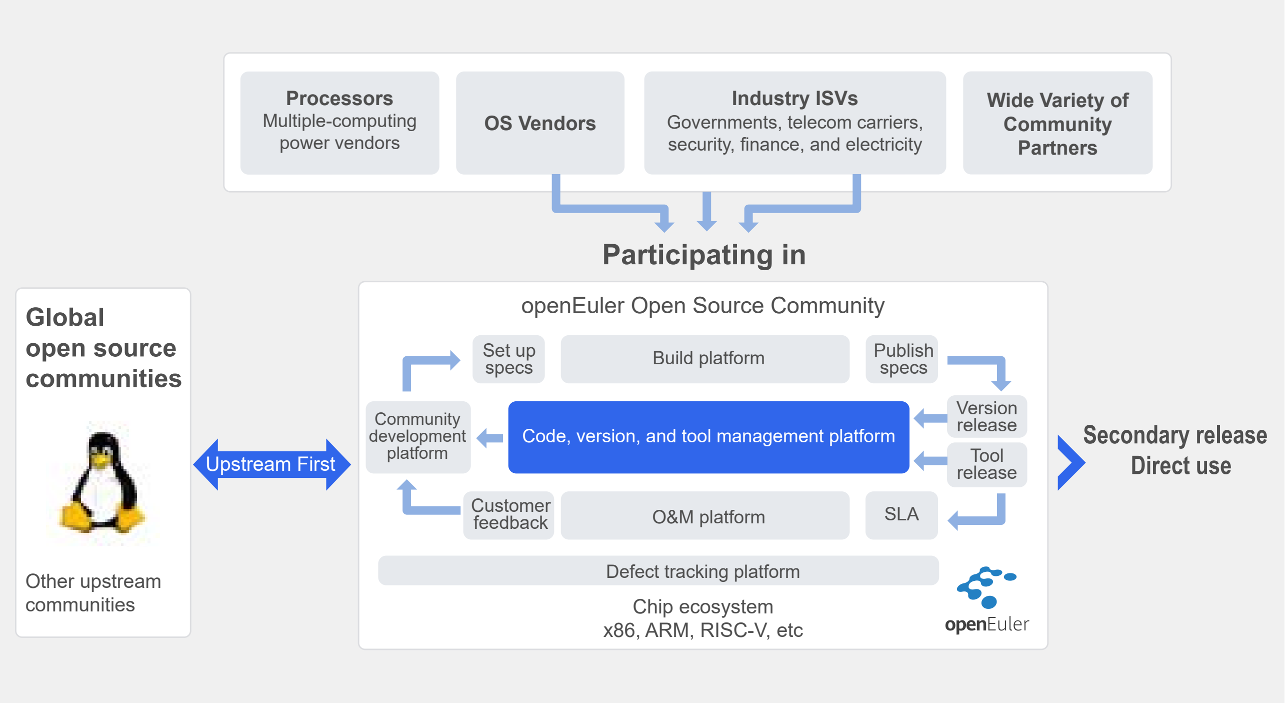
- **iSula:** iSulad supports local volume management. isula-build incorporates functions of image pulling and pushing.
- **StratoVirt & virtualization:** openEuler supports memory elasticity, huge pages, and enhanced I/O subsystems, and leverages multi-channel concurrency to improve I/O performance.
- **OpenStack & Kubernetes:** openEuler is designed for cloud applications. It integrates the two mainstream pieces of cloud scheduling and management software to build a solid cloud base.
- **High availability (HA) cluster solution:** The HA cluster solution implemented by KylinSoft enables failover within seconds.

Flourishing community ecosystem:

- **Desktop environments:** openEuler supports UKUI, DDE, Xfce, and other desktop environments.

Platform Framework

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


Processors
Multiple-computing power vendors
OS Vendors
Industry ISVs
Governments, telecom carriers, security, finance, and electricity
Wide Variety of Community Partners
Participating in
Global open source communities
Other upstream communities
Upstream First
openEuler Open Source Community
Set up specs
Build platform
Publish specs
Community development platform
Code, version, and tool management platform
Version release
Tool release
Customer feedback
O&M platform
SLA
Defect tracking platform
Chip ecosystem
x86, ARM, RISC-V, etc
openEuler
Secondary release
Direct use

Operating Environment

Physical Machines

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

For details about hardware compatibility, see the openEuler compatibility list at https://openeuler.org/en/compatibility/.

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

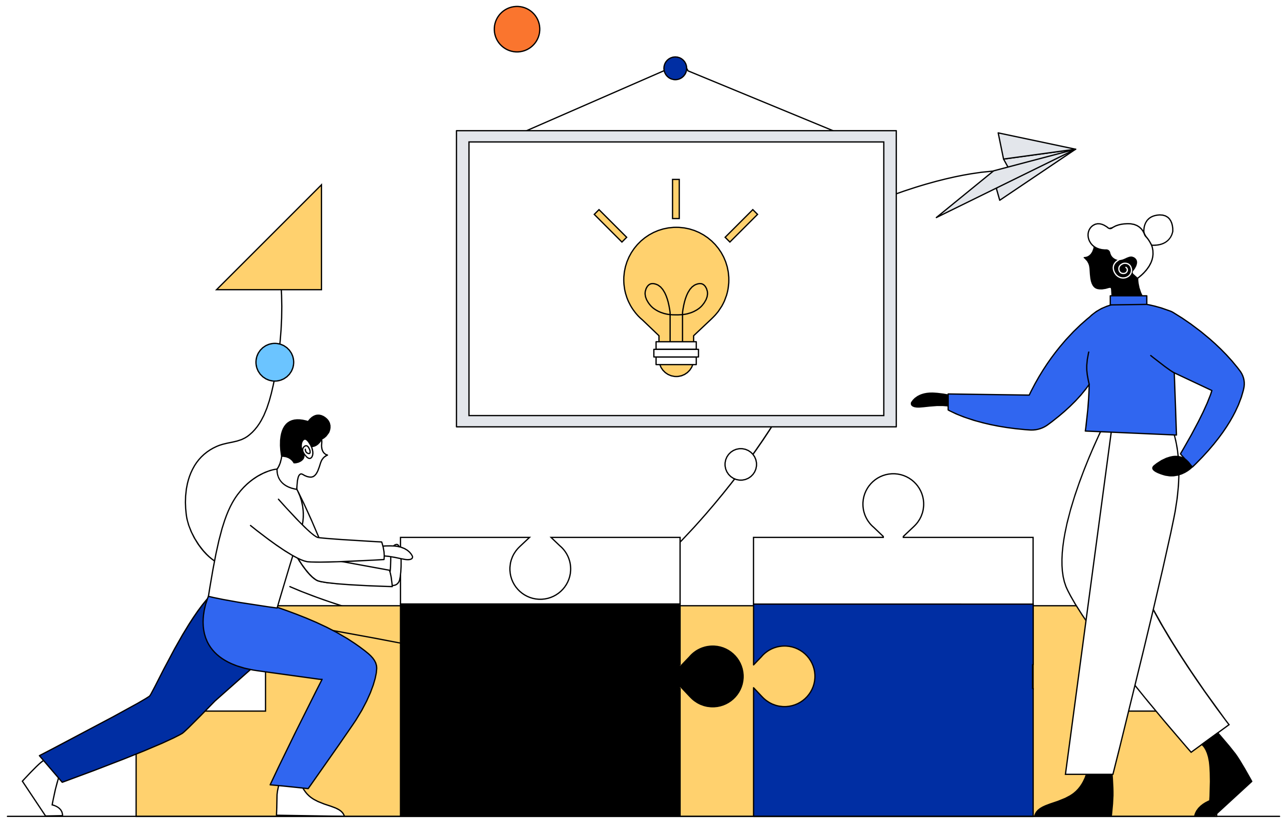
Virtual Machines

openEuler supports the following virtual machines (VMs):

1. centos 7.9 qemu 1.5.3-175.el7 libvirt 5.0.0-1.el7
2. centos 8 qemu 5.1.0-20.el8 libvirt 6.6.0-7.3.el8
3. fedora 29 qemu 3.0.0-1.fc29 libvirt 4.7.0-1.fc29
4. fedora 32 qemu 4.2.0-7.fc32 libvirt 6.1.0-2.fc3

Item	Configuration Requirement
Architecture	AArch64, x86_64
CPU	2 CPUs
Memory	At least 4 GB
Hard drives	At least 20 GB

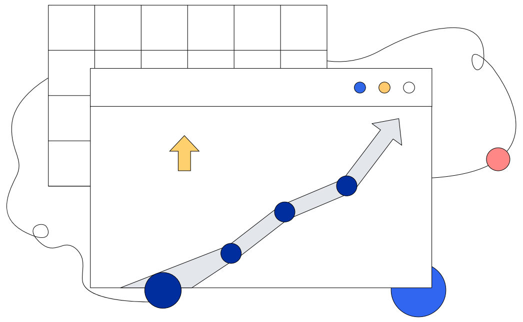
Kernel Innovations



New Technologies in the openEuler Kernel

openEuler 21.03 is built based on Linux kernel 5.10, and introduces more than 20 enhancements in terms of functionality and performance.

1. Scheduler optimization: The optimized fairness of Completely Fair Scheduler (CFS) tasks and the NUMA-aware asynchronous call mechanism combine to bring a significant improvement in NVDIMM initialization. Optimized scheduling policies of SCHED_IDLE significantly shorten the scheduling delays of prioritized tasks and reduce interference to other tasks. openEuler 21.03 also has an optimized NUMA balancing mechanism, which brings better affinity, higher utilization, and fewer invalid migrations.
2. Enhanced CPU isolation: The isolation of interrupts and unbound kthreads further enhances the isolation of CPU cores and minimizes mutual interference between services.
3. Inter-process communication optimization: The optimized pipe_wait and epoll_wait wakeup mechanisms improve the performance of waking up multiple waiting threads.
4. Enhanced memory management: openEuler 21.03 offers refined functions of memory initialization, memory control, statistics, heterogeneous memory, and hot swap, as well as more effective user control interfaces. A series of optimizations in hotspot lock, semaphore, radical memory, defragmentation, VMAP, and vmalloc significantly improve the memory application efficiency. Enhanced memory maintenance and test features, including KASAN, kmemleak, slub_debug, and out of memory (OOM), reduce the difficulty in locating and resolving memory problems.
5. Single-thread porting performance optimized for control groups (cgroups): openEuler 21.03 eliminates the dependence on the read and write semaphores of thread groups. It introduces time namespaces to facilitate container porting.
6. Limited number of file handles used in a container: File handles include common file handles and network sockets. When starting a container, you can set the --files-limit parameter to specify the maximum number of handles that can be opened in a container.



