Huawei Cloud EulerOS（HCE）

User Guide

Issue 01

Date 2025-09-19

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Usage Overview

You can use Huawei Cloud EulerOS in the following ways:

- Select an HCE public image when creating an ECS for the first time.
- Change the OS to HCE.

  If only the ECS OS needs to be changed but other details (such as network interfaces, disks, and VPNs) need to be kept the same, as long as the software is loosely coupled with the OS, you can **change the OS** to HCE with only a few additional changes needed.
- Migrate the OS to HCE.

  If you want to change the ECS OS but retain other details (such as network interfaces, disks, and VPNs) and the OS settings, you can **migrate the OS** to HCE.

  NOTE

  The OS can only be migrated to Huawei Cloud EulerOS 2.0 Standard Edition, or Huawei Cloud EulerOS 1.1 CentOS-compatible Edition.

Item	OS Change	OS Migration
Data backup	• Data in all partitions of the system disk will be lost, so back up the system disk data before you change the OS. • Data in data disks remains unchanged.	• System disk data is not lost, but you are still advised to back up the system disk data, in case there are any system software exceptions. • Data in data disks remains unchanged.
Custom settings	After the OS is changed, custom settings such as DNS and hostname will be reset and need to be reconfigured.	After the OS is migrated, custom settings such as DNS and hostname do not need to be reconfigured.

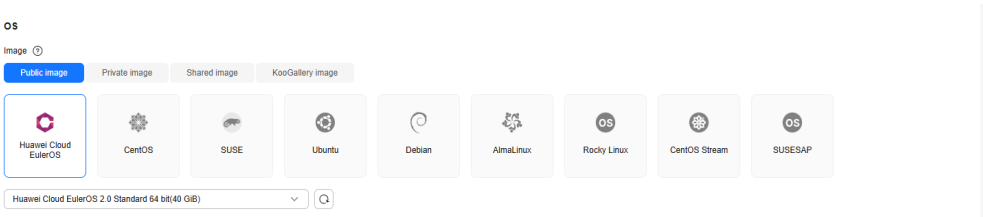
2 Selecting HCE as the Public Image When Creating an ECS

Procedure

1. **Log in to the ECS console**.
2. Select HCE as the public image.

   On the **Configure Basic Settings** page, select **Public Image**, and select **Huawei Cloud EulerOS** and an image version. For details about the other steps to purchase an ECS, see **Purchasing an ECS**.

   Figure 2-1 Selecting a public image



3 Changing an OS to HCE

Constraints

- If you want to change the OS of a yearly/monthly ECS, the system disk capacity may be insufficient for the new image. If this is the case, you need to detach the system disk from the ECS and expand its capacity before changing the OS.
- There must be at least one EVS disk.
- Changing from BIOS to UEFI is not supported.

Important Notes

- After the OS is changed, the original OS will not be retained, and the original system disk and data on the system disk will be deleted, including data on the system partition and other partitions. Back up the data before changing the OS. For details, see **Backing Up ECS Data**.
- The data in data disks remains unchanged.
- After the OS is changed, the IP and MAC addresses remain unchanged.
- The system will automatically start after the OS is changed.
- After the OS is changed, the system disk type cannot be changed.
- After you change the OS, you need to deploy services in the new OS.
- After the OS is changed, custom settings such as DNS and hostname will be reset and need to be reconfigured.

Billing Rules

- After the OS of a pay-per-use ECS is changed, the price may increase if the new OS needs more disk capacity.

Prerequisites

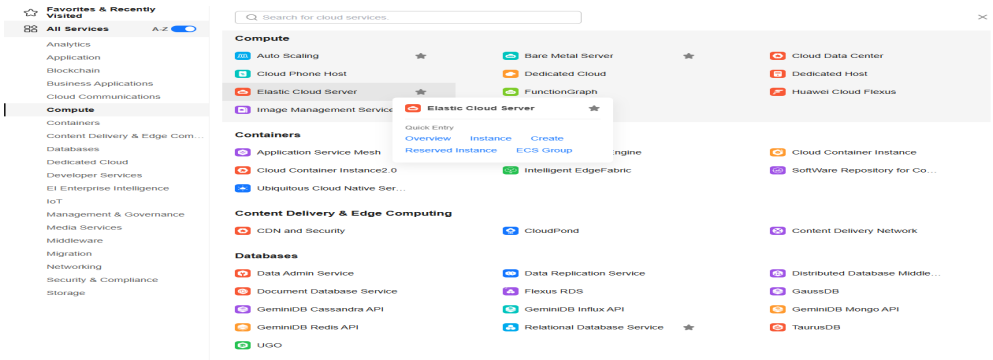
- The source OS has a system disk attached.
- If a password is used to log in to the source OS but a key pair is required to log in to the target OS, the key pair has to be available.
- If you use a private image to change the OS, create a private image first. For details, see **Image Management Service User Guide**.