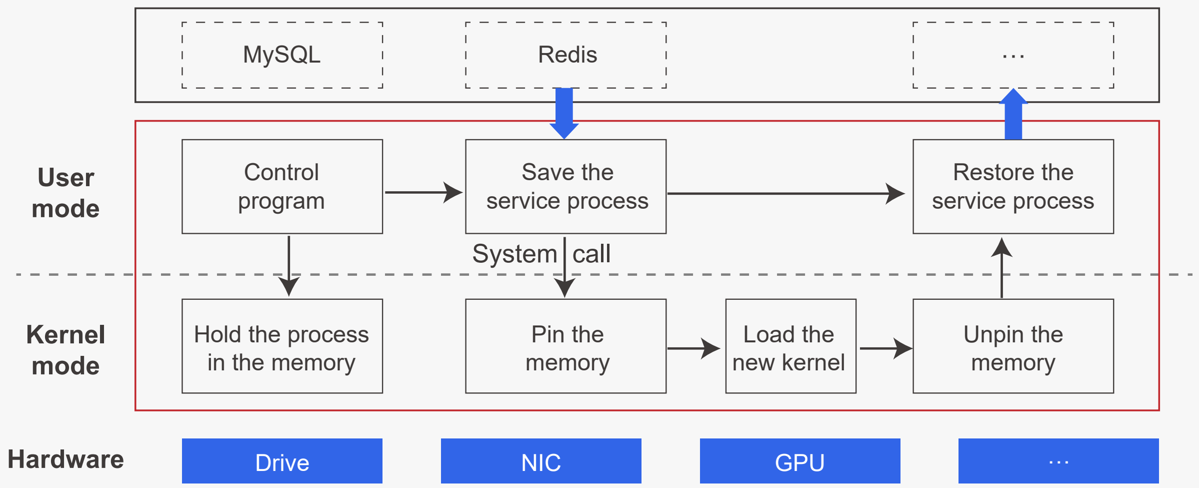
7. Pressure Stall Information (PSI) capability: This is a method for evaluating the pressure of system resources such as CPU, memory, and I/O. An accurate detection method helps resource users determine a proper workload and enables the system to apply an efficient resource scheduling policy. The benefits include maximized system resource utilization and improved user experience.
8. Early Departure Time (EDT) model for sending TCP packets: It breaks the limitation of the TCP framework. This model adds an EDT timestamp to each data packet based on the scheduling policy, avoiding the delays caused by large queue buffers and bringing about a substantial improvement in TCP performance.
9. MultiPath TCP: It improves performance and reliability in mobile and data scenarios. MultiPath TCP supports concurrent transmission of multiple substreams in load balancing scenarios.
10. A new and more lightweight logging method in Ext4: - fast commit. It greatly speeds up time-consuming operations such as fsync and delivers a better performance.
11. dm-writecache: It improves the sequential write performance of large solid-state drive (SSD) blocks and the performance of the double data rate (DDR) persistent memory.
12. io_uring: This is a new asynchronous I/O framework that supports the polling mode. In polling mode, io_uring brings about a substantial performance improvement, letting io_uring deliver a performance close to SPDK. io_uring is more effective when the queue depth is relatively high.
13. Enhanced commercial use of Integrity Measurement Architecture (IMA): Based on the open source IMA solution, the security, performance, and usability are enhanced to facilitate commercial use.
14. Per-task stack inspection: strengthens protection against Return Oriented Programming (ROP) attacks.
15. ILP32 support: The Kunpeng 920 ARM64 environment supports 32-bit applications.
16. Memory System Resource Partitioning and Monitoring (MPAM): Cache QoS and memory bandwidth control can be applied on the ARM64 architecture.
17. NMI mechanism based on the SEDI and PMU: This mechanism enables hard lockup detection. Additionally, it enables perf nmi to perform more accurate performance analysis.
18. VM hot swap: CPU hot swap can be enabled in the ARM64 environment to obtain more flexible resource configurations.
19. Enhanced ARM64 kdump: Memory reservation for addresses greater than 4 GB enables kdump to support machines with larger memory.
20. Raspberry PI: The openEuler 21.03 kernel natively supports the Raspberry PI, and it can be used for source code debugging on the Raspberry PI.
21. RISC-V: The RISC-V platform supports kernel-based virtual machines (KVMs).
22. Hardware: Hi1822 iNICs are supported.

Live Kernel Upgrade

A live kernel upgrade, including CVE vulnerability fixes and security kernel replacement, does not interrupt ongoing services. Before a live kernel upgrade, the system checks the resource usage status. The system uses a quick freezing technology to freeze the status of the resources occupied by services, uses a quick loading technology to load the new kernel, and then restores the frozen resource status.

To ensure that services are not interrupted when rectifying kernel errors, the hot patch technology comes into play. However, hot patches can resolve only 20% of problems due to their limitations. For example, they are unable to change data structures, rectify inline functions, or rectify complex logic problems. Hot patches also may complicate data center O&M. For example, the O&M baseline is hard to set up when there are a lot of patches installed. The live kernel upgrade technology has emerged as the solution to these problems.

Feature Description



- Save and restore process-level status quickly
- Reset the system quickly without peripheral correlation

1. Control program: It takes command of the whole live kernel upgrade process, exchanges kernel-mode patch information and user-mode daemon information, performs the live kernel upgrade for specified services, and if the upgrade fails, rolls back the kernel to the original version.
2. Save the service process: Use system checkpoints to save the service process status and resource status, ensuring the consistency between the two statuses.
3. Load the new kernel: Use the system Kexec mechanism to quickly load the new kernel, recovering end-to-end services within seconds.
4. Restore the service process: Use the System Restore feature to restore the saved service status and resources.

Application Scenarios

Application scenario 1: kernel CVE fixes
Typical applications, such as Nginx, Redis, and MySQL, run on a physical or virtual machine. They generate many keep-alive connections and occupy a large amount of memory. When the physical or virtual machine has a serious CVE in the kernel, to fix this CVE, there are three countermeasures available to fix this CVE:

1. Restarting the host: Users who have logged in to the host will perceive the service interruption. In addition, the service interruption is unacceptably long because the restart takes a long time.
2. Service migration: Typical services occupy a large amount of memory, so the entire memory must be migrated. As a result, the service performance remains deteriorated for a long time.
3. Hot patches: They cannot resolve all of the problems and cannot even be installed for some of the problems. As an alternative solution, a live kernel upgrade can fix kernel bugs while ensuring service continuity and maintaining the performance level.

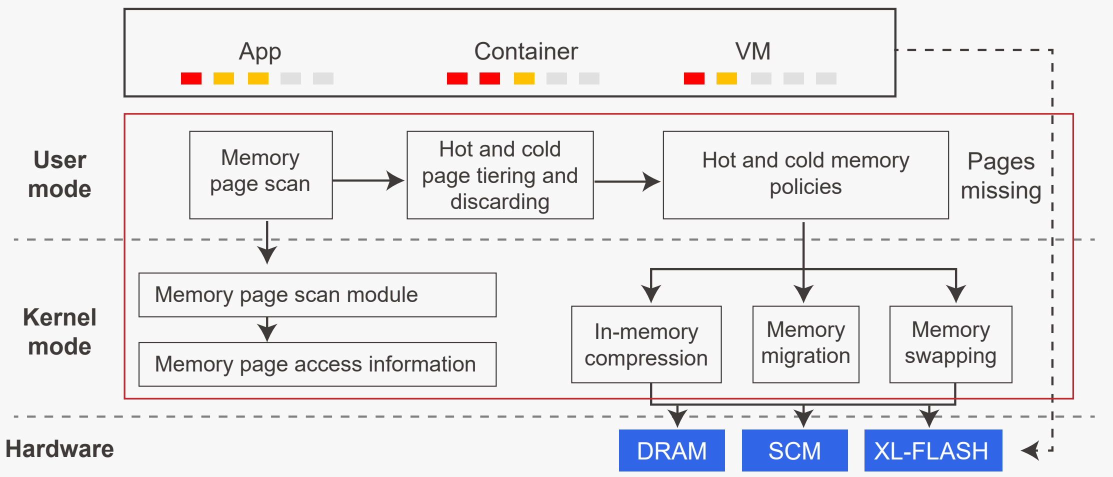
Application scenario 2: minor-release kernel upgrade
Throughout the OS lifecycle, new features must be introduced to meet growing customer requirements. New kernel features have a great impact on services in performance, security, and commissioning. To introduce a new feature, we may need to restart the service and host. As a result, the service client will be aware of the service interruption. To avoid the service interruption, we can choose the live kernel upgrade solution, which incorporates new kernel features without users being made awareness.

Tiered Memory Expansion etMem

Today's memory manufacturing processes have nearly reached the peak of how advanced they are. The mature ARM ecosystem is making the cost per CPU core lower and lower. Databases, VMs, big data, AI, and deep learning call for increasingly higher computing power and memory capacities. Limited memory capacities have become a challenge for service growth.

Tiered memory expansion has proven itself as a solution for that challenge. DRAM and low-speed memory media, such as storage class memory (SCM) and Apache Pass (AEP), form a multi-tier memory structure. Automatic memory scheduling redirects hot data to the DRAM high-speed memory area and cold data to the low-speed memory area. The tiered memory structure increases the memory capacity and ensures efficient and stable running of core services. etMem is ideal for applications that use a large amount of memory but do not access the memory frequently.

Feature Description



- Accurate identification of hot and cold pages; in-service automatic swapping
- Configurable process-level policy control

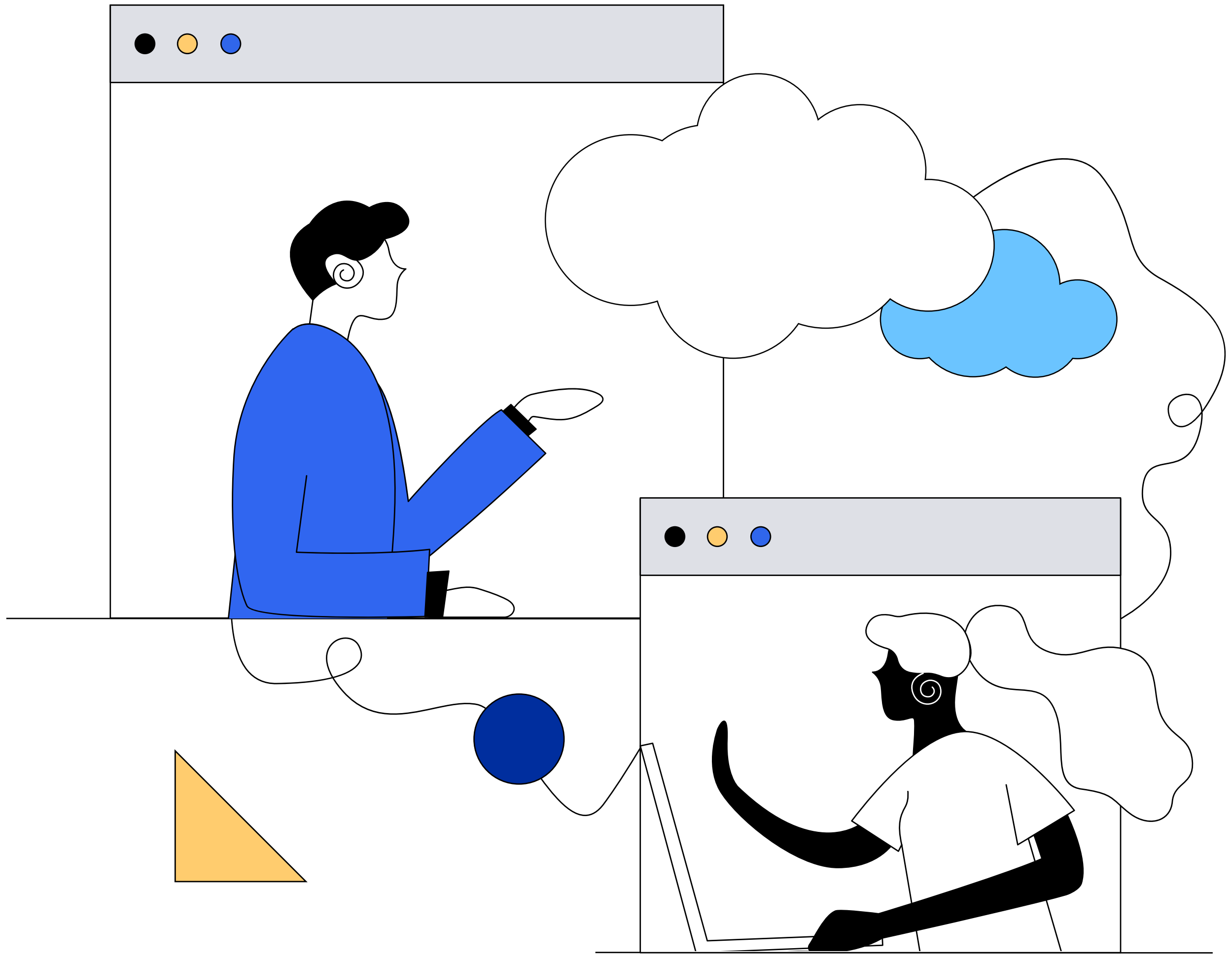
1. **Process-level control:** etMem supports processes that use configuration files to expand the memory. Compared with the native LRU-based pageout kswap mechanism of the OS, etMem is more flexible and accurate.
2. **Memory scanning:** The new kernel function is triggered by the user-mode etMem process. It scans for the memory access of a specified process and returns the scan result.
3. **Cold and hot tiering:** You can select a cold and hot tiering policy configuration file to classify obtained memory access results into hot memory and cold memory.
4. **Discarding policy:** The cold memory is discarded when it meets the conditions specified in the etMem configuration file and the system environment configuration. The discarding process uses the native kernel capability, which is secure and reliable and does not affect user experience.

Application Scenarios

Application scenario: tiered memory expansion for service processes on a node

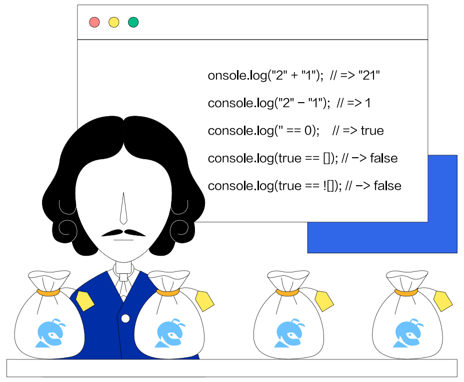
etMem is ideal for applications that use a large amount of memory but do not access the memory frequently, such as MySQL, Redis, and Nginx. All memory expansion operations are performed within a node and no cross-node operations are involved.

Cloud Base



Container Infrastructure iSula

iSula is a container technology project of openEuler. It includes multiple pieces of software in the full-stack container ecosystem. Compared with Docker, which is compiled using Golang, iSulad is a new universal container engine that meets different requirements in the CT and IT fields through a unified architecture. The lightweight container solution is implemented using C/C++ and is smart, flexible, and fast. It is not limited by hardware specifications and architectures, causes less overhead, and applies to a wide range of scenarios.



```
onsole.log("2" + "1");  // => "21"
console.log("2" − "1");  // => 1
console.log('' == 0);    // => true
console.log(true == []); // −> false
console.log(true == ![]); // −> false
```

New Features

Compared with openEuler 20.09, openEuler 21.03 has the following feature updates:

- **iSulad for local volume management**

With iSulad, you can create local volumes for data persistence when you are creating or running a container. The data written into the volumes is persistent even after you delete the container. You can run volume management commands provided by iSula to manage local volumes.

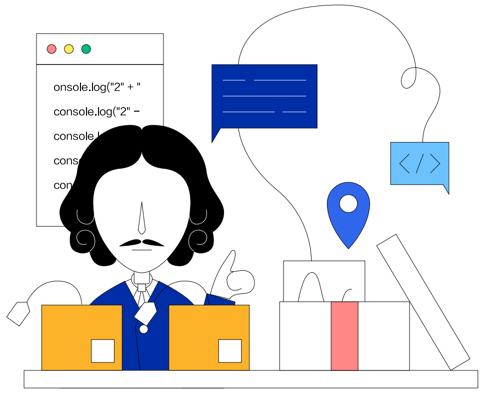
- **New features of isula-build**

1. Pull and push commands. Pull images to local drives and push images to a remote repository.
2. Manifest list management. Use the same manifest in different architectures to obtain the images of these architectures. A manifest contains create, annotate, inspect, and push commands.
3. OCI image format. Images in the OCI format can be built, saved, imported, exported, pushed, and pulled.
4. Designated user groups: Components of isula-build can run in designated user groups. Common users in a user group can run isula-build to improve security.
5. Batch importing and exporting of images. Multiple images can be saved to a tarball, or a tarball containing multiple images can be imported.

Virtualization Platform StratoVirt

StratoVirt is an enterprise-grade virtualization platform designed for cloud data centers. "Strato" is abbreviated from "stratosphere", which is the layer of Earth's atmosphere immediately above the troposphere. It refers to a light protective layer that protects services on the openEuler platform.

Rather than QEMU, which is heavily coded and targeted by frequent CVE vulnerabilities, new Rust-based virtualization architectures and components, such as CrosVM, FireCracker, and Rust-VMM, have emerged as more recognized virtualization solutions. These Rust-based virtualization solutions are secure and lightweight. They deliver high performance, low loss, and flexible component grouping.



```
onsole.log("2" + "
console.log("2" −
console.l
cons
con
```

StratoVirt has the following advantages:

- **Robust security:** StratoVirt offers language-level security based on Rust. Its advanced model design minimizes the attack surface and physically isolates each tenant.
- **Lightweight and low-noise:** When StratoVirt uses a simplified device model, it can start within 50 ms, control the noise floor within 4 M, and process serverless workloads.
- **Software and hardware collaboration:** StratoVirt supports x86 and Kunpeng-V virtualization.
- **Lightning fast scaling:** StratoVirt helps realize device scaling within milliseconds, providing flexible resource scaling capabilities for lightweight workloads.

New Features

- **Memory elasticity:** StratoVirt allocates and reclaims memory on the real-time memory demand of each workload. The memory reclamation speed can reach up to 3 GB/s through virtio-balloon.
- **Huge pages:** On lightweight VMs, StratoVirt supports huge pages to provide them with contiguous physical memory pages, improving the memory access efficiency on lightweight VMs.
- **I/O subsystem enhancement:** StratoVirt provides the multi-channel concurrent I/O capability. I/O-QoS improves the flexibility and stability of I/O traffic management on VMs.
- **System call filtering:** With the simplified device model design and the Seccomp filtering capability, StratoVirt requires calling only 35 systems in the simplest configuration, narrowing the system attack surface.

OpenStack

OpenStack is a joint open source project initiated by NASA and Rackspace, and is licensed by Apache. OpenStack has grown into a prosperous ecosystem that is suitable for nearly all cloud environments, due to its advantages such as simple deployment, massive scalability, and unified standards. OpenStack provides IaaS solutions with its mutually supplementary services, which are integrated through APIs.

OpenStack Victoria is a stable release in the OpenStack community in 2020, including multiple modules such as computing, storage, network, PaaS, security, and cluster management. Among these modules, Nova, Keystone, Neutron, Glance, Ironic, and Horizon have been adapted to openEuler.

The OpenStack SIG, led by China Unicom, has been responsible for the OpenStack porting and adaptation.