- If the image of a specific instance is required, ensure that a private image has been created from that instance.
- If a local image file is required, ensure that this image file has been imported into the cloud platform and registered as a private image.
- If a private image from another region is required, ensure that this image has been replicated to the target region.
- If a private image from another account is required, ensure that the image has been shared with you.

Procedure

1. Log in to the management console.
2. Click ☰. Choose **Compute** > **Elastic Cloud Server**.

   Figure 3-1 Selecting an ECS



3. Locate the row containing the target ECS. Choose **More** > **Manage Image** > **Change OS** in the **Operation** column.

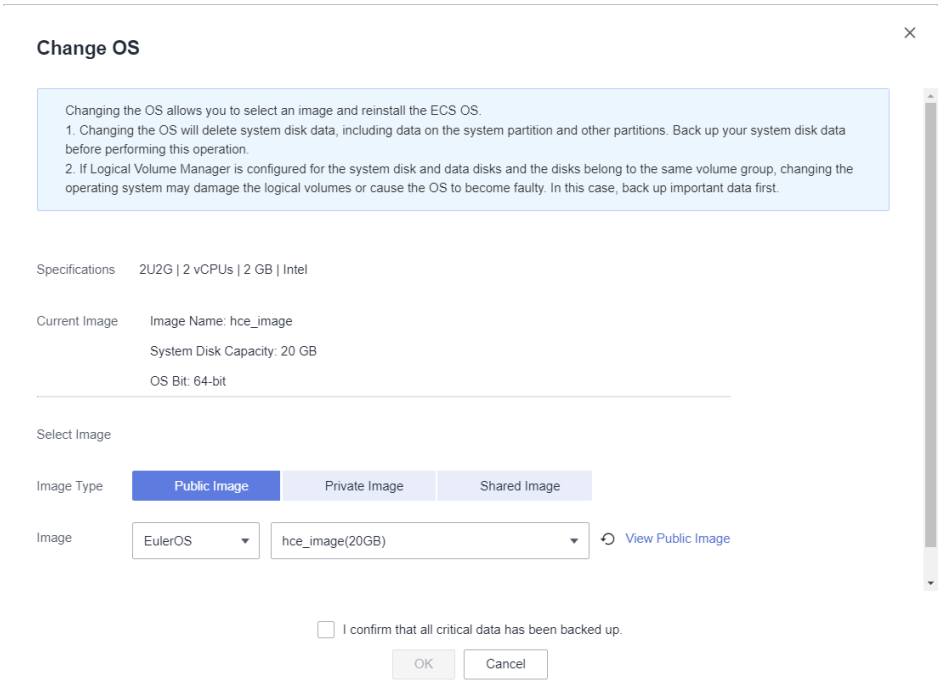
   Stop the ECS before this step, or select **Automatically stop the ECSs and change their OSs** in this step.
4. Change the **Image Type** and **Image**.

   NOTE

   For a yearly/monthly ECS, if the system disk size is smaller than the image size, you must detach the system disk, expand its capacity, and attach it to the ECS before changing the OS.

   For details about how to expand the system disk capacity, see **Disk Capacity Expansion**.

Figure 3-2 Changing the OS



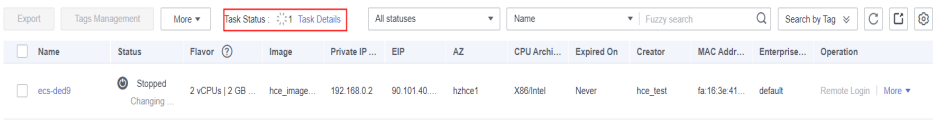
5. Set **Login Mode**.

   If the ECS uses a key pair for login authentication, you need to provide a key pair. For details, see in the *Elastic Cloud Server (ECS) User Guide***Passwords and Key Pairs**.

6. Click **OK**.
7. In the **Change ECS OS** dialog box, confirm the specifications, read and agree to the agreement or disclaimer, and click **OK**.

   The ECS status changes to **Changing OS**. When **Changing OS** disappears, the change is successful.