Feature Description

- Keystone provides authentication information and token management, creation, and modification functions for other OpenStack components, and uses a database such as MySQL to store authentication information.
- Nova provides VM creation, running, migration, and snapshot services, and provides APIs for interconnecting with controller nodes. The controller nodes deliver tasks and use nova-api for communication.
- The Neutron network management service manages the network node topology. It manages communication between private and public networks, communication and topology between VM networks, and firewalls on VMs. It offers the Neutron UI in Horizon.
- Glance provides image services including discovering, registering, and retrieving VM images. Glance manages images provided during VM deployment, including image import, formats, and templates.
- Cinder is a block storage service for OpenStack. It virtualizes a hard drive that you can attach to a VM regardless of what file system the VM is using. You can perform common operations on the new hard drive, for example, attaching, detaching, and formatting it, and converting its file system.
- Ironic deploys and manages bare metal servers, and maintains their lifecycles.

- Horizon provides a GUI-based operation console. It calls APIs to display and manage internal OpenStack resources.

Application Scenarios

Application scenario 1: deployment of a cloud computing platform

You can use the openEuler software installation source to deploy the OpenStack platform. This platform provides basic cloud platform functions.

Application scenario 2: development of a commercial cloud computing platform

Cloud computing service providers that rely on OpenStack can directly use openEuler as the server OS, and perform customized development based on openEuler

HA Cluster Solution

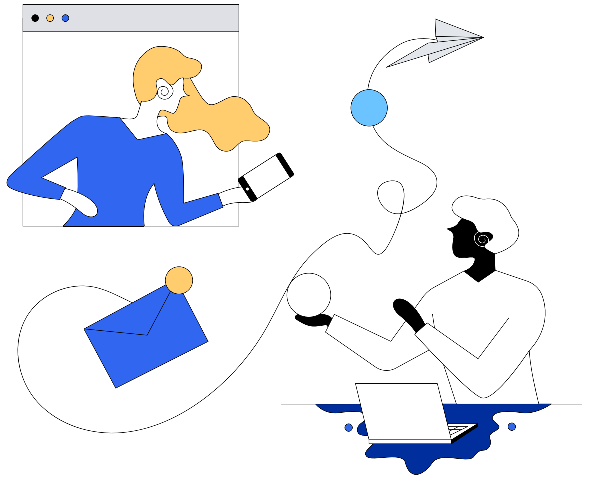
The HA cluster software delivers a general-purpose HA cluster solution. It presents an HA environment that ensures service continuity, continuous data protection, and disaster recovery. This solution provides various HA functions such as health check and second-level switchover. It eliminates single-point cluster faults caused by software bugs, hardware damage, or misoperations. The HA cluster solution maximizes the stability and reliability of mission-critical applications and data in a cluster or standalone system.

The HA SIG, led by KylinSoft, is responsible for the porting and adaptation of the HA cluster solution, which has been incorporated in openEuler.

Feature Description

- **Flexible cluster backup modes**

The HA cluster solution offers multiple cluster options, including dual-node hot backup, dual-node mutual backup, and multi-node backup (N+M), with a combination of multiple physical machines or physical and virtual machines. It is compatible with mainstream file systems and storage devices to fulfill various service protection needs.



- **Web-based, graphical cluster management**

The HA cluster solution provides a web-based GUI for cluster management. The HA-APIs are a set of back-end interfaces developed by KylinSoft. The HA SIG used the Go language to rebuild the API code and released back-end APIs V1.0. Compared with front-end management platforms, the web-based GUI of the HA cluster solution is intuitive and easy to operate. You can perform common resource and protection operations on the web-based GUI.

- **Heartbeat modes**

The HA cluster solution supports both single-heartbeat and dual-heartbeat modes to monitor shared data resources in an all-round manner.

- **Second-level switchover and automatic switchback**

The HA cluster solution has comprehensive protection mechanisms and a wide range of application agents. Once a server or resource becomes faulty, the system detects the fault and performs a switchover within seconds. After the fault is rectified, services are automatically switched back to the original server or resource. The automatic switchover and switchback eliminate downtime, keeping enterprise-grade applications online around the clock.

- **Data integrity protection in extreme conditions**

The shared data resources are monitored to ensure the server data integrity when an extreme fault occurs in a two-node cluster.

- **Neglectable system resource usage**

The HA cluster solution requires a low system resource usage and does not compete with protected applications.

- **Complete logs**

The HA cluster solution provides complete logs and debugging information to facilitate the monitoring and management by system administrators.

Application Scenarios

Application scenario 1: service continuity assurance

Enterprises' mission-critical applications must be kept online around the clock and no downtime is acceptable. The HA cluster software supports two-node hot backup, two-node mutual backup, and multi-node (N+M) modes. It automatically switches applications from the faulty server over to a normal server. The HA cluster software can detect a fault within 10 seconds and complete the switchover within 1 second.

Application scenario 2: database and system service protection

The HA cluster solution supports mainstream OSs and application software such as Nginx, httpd, and MariaDB, and allows secondary development. When the software on a cluster node is faulty, the HA cluster solution automatically switches the databases and system services from the faulty node over to a normal node.

Flourishing Ecosystem

pkgship: Software Package Information and Dependency Management Tool

pkgship is a tool used for managing RPM software package dependencies. It allows developers to query, introduce, upgrade, and delete the software packages, and helps evaluate application compatibility. It is designed to help developers quickly understand the dependencies between software packages of different versions and OSs.

pkgship provides front-end openEuler services for developers to query detailed dependency graphs and levels, and download files containing dependencies between software packages.

Feature Description

- **Import package information:** Based on different software library servers configured, pkgship can import information about all software packages in the OS of the corresponding version and integrate and analyze their dependencies.
- **Query package information:** Developers can query imported RPM source code and binary package information, including the version, license, and description, as well as the files in the software packages.
- **Query package dependency information:** pkgship obtains the dependency list, graph, and files containing software package dependency information, including installation dependency, compilation dependency, and self-dependent and dependent relationships.

Application Scenarios

Application scenario 1: evaluating software package compatibility

Evaluating the differences of one software package in different OS versions is an important reference to determine software package compatibility. Based on different Linux distributions and different versions of the same Linux distribution, pkgship provides developers with the version and file information of the same software package, helping developers determine the similarities and differences between software packages.

Application scenario 2: introducing new software packages

When new software packages are introduced to an open source community, problems often occur during software package compilation and installation. This is due to the compilation and installation dependencies that vary between software packages. The dependent software package also introduces other dependencies. pkgship helps developers quickly identify the scope of dependencies to improve the introduction efficiency.

Application scenario 3: upgrading and deleting software packages

Before upgrading or deleting a software package in a version, developers need to evaluate the impact scope of the software package. pkgship helps identify which software packages depend on the software package, helping developers quickly determine the impact scope.

Desktop Environments

UKUI

UKUI is a lightweight desktop environment developed by the Kylin team based on the Linux distribution. UKUI 3.0, developed using Qt, simplifies operations and balances visual display and interactions, for a natural, comfortable interface.

UKUI focuses more on ease-of-use and agility, and its components are less dependent on each other or on other suites and predominately run independently. It provides a reliable and efficient desktop environment with user-friendly interaction functions for openEuler 21.03.

Feature Description

1. **Control panel:** Allows developers to perform basic system settings, such as date and time, personalized settings, and device management.
2. **Start menu:** Manages all applications installed in the system and intelligently displays frequently used applications. Developers can switch between the default window and full-screen modes, and search for required content using Chinese, English, Pinyin, and initial letters.
3. **Taskbar:** Supports dark and light themes and other special effects (such as frosted glass), and allows real-time preview of files, folders, terminals, web pages, and images.
4. **Sidebar:** Divided into two parts. The upper part is used to manage notifications, and the lower part is used to operate the clipboard and small plug-ins, such as the alarm clock and notes.
5. **File manager:** Combines the search bar and address bar, and provides multi-tab display and convenient file search.

Application Scenarios

Application scenario 1: personal computers (PCs)

UKUI provides a desktop environment that is easy to operate.

Application scenario 2: servers

UKUI ensures balanced resource usage, providing stable and precise functions that simplify operations, offering best-in-class UX.

DDE

Deepin Desktop Environment (DDE) is a secure, stable, and easy-to-use Linux desktop environment. Structurally, it comprises the frontend written in Qt, backend written in Go, Gala as the window manager, and LightDM as the display manager.

Feature Description

1. **Control center:** Unity Operating System (UOS) uses the control center to manage basic system settings, including account management, network settings, date and time, and personalization, display, and system and application update settings.
2. **Window manager:** Shows different windows in different workspaces. Developers can manage the windows by group using the window manager.
3. **Desktop widgets:** Allows developers to set the wallpaper and screen saver and create shortcuts for frequently used applications on the desktop. It also provides common widgets such as the notification and clipboard.
4. **File manager:** Manages and presents files in a graphical manner, and provides file operation and retrieval functions. In addition, it supports basic CD-ROM burning, external devices, and network mounting.
5. **Dock:** Indicates the taskbar displayed at the bottom of the desktop. It consists of the launcher, application icons, tray, and system plug-ins. Developers can open, create, close, or exit an application, set the volume, connect to a Wi-Fi network, or access the shutdown interface on the dock.

6. **Launcher:** Manages all applications installed in the system. Developers can quickly find the target application using the categorized navigation bar or search function.
7. **Deepin Tool Kit (DTK):** Developed based on Qt5 and provides a bundle of user interface (UI) graphics libraries and tool libraries to facilitate development and debugging.

Xfce

Xfce is a lightweight desktop environment for UNIX-like OSs. It is a fast, lightweight, and user-friendly option that offers one of the industry's most appealing interfaces. It is designed to improve work efficiency, save system resources, and quickly load and execute applications.

Xfce consists of independent software components, which can be used independently or together.

Xfce is written in C language. In the most recent version (4.14), Xfce ports all core components to Gtk3 and GDBus, and most components are supported by GObject Introspection.

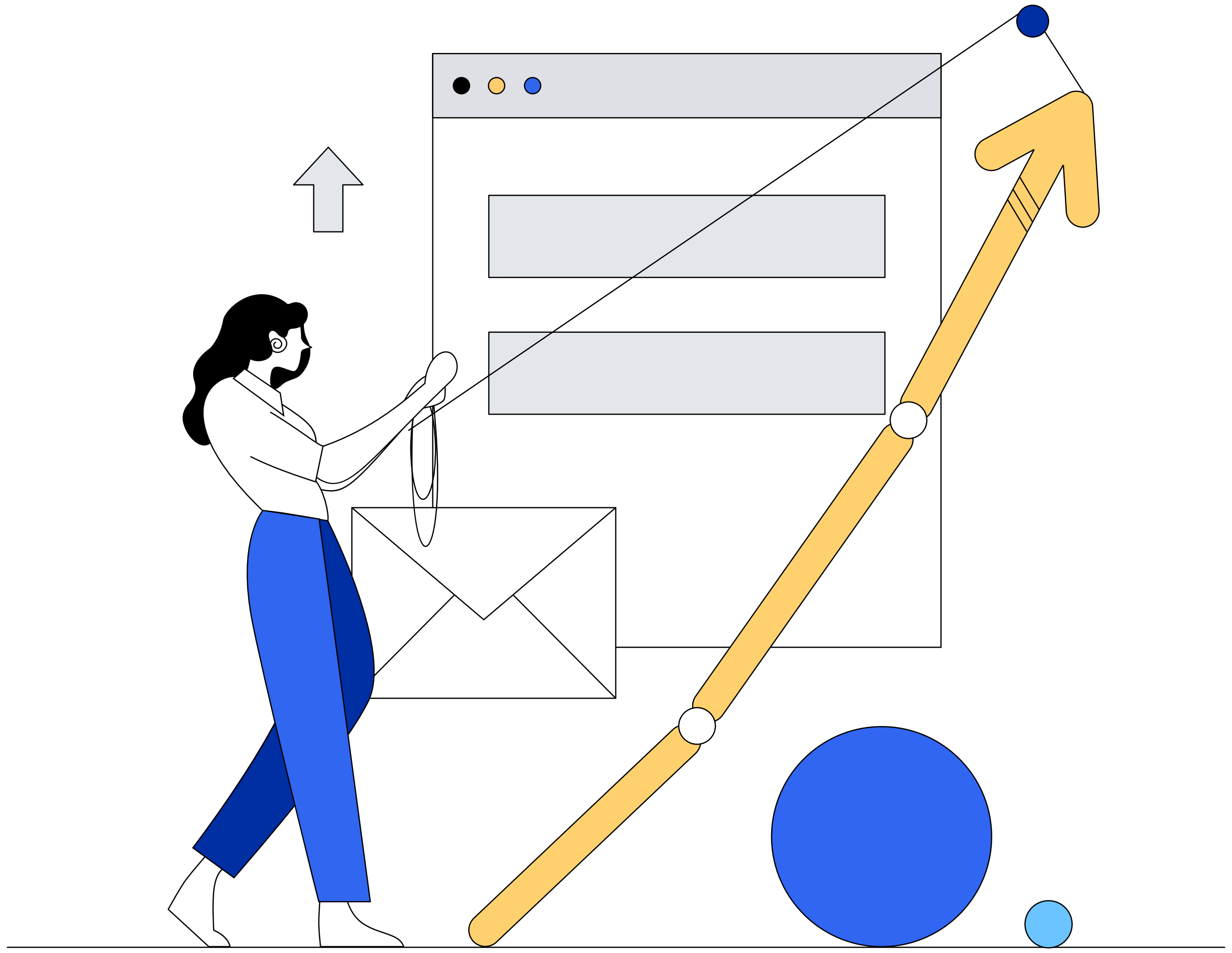
Xfce provides desktop environment support for openEuler 21.03 to extend openEuler's capabilities in the office desktop field.

Xfce is ported by Xfce Special Interest Group (SIG) led by TurboLinux.

Feature Description

- Application Finder (xfce4-appfinder): Quickly runs programs and commands.
- Configuration Storage System (xfconf): Provides a D-Bus-based configuration storage system.
- Desktop Manager (xfdesktop): Sets the desktop background image, icons, launcher, and directories.
- Development Tools (xfce4-dev-tools): Provides the Xfce4 development tools.
- Helper Applications (exo): Manages preferred applications and compiles .desktop files.
- File Manager (thunar): Manages Xfce files.
- Menu Library (garcon): Provides the menu library.
- Panel (xfce4-panel): Launches applications, switches between opened windows, program menu, and switches workspaces and menu plugins to browse applications or directories.
- Power Manager (xfce4-power-manager): Manages device power.
- Session Manager (xfce4-session): Saves the desktop status and the programs that will be restored upon the next startup.
- Settings Manager (xfce4-settings): Manages Xfce4 configurations.
- Thumbnail Service (tumbler): D-Bus service, which requests URI schemes and MIME-type thumbnails for applications.
- Utility Sharing Library (libxfce4util): Shares common non-GTK and utilities between Xfce applications.
- Widget Sharing Library (libxfce4ui): Shares commonly used Xfce widgets between Xfce applications.
- Window Manager (xfwm4): Manages windows.

Enhanced Features



Bisheng JDK

Bisheng JDK is an enhanced JDK developed by Huawei based on Open Java Development Kit (OpenJDK). It features high performance and high availability, optimizes ARM performance, and is suitable for supercharging production environments in any field. At present, Bisheng JDK can run on over 500 Huawei products in a multitude of industry scenarios. Bisheng JDK supports OpenJDK 8 and OpenJDK 11. Bisheng JDK 8 is compatible with Java Platform, Standard Edition (Java SE) and continuously contributes to the upstream OpenJDK community.

Bisheng JDK has the following advantages:

1. **Stable and efficient:** In benchmark tests, such as SPECjbb, Bisheng JDK delivers much better performance than AdoptOpenJDK, providing stable and efficient system running.
2. **Combination of software and hardware:** It fully utilizes the hardware features of the Kunpeng server for higher efficiency.
3. **Guaranteed security:** It synchronizes release updates with OpenJDK community editions, performs strict analysis and control, and applies patches to CVE vulnerabilities as and when needed.
4. **Open source:** It provides free and open sourced code.

Feature Description

- **Java Flight Recorder (JFR):** JFR is an online tool used for collecting diagnostic and profiling data that ensures minimal performance overheads. For applications that run for a long time where the default settings are used, the performance impact is less than 1%. To enable JFR in the production environment, launch the Java application (-XX:+FlightRecorder). A dump file will be generated for analysis. JFR can work with Java Mission Control (JMC) for better performance analysis visualization.
- **AppCDS (available in Bisheng JDK 8):** AppCDS is an extended version of Class-Data Sharing (CDS). It dumps application classes into JSA files to extend the class usage scope and improves the application startup and loading speed.
- **Garbage-First garbage collector (G1 GC) (available in Bisheng JDK 8):** The Java Virtual Machine (JVM) can detect the application load decrease and idle memory of the Java heap, automatically reduce the Java heap occupancy, and return the idle memory resources to the OS. G1 helps reduce overheads in the container scenario where resources are paid for based on subsequent usage.
- **Z Garbage Collector (ZGC) (available in Bisheng JDK 11):** ZGC is a scalable low-latency garbage collector with a maximum GC pause time of 10 milliseconds, a figure that does not change despite an increase of heap size. It is available in ARM64.
- **NUMA-Aware for the G1 GC (available in Bisheng JDK 11):** The JVM can make full use of hardware features. During application running, the memory of the local node is preferentially used for object allocation. During garbage collection, memory replication is preferentially performed on the same node to ensure data affinity of applications after garbage collection.
- **New rapid serialization (available in Bisheng JDK 8 and Bisheng JDK 11):** This function reduces redundant new transmissions and improves the efficiency of serialization and deserialization.

Application Scenarios

Application scenario 1: big data applications

Bisheng JDK is optimized for big data applications in terms of the runtime, GC, and Just-In-Time (JIT) compiler. For example, memory allocation and reclamation in GC are optimized, and redundant memory barriers in JIT code are eliminated. In the benchmark test of big data applications, Bisheng JDK delivers 5% to 20% higher performance than AdoptOpenJDK. In addition, the optimized features in Bisheng JDK, such as AppCDS and ZGC, further improve performance.

Application scenario 2: G1 GC for cloud computing

Memory scaling effectively prevents resource wastage. According to tests in low-traffic scenarios, the memory can be saved by more than 60%. In addition, Bisheng JDK uses a dynamic memory release policy to ensure smooth changes in memory usage.

Application scenario 3: Java applications on the Kunpeng server

Bisheng JDK optimizes the weak memory model of the Kunpeng server to avoid invalid memory barriers. NUMA-Aware for the G1 GC fully unleashes hardware performance and improves the memory access efficiency of applications. Bisheng JDK provides enhanced Java tools, such as JFR and jmap, to help developers quickly profile performance and locate faults.

GCC

Kunpeng GCC implements in-depth software and hardware co-optimization, and explores the ultimate performance in fields such as Scalable Vector Extension (SVE) and pipelines. Kunpeng GCC has the following advantages:

1. **Excellent performance:** Kunpeng GCC fully utilizes the hardware features of the Kunpeng server to achieve higher operating efficiency. In the benchmark tests such as SPEC CPU 2017, Kunpeng GCC delivers much better performance than GCC 9.3 of the upstream community.
2. **Open source:** Kunpeng GCC provides innovative distributions, open sourced code, and other contributions to upstream communities.

Feature Description

- **mcmodel=medium addressing in the ARM architecture:** Allows data (> 4 GB) to be properly accessed, which resolves the error caused by buffer overflow.
- **Quadruple-precision floating points in the ARM architecture:** Effectively improves the precision of floating point arithmetic.
- **SVE vector optimization in the ARM architecture:** Significantly improves program running performance for machines that support SVE instructions.
- **Matrix multiplication of bfloat16 (BF16) elements in the ARM architecture:** Supports on-demand instructions and intrinsic functions.
- **New loop optimization:** If this function is enabled, redundant loops can be effectively reduced, and multiple loops can be properly split and combined to improve program running performance.
- **Structure optimization:** The positions of structure members are rearranged so that frequently accessed structure members are placed in continuous memory space, improving the cache hit ratio and program running performance.