Figure 3-3 Checking the task details



Follow-up Operations

- If the OS before and after the change is Linux, and auto-mount on startup has been enabled for the data disk, the auto-mounting information will be lost after the OS is changed. You will need to update the **/etc/fstab** configuration:

  a. Write the new partition information into **/etc/fstab**. It is a best practice to back up the **/etc/fstab** file before writing data into it.

     To enable auto-mounting on startup, see **Initializing a Linux Data Disk (fdisk)**.

  b. Mount the partition to use the data disk.

     mount *diskname mountpoint*

c. Check that the partition was mounted successfully.
```
df -TH
```

- If the OS change fails, repeat steps 2 to 7 to retry.
- If the retry fails, contact customer service for manual recovery.

4 Migrating an OS

4.1 Using x2hce-ca to Evaluate the Compatibility

4.1.1 Overview of x2hce-ca

x2hce-ca is a free tool provided by Huawei Cloud used to evaluate the compatibility between applications and the OS.

You can use x2hce-ca to evaluate the compatibility before migrating an OS.

Table 4-1 x86 public images that support compatibility evaluation

OS Series	Source OS	Target OS
HCE	64-bit: Huawei Cloud EulerOS 1.1	HCE 2.0 Standard Edition (64-bit)
EulerOS	64-bit: EulerOS 2.11/2.10/2.9/2.5/2.2	HCE 2.0 Standard Edition (64-bit)
CentOS	64-bit: CentOS 7.9/7.8/7.7/7.6/7.5/7.4/7.3/7.2/7.1/7.0 64-bit: CentOS 8.3/8.2/8.1/8.0	HCE 2.0 Standard Edition (64-bit)
	64-bit: CentOS 7.9/7.6	Huawei Cloud EulerOS 1.1 CentOS-compatible Edition

Table 4-2 Arm public images that support compatibility evaluation

OS Series	Source OS	Target OS
EulerOS	64-bit: EulerOS 2.11/2.10/2.9/2.8	HCE 2.0 Standard Edition (64-bit, Arm)

4.1.2 Constraints

- Do not run x2hce-ca in the runtime environment because additional resource packages are created during its installation. The x2hce-ca tool can only be installed and used in HCE and CentOS.
- The x2hce-ca tool can scan only files in .jar, .py, .pyc, .bin, .sh, .rpm, or .ko format. RPM files must be written in C, C++, Java, or Python.
- The x2hce-ca tool does not support rollback. If a task is interrupted unexpectedly, you can retry. The residual files in the **/opt/x2hce-ca/** directory do not affect the use of the tool.
- The x2hce-ca tool can run only in the system that meets the requirements in the table below.

Table 4-3 Hardware requirements for running x2hce-ca

Hardware Type	Description
Architecture	x86_64, aarch64
CPU	Dual-core or higher
Memory	At least 8-GB available memory
System disk/Root partition	20 GB or above

4.1.3 Installing x2hce-ca

1. Confirm that the repository is configured correctly.

   Check whether the parameters in the **/etc/yum.repos.d/hce.repo** file are configured correctly. The correct configuration for HCE 2.0 is as follows:

```
[base]
name=HCE $releasever base
baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

[updates]
name=HCE $releasever updates
baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2

[debuginfo]
name=HCE $releasever debuginfo
baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
enabled=0
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
```


NOTICE

To install x2hce-ca on CentOS, you do not need to configure the repository. Download the latest x2hce-ca-hce from **https://repo.huaweicloud.com/hce/2.0/updates/**.

2. Install x2hce-ca.

   Run **yum install -y x2hce-ca-hce** to install x2hce-ca. After the installation is complete, the directories listed in **Table 4-4** will be generated.