Application Scenarios

Application scenario 1: high-performance computing (HPC)

In the HPC test of Weather Research and Forecasting (WRF) applications, Kunpeng GCC delivers 10% higher performance than GCC 9.3 of the upstream community.

Application scenario 2: applications running on the Kunpeng server

In the SPEC CPU 2017 benchmark test, Kunpeng GCC delivers over 10% higher performance than GCC 9.3 of the upstream community.

secGear: Confidential Computing Framework

Confidential computing is a technology used in cloud computing and data centers that isolate sensitive data in a protected enclave. It provides a hardware-based trusted execution environment (TEE) for software that processes sensitive data. Content in the TEE is inaccessible to requests from common environments.

Intel Software Guard Extensions (SGX) for x86 servers allocates some main memory to the TEE, encrypts the TEE memory, and decrypts the memory when the CPU loads the memory. In this case, sensitive data remains protected in memory against malicious exploits.

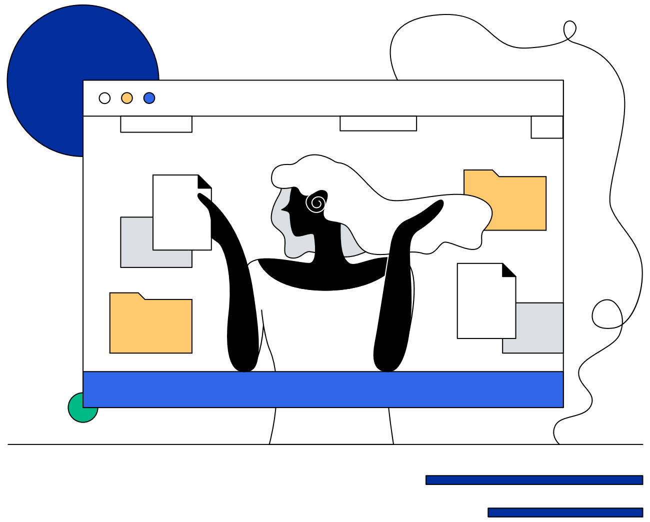
Arm TrustZone implements memory access control. The CPU in the normal world is blocked from access to the memory and devices allocated to the secure world. Malicious attackers cannot access the content in the secure world unless they crack the software in the secure world.

secGear is an application development framework based on the confidential computing technology. It simplifies the compilation of security applications to improve the development efficiency.

The implementation and programming interfaces of confidential computing vary according to the architecture. However, secGear supports multiple chip architectures, such as x86 and ARM, to provide consistent programming experience.

Feature Description

- Provides APIs for applications to manage the lifecycle of confidential computing and call secure functions.
- Provides POSIX-like interfaces for TEE application development to achieve consistent programming experience in the REE.
- Supports data encryption and persistence in the TEE and supports local and remote attestation.
- Provides necessary code generators and binary file signers for confidential computing.
- Supports the C programming language.
- Supports Intel SGX and Arm TrustZone.



Application Scenarios

Application scenario 1: multi-party computation (MPC)

MPC allows an organization or entity to use the data owned by multiple parties to perform computation without disclosing its own data to a third party. For example, there are three companies, each company has its own data, and the data is inaccessible to any third parties. How do we avoid data leakage if the data of the three companies needs to be merged and computed through AI training?

In such a case, the three companies must use a confidential computing technology to transmit data to each other's TEE through an encrypted channel. Remote attestation ensures that the TEEs and code are safe and secure, and each company shares data for AI training without disclosing its own data.

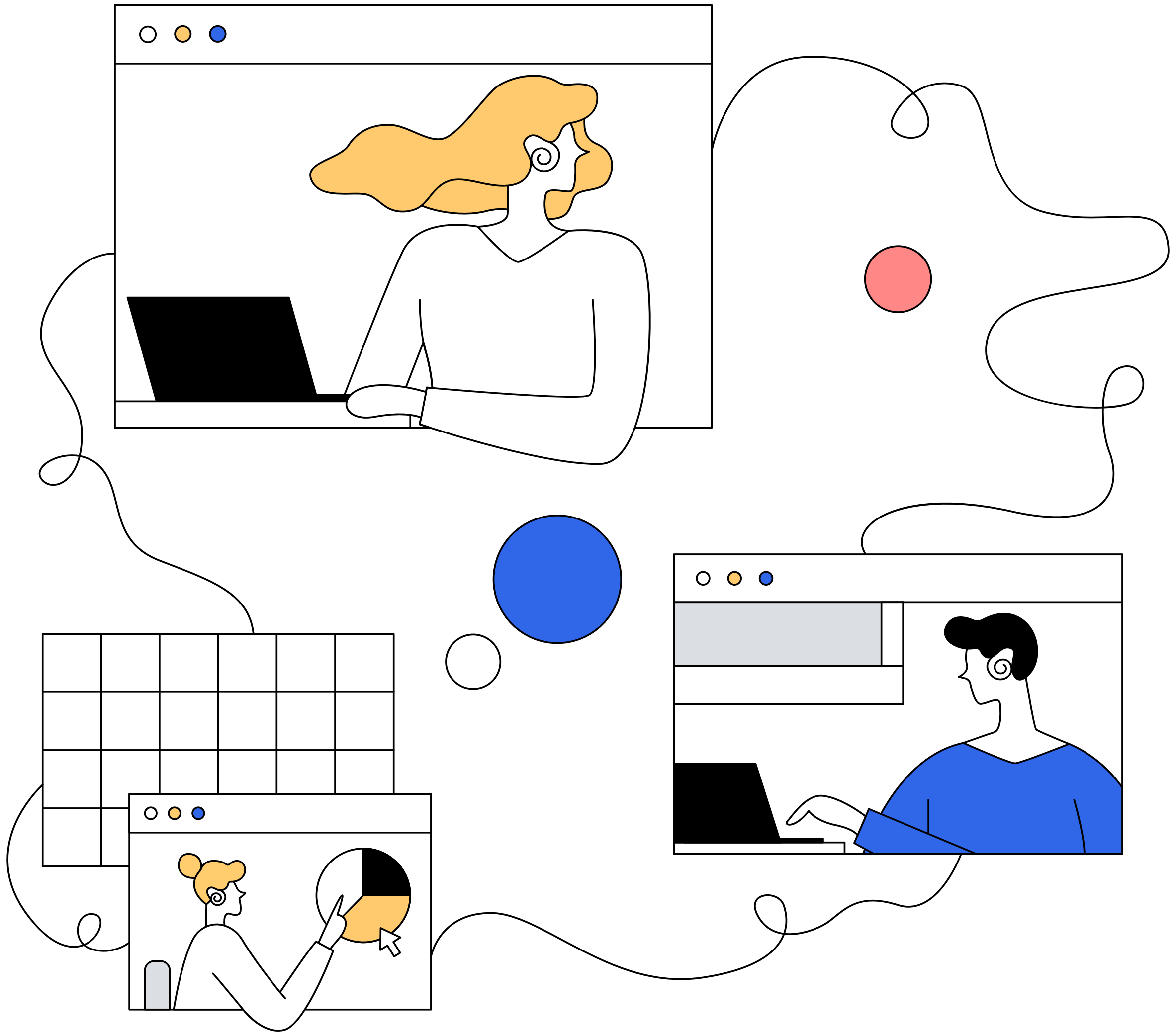
Application scenario 2: key management service

The key management services of public clouds now implement secure management of keys based on the hardware security module (HSM). With secGear, security modules based on confidential computing can be developed to replace the HSM.

Application scenario 3: secure database

In certain scenarios, to protect the certain confidential data, the database owner wants to ensure applications can only access the return result using SQL statements, restricting access to the database. Given the circumstances, secGear is a good choice to develop secure database applications, because this ensures that the database runs in a TEE for top-quality data security.

Community Governance



Vision

Build an innovative platform by cooperation within the openEuler community; build a unified and open community of openEuler OS to promote the multiple process architecture; and promote a prosperous software and hardware ecosystem.

Community Communication

openEuler contains many projects that are organized into community groups. Communication channels for these groups, including mailing lists, can be found in the relevant README files.

Mailing List

You can start a discussion on an open topic by sending an email to the relevant mailing list.

Visit https://openeuler.org/en/community/mailing-list/ to find a community mailing list. The following steps describe how to subscribe to a mailing list.

You can subscribe to a mailing list by visiting the web page or by email.

Web page

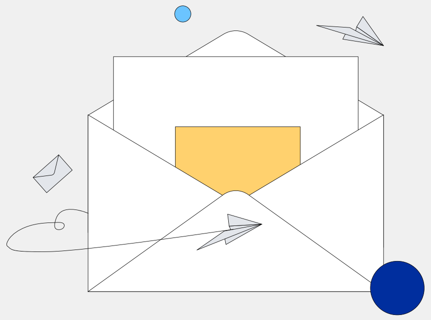
1. Click the list name in https://openeuler.org/en/community/mailing-list/ to go to the subscription page.
2. Enter the subscription email address and click Subscribe.
3. Log in to the mailbox and reply to the confirmation email sent from openeuler.org.

After this, you will receive a second email containing "Welcome" from @openeuler.org, indicating that you have successfully subscribed.

Note: If you have not received the "Welcome" email, it means subject form of the reply email is incorrect. In this case, reply to the original email again and include the original subject.

Email

1. Send an email with the title "subscribe" to the subscription address displayed next to the project name on the mailing list. The subscription address must consist of the list address and include the suffix "-join" (see example below).
2. Reply to the confirmation email sent from openeuler.org.



Take Dev (dev@openeuler.org) as an example. The email is as follows:

To: dev-join@openeuler.org

Subject: Subscribe

Body: NA

Currently, there are two types of mail subjects: announce and discussion. The method of sending a discussion email to a list is the same as that of sending emails to private addresses. It is a good practice, though not mandatory, to add a pair of square brackets and a subject as the prefix to the email subject. An announce email is used only to announce messages or precautions, therefore we cannot submit an issue based on this type.

Note: If you cannot receive any messages in your inbox, check your junk/spam mailbox.

To unsubscribe from a mailing list, perform the following steps:

1. Send an email whose subject is "unsubscribe" to the list of addresses you want to unsubscription from. Generally, you can add a suffix "-leave" after the list address (eg: dev-leave@openeuler.org).
2. Confirm the information in the confirmation email and reply.

After replying the confirmation email, the unsubscription is successful.

Obtaining Help

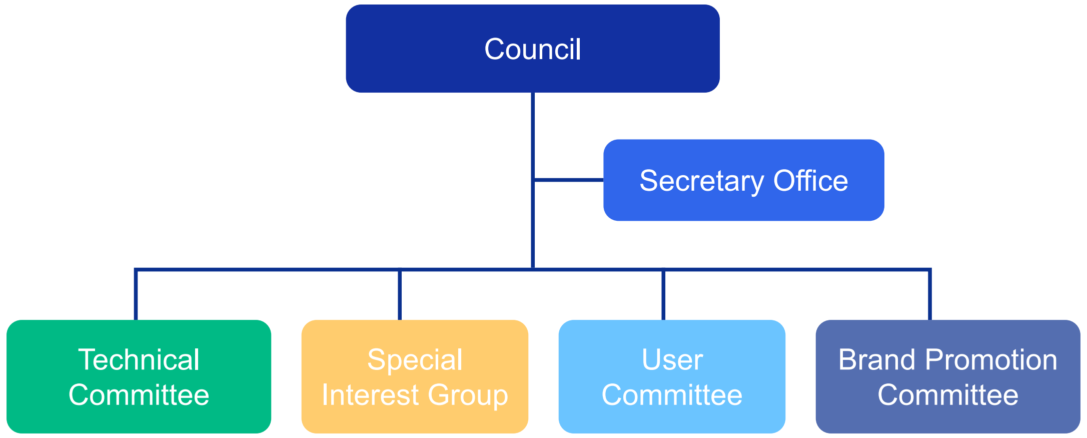
If you encounter any problems during the mailing process, contact the infrastructure support team:

- Email: infra@openeuler.org

If you find any bugs related to the mailing list, please submit an issue to the infrastructure team. For details about submitting an issue, see the descriptions below.

Community Governance Organization Structure

The community governance organization guides the development direction of the openEuler community, and comprises the Council, Secretary Office, Technical Committee, User Committee, Brand Promotion Committee, and Special Interest Groups (SIG).



The Council formulates the long-term development plan and guides individual and collective policies for the community. It reviews the work of the user committee and brand promotion committee, and manages the work plan of each committee. The council is also responsible for promoting the openEuler community and related OS distributions to various industries worldwide for wide-scale usage and ecosystem construction.

Secretary Office

Daily work of the Secretary Office under the guidance of the openEuler Council is as follows:

1. Implements Council resolutions, organizes and holds Council meetings, and coordinates with community agencies.

2. Prepares quarterly and annual work reports of the community, and releases the reports after approval from the management board.

Technical Committee

The openEuler Technical Committee (TC) is the technical leader of the openEuler community.

The main responsibilities of the TC are as follows:

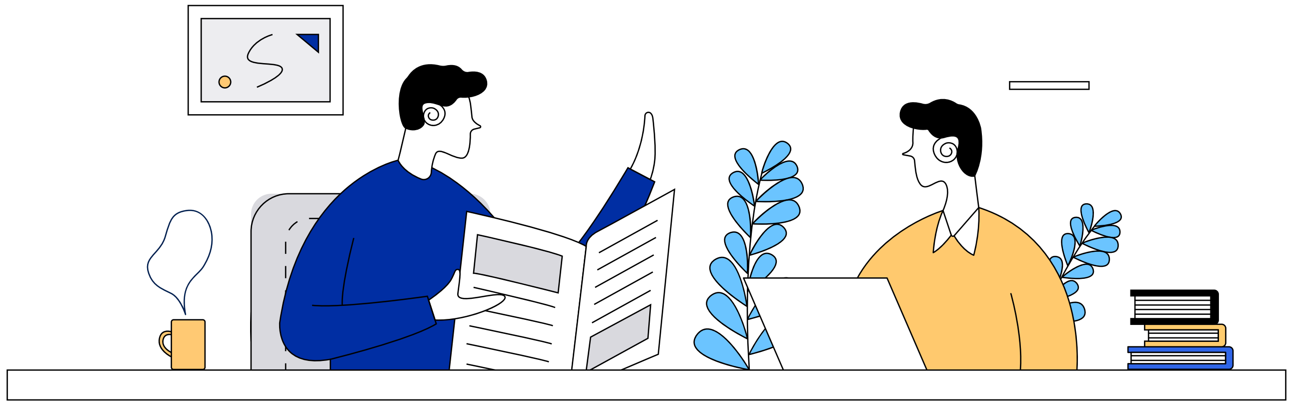
1. Makes the final decision on technical matters.
2. Finalizes the vision and direction of community technology development.
3. Establishes, coordinates, and performs other duties for the community SIGs. In addition, resolve the collaboration conflicts (if any) between SIGs, and coaches, reviews, and monitors the daily operation of the SIGs.
4. Implements the daily development work of the community and ensures the high-quality release of openEuler OS versions.
5. Leads innovation of system architecture, kernel, virtualization, cloud native, and security technologies to ensure continuous competitiveness.
6. Guides the community to establish original open-source projects and continuously build the technical influence of the community.

Conference Organization

Official conference (public): The TC holds a public online discussion at 10:00 a.m. (GMT+8) on alternating Wednesdays.

Details on how to join the conference will be released in the mailing list one day in advance.

This conference is open to developers who are interested in the openEuler community.



Special Interest Group

A Special Interest Group (SIG) is a team designated to a domain that holds regular tasks and activities to achieve the delivery objectives. SIGs have transparent rules, and must comply with the openEuler code of conduct. Anyone can join an SIG and contribute to the group projects. You can find all SIGs at: https://openeuler.org/en/sig/sig-list/.

User Committee

The user committee is the organization responsible for communicating with the end users of the openEuler community, and has the following responsibilities:

- Collects technical and product requirements for the openEuler community edition, releases openEuler OS versions, and organizes other open source projects of the openEuler community. In addition, provides feedback to the technical committee and the council, promotes the technical roadmap of the technical committee to meet the requirements of end users, and improves marketspace and long-term planning.
- Organizes best practices of the openEuler community edition, OS release version based on the openEuler community technology, and other open source projects of the openEuler community, and cooperates with the Brand Promotion Committee to publicize the cases with the authorization of end users.

Brand Promotion Committee

The Brand Promotion Committee is responsible for promoting the openEuler and related brands of the community. Its main responsibilities are as follows:

- Promotes the openEuler OS technology and community to improve the influence of the openEuler brand.
- Guides the wide use of the openEuler OS technology to build a global ecosystem.

Contributions

To contribute to the openEuler community, the first step is to select your desired project from the SIG/project list of openEuler. Once you are subscribed, you can attend SIG/project meetings and subscribe to the mailing list. An SIG or a project is usually composed of a series of help-wanted issues that you can work on.

Signing CLA

You must sign a Contributor License Agreement (CLA) before you can contribute to the community.

Community Code of Conduct

he openEuler community complies with the code of conduct stipulated in the Contributor Convention V1.4. For details, see the V1.4 version.

To report explicit or inappropriate behaviors, you can contact the openEuler Technical Committee at: tc@openeuler.org.

Commitment of Contributors

To maintain an open and professional environment, the openEuler community will not tolerate harassment, bullying, aggression or bad behavior of any kind. These include but are not limited to the basis of the following categories: race, color, ethic or national origin; age; nationality; sex, gender, or sexual orientation; bodily shape; physical or mental disabilities; or level of experience, education, or social status.

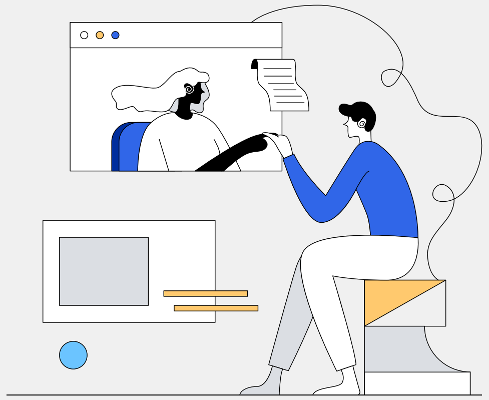
Our Principles

Actions that contribute to the creation of a positive environment include but are not limited to:

- Friendly and inclusive choice of words
- Respectful of diverse viewpoints and experiences
- Open to criticism and suggestions
- Prioritize actions in the interests of the community

The openEuler community prohibits harassment, explicit, and any other unacceptable behavior, online or otherwise, including but are not limited to:

- Pornographic, violent, sexual, or other explicit comments, images, and videos
- Comments that are deemed disruptive, inciting hate, rumors, insults, or derogatory statements, and attacks on personal and political beliefs
- Public or private harassment (sexual, bullying or otherwise) to colleagues or subordinates
- Release other private information, such as addresses and email addresses, without explicit prior consent
- Other misconducts that can be justified as a violation of professional ethics



Our Commitment

Maintainers of the community and related projects hold the right to interpret "proper conduct", and fully and impartially correct misconduct when needed. Maintainers hold the right to remove, edit, and reject any and all comments, code, or other issues that violate the code of conduct. Community project maintainers may temporarily or permanently ban any participant whom they consider to be improper, threatening, offensive, or harmful.

Application Scope

The code of conduct applies to everyone, everywhere, who participate in the openEuler community. The code of conduct applies to anyone who represents the community, as well as the public platforms where the person joins.

Representing the community includes but is not limited to: using the official email of the community, posting messages on official media accounts, and participating in online or offline activities as a designated representative.

The code of conduct on behalf of the community may further be defined and interpreted by the Secretary Office and submitted to the Council for approval and issuance.

Supervision and Investigation

To report abuse, harassment and misconduct found within each project team, send emails to the Secretary Office at: secretary@openeuler.io.

The Secretary Office will consider and investigate all complaints and respond accordingly. The project team is obligated to keep whistleblower information confidential.

Implementation

Send an email to tc@openeuler.org to report abuse, harassment, and misconduct to the project team.

The maintenance team will review and investigate all complaints and respond to them as necessary. The project team is obligated to keep whistleblower information confidential. Specific implementation guidelines may be issued separately later.