   Table 4-4 Directories generated after x2hce-ca is installed

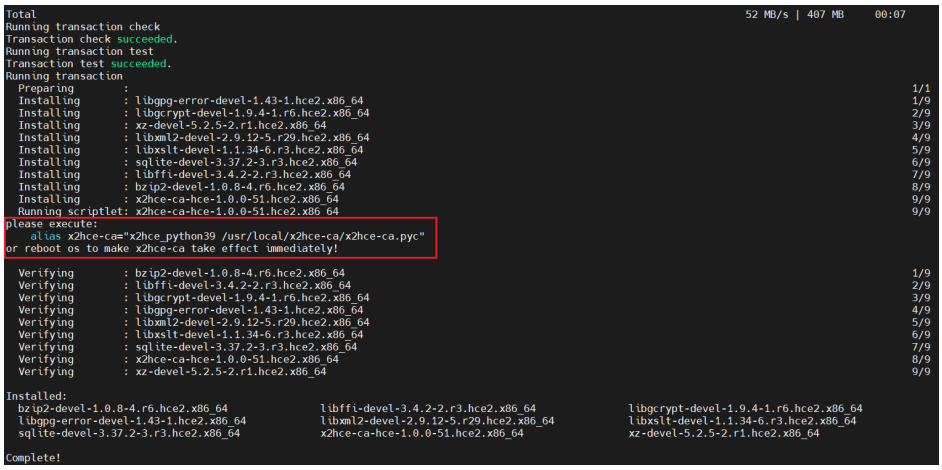
Directory	Description
/var/log/x2hce-ca	Stores log files of the x2hce-ca tool.
/var/log/aparser	Stores log files of the parser.
/opt/x2hce-ca/output	Stores the reports generated by the x2hce-ca tool.
/opt/x2hce-ca/scan	Stores the application software packages to be scanned.
/etc/x2hce-ca/config	Stores static configuration files.
/etc/x2hce-ca/database_2.0.0.630	Stores database files.
/usr/local/x2hce-ca	Stores program files.
/usr/local/x2hce-python-3	Python installation directory of the tool

   NOTICE

   To install x2hce-ca on CentOS, run **yum install -y x2hce-ca-hce-1.0.0-53.hce2.x86_64.rpm**. Use the version of your downloaded x2hce-ca-hce.

3. To make x2hce-ca take effect, perform operations as prompted after the installation is complete. **Figure 4-1** is an example of the prompt, prompting you to run **alias x2hce-ca="x2hce_python39 /usr/local/x2hce-ca/x2hce-ca.pyc"** or restart the OS.

Figure 4-1 Example prompt after the installation of x2hce-ca-hce



4.1.4 Evaluating Software Compatibility

Scanning Methods

The x2hce-ca tool can:

- Scan a single application package or multiple application packages on the source OS.
- Scan all application packages in a given directory or multiple directories on the source OS.

Procedure

1. Log in and switch to **root**.
2. Scan the application packages to check for compatibility.
   x2hce-ca scan <option> [-os_name *Source OS name*] [-target_os_name *Target OS name*]

   NOTE

   Run the following command to verify the default Java version:

   java -version

   - If Java 1.8.0 is installed on the target server, subsequent scanning is automatically performed.
   - If Java 1.8.0 is not installed on the target server, perform the following operations (which vary depending on the OS):
     - If the OS is HCE 2.0, the missing Java dependencies java-1.8.0-openjdk-devel, java-1.8.0-openjdk, and java-1.8.0-openjdk-headless will be installed automatically.
     - If the OS is not HCE 2.0, an error message will be displayed to remind you to install the missing Java dependencies. Run the following command to install the Java dependencies:

       yum -y install java-1.8.0-openjdk-devel
   - If Java of multiple versions is installed on the target server and the default version is not 1.8.0, run the following command and set it to 1.8.0:

     update-alternatives --config java

   <option> has the following settings:

- **Dir_Name/App_Name**: scans a single application software package.

  The following uses x86_64 and aarch64 as examples:

  For example, if you want to scan the application package **NetworkManager-1.18.8-1.el7.x86_64.rpm** in the **/mnt/** directory, run the following command:

  ```
  x2hce-ca scan /mnt/NetworkManager-1.18.8-1.el7.x86_64.rpm -os_name centos7.9 -target_os_name hce2.0
  ```

  For example, if you want to scan the application package **NetworkManager-1.18.8-1.el7.aarch64.rpm** in the **/mnt/** directory, run the following command:

  ```
  x2hce-ca scan /mnt/NetworkManager-1.18.8-1.el7.aarch64.rpm -os_name EulerOSV2.0SP8arm -target_os_name hce2.0arm -arch aarch64
  ```

  NOTE

  The default value of **-arch** is **x86_64**.

- **Dir_Name1/App_Name1 Dir_Name2/App_Name2**: scans multiple application software packages.

  The following uses x86_64 as an example:

  For example, if you want to scan the application software package **grep-3.4-0.h3.r3.eulerosv2r9.x86_64.rpm** in **/opt/x2hce-ca/scan/** and the application software package **groff-1.22.4-5.h1.eulerosv2r9.x86_64.rpm** in **/opt/x2hce-ca/scan/rpm/**, run the following command:

  ```
  x2hce-ca scan /opt/x2hce-ca/scan/grep-3.4-0.h3.r3.eulerosv2r9.x86_64.rpm /opt/x2hce-ca/scan/rpm/groff-1.22.4-5.h1.eulerosv2r9.x86_64.rpm -os_name centos7.9 -target_os_name hce2.0
  ```

- **-b Dir_Name**: scans all application packages in a single directory.

  For example, if you want to scan all application packages in **directory1**, run the following command:

  ```
  x2hce-ca scan -b directory1 -os_name centos7.9  -target_os_name hce2.0
  ```

- **-b Dir_Name1 Dir_Name2**: scans all application packages in multiple directories.

  For example, if you want to scan all application packages in **directory1** and **directory2**, run the following command:

  ```
  x2hce-ca scan -b directory1 directory2 -os_name centos7.9 -target_os_name hce2.0
  ```

  NOTE

  A single directory can contain a maximum of 750 files of up to 900 MB. Excessive software packages may cause tool failures.

- **-l rpm_Name**: scans a piece of locally installed software.

  For example, to scan OpenSSL, run the following command:

  ```
  x2hce-ca scan -l openssl
  ```

- **-l rpm_Name1,rpm_Name2...**: scans more than one piece of locally installed software.

  For example, to scan OpenSSH and OpenSSL, run the following command:

  ```
  x2hce-ca scan -l openssl,openssh
  ```

  NOTE

  -l is available for x2hce-ca only in CentOS.

Table 4-5 Type

Parameter	Type	Description
-os_name	String	Source OS. This parameter is optional. The default value is **centos7.9**. For details about other parameters, see **Overview of x2hce-ca**. For example, if this parameter is set to **centos8.2**, CentOS 8.2 is selected as the source OS.
-target_os_name	String	Target OS. This parameter is optional. The default value is **hce2.0** for HCE 2.0. For other parameters, see **Overview of x2hce-ca**. For example, if this parameter is set to **-target_os_name hce1.1**, the target OS is Huawei Cloud EulerOS 1.1.

3. Analyze the evaluation results.

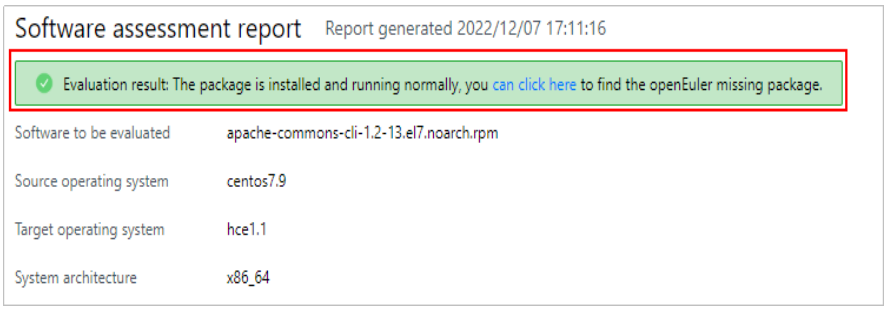
   For example, after the x2hce-ca tool scans three RPM packages in the **/tmp/x2hce-ca_test** directory, it displays the following outputs.

   Figure 4-2 Scan result

   ```
   [x2hce_ca@localhost ~]$ x2hce_ca scan -b /tmp/x2hce_ca_test/ -os_name centos7.9 -target_os_name hce1.1
   2022-08-31 04:21:59,808 - USER_ID:1001 - INFO - Log save directory: /var/log/x2hce_ca
   2022-08-31 04:21:59,812 - USER_ID:1001 - INFO - x2hce_ca scan /tmp/x2hce_ca_test/ -os_name centos7.9  -target_os_name hce1.1 -arch x86_64 -b
   2022-08-31 04:21:59,849 - USER_ID:1001 - INFO - Start analyse /tmp/x2hce_ca_test/qt-4.8.7-8.el7.x86_64.rpm
   2022-08-31 04:22:01,993 - USER_ID:1001 - INFO - Start scanning Jar Package...
   2022-08-31 04:22:01,993 - USER_ID:1001 - INFO - No jars found
   2022-08-31 04:22:02,681 - USER_ID:1001 - INFO - Start analyse /tmp/x2hce_ca_test/texinfo-5.1-5.el7.x86_64.rpm
   2022-08-31 04:22:03,547 - USER_ID:1001 - INFO - Start scanning Jar Package...
   2022-08-31 04:22:03,547 - USER_ID:1001 - INFO - No jars found
   2022-08-31 04:22:04,294 - USER_ID:1001 - INFO - Start analyse /tmp/x2hce_ca_test/zip-3.0-11.el7.x86_64.rpm
   2022-08-31 04:22:04,654 - USER_ID:1001 - INFO - Start scanning Jar Package...
   2022-08-31 04:22:04,654 - USER