Anyone who fails to comply with or execute the code of conduct may be temporarily or permanently disqualified from participating in the project, until the matter is resolved with the project owner or other members.

Community Contributions

We work hard to ensure that the documents and software found within the community are best-in-class. However, we are aware that documents can be improved (such as the one you are reading), code needs to be reviewed, functions or variables can be reconstructed or commented out, and test cases can be supplemented and optimized. We will help you understand the organization of openEuler SIGs and walk you through your first contribution.

Introduction to the SIG

SIGs, short for Special Interest Groups, are organizations within the openEuler community that are designed to better manage and improve the work process.

- SIGs are open to anyone to join and contribute.
- An SIG is established for one or more specific technology projects. SIG members promote the output of deliverables and strive to make the deliverables a part of the openEuler community.
- Core members of the SIG govern the group. Core members are based on the accumulated experience and contributions.
- Each SIG has one or more projects on Gitee, with each project comprising one or more repositories. The SIG deliverables are stored in these repositories.
- As a member, you can submit issues in the repository of a specific SIG, participate in discussions, resolve issues, and participate in reviews.
- Members can communicate with other members in the SIG through the mailing list and during video conference.

Finding the most relevant SIG of your domain can help you obtain a quicker response of the issues submitted. There are two methods to find the SIG.

- **Method 1:** To find your desired SIG or project, you can view the list of all SIGs established in the openEuler community. The following information about the SIG is also provided:
  - Projects under the SIG and subsequent the repository addresses
  - Communication methods within the SIG, including mailing list and video conference
  - Contact information of maintainer
- **Method 2:** If you know the project name, you can perform fuzzy search in the repository list on the openEuler website to quickly locate the home page address of the target project. Generally, you can find the SIG information, communication methods, group member usernames, and contact information in the README.md file of the project home page address.

If you cannot locate the SIG you are interested in using either of the preceding methods, you can send an email to community@openeuler.org for help. You are advised to use [Development Process Question] as the subject of the email, and include details of the SIG or project that you are looking for in the body.

How to Contribute

Review an Issue

- **Find the issue list:** On the toolbar of the home page (Repository of the project on Gitee) of the project that you are interested in, click Issues to find the issue list of the SIG For example, the issue list address of the community team is https://gitee.com/openeuler/community/issues.
- **Review an issue:** If you want to handle one of the issues, you can assign it to yourself. In the comment box, enter /assign or /assign @[yourself] in the comment box. The community bot will assign the issue to your account, and your name will be displayed in the owner list.
- **Participate in discussion of an issue:** Each issue may have been discussed by participants. If you are interested in the issue, you can comment on it in the comment box.

Raise Questions or Suggestions

- **Raise a problem:** If you find a problem or bug, you can raise the problem to the admin by submitting an issue on the issue list repository of a specific project. When submitting an issue, comply with the issue submission rules.
- **Feedback:** You can share your opinions or suggestions in the SIG by submitting an issue, which is a public message that is open to everyone. To attract more attention, you can also attach the issue link to the email and send it to everyone through the mailing list.

Set Up the Development Environment

1. Install openEuler.
2. Prepare the development environment.
3. Download and configure software packages.

For details, see Appendix 1.

Participate in Coding

- **Understand the SIG and development precautions in the project**

The coding language, environment, and coding conventions may differ between each SIG project. If you want to understand and participate in coding, we provide contributor guides tailored to each project. The guide is generally found in the CONTRIBUTING.md file on the SIG homepage, alternatively, you can directly find the file in README.md of the project.

In addition to these documents, the SIG may also provide other guidance information located in the specific community directory of the SIG or its projects. If you do not find relevant information or have questions about related information, you can submit an issue in the SIG or send questions to the mailing list of the SIG to which the project belongs. If you do not receive a response, you can ask community@openeuler.org for help.

- **Download code and pull a branch**

To contribute code, you need to learn how to download code from Gitee and incorporate code appealing for a pull request (PR). The method of using the hosting platform is similar to that of GitHub. If you have used GitHub before, you can roughly understand or even skip this chapter.

- **Modification, construction, and local verification**

After completing the modification on the local branch, perform the construction and local verification.

- **Submit a PR**

When you submit a PR, it means that you are already contributing code to the community.

- **Add a Software Package**

After a software package is added to Gitee, openEuler automatically creates a repository with the same name on openEuler:Factory on OBS. In this way, when the code is submitted to the created Gitee repository, the code compilation is automatically checked.

Review the Code

openEuler is an open community. We hope that all participants in the community can become active reviewers.

To make submissions easier to accept, all contributors must:

- Comply with the coding conventions of the SIG, if any.
- Prepare complete submission information.
- If you submit a large chunk of code at one time, you are advised to separate the large content into a series of logically small sets. Submitting the content separately will make it easier for reviewers to understand your ideas.
- Label the PR with the appropriate SIG and monitor labels. This ensures the community bot will help you better complete the entire PR process.

For reviewers, it is strongly recommended that you follow the code of conduct and respect each other for better collaboration. The article The Gentle Art Of Patch Review is a good resource for key review points that code reviewers are expected to understand. Among which, they should encourage new contributors to actively participate in the program, instead of discouraging new contributors by giving overly critical comments. In particular, pay attention to the following points during review:

- Is the idea behind the contribution sound?
- Does the contribution feature correct formatting?
- Will the contribution need further modification?

Note: If your PR does not draw enough attention, you can seek help from the mailing list of the SIG or send email to dev@openeuler.org.

Select the Community Component Package

Creating an RPM package, also called packaging, refers to the task of compiling and binding software and metadata, such as the full name of the software, description, and dependency list required for normal running. This is to allow users to install, remove, or upgrade the corresponding software using the package management tool.

Packaging guide

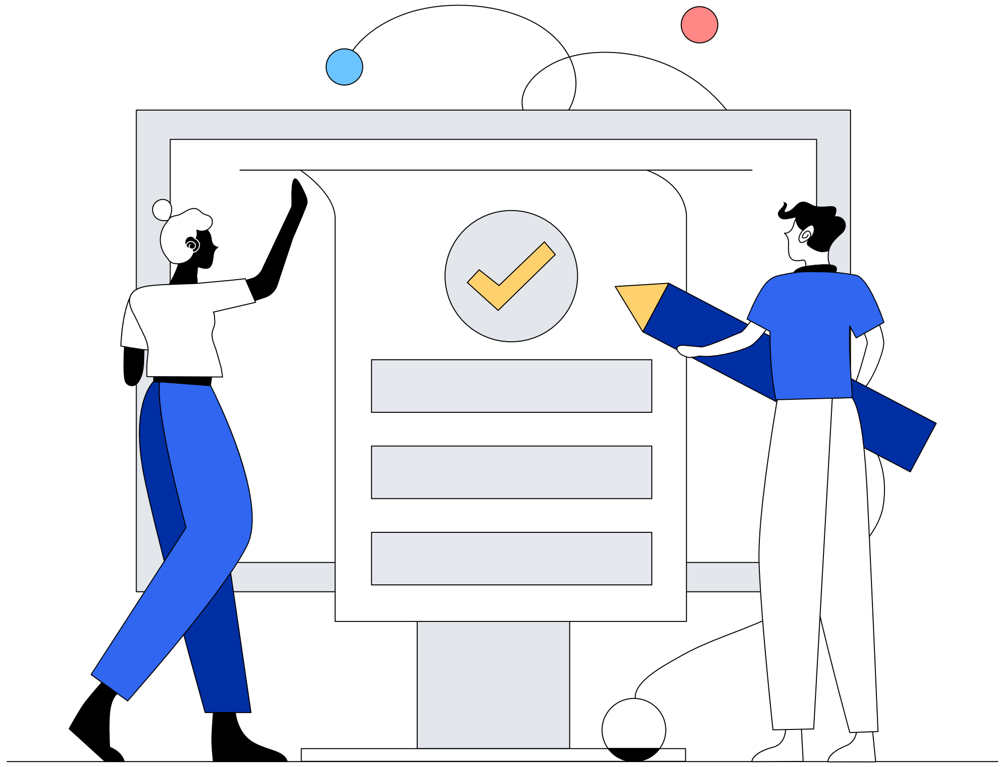
openEuler standardizes a variety of open source projects into a coherent system. Therefore, a packaging guide is drafted to standardize RPM development.

- openEuler complies with Linux Standard Base (LSB) to reduce the differences between distributions. openEuler also complies with the Linux Filesystem Hierarchy Standard (FHS). This standard is a reference on how to manage the Linux file system hierarchy.
- In addition to following these general rules that are followed by Linux distributions, this document standardizes the actual details of packaging for the openEuler community version.

Disclosure of Community Security Issues

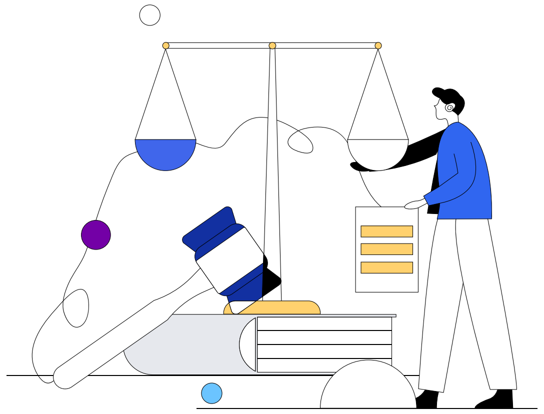
- Security handling process
- Security disclosure information

For details about the security handling process and security disclosure information, see Appendix 2.



Copyright Description

Trademarks



Appendixes

Appendix 1: Setting Up the Development Environment

Environment Preparation	Address
Downloading and Installing openEuler	https://openeuler.org/en/download/
Preparing the Development Environment	https://gitee.com/openeuler/community/blob/master/en/contributors/prepare-environment.md
Building a Software Package	https://gitee.com/openeuler/community/blob/master/en/contributors/package-install.md

Appendix 2: Security Handling Process and Security Disclosure Information

Disclosure of Community Security Issues	Address
Security Handling Process	https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-process.md
Security Disclosure Information	https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-disclosure.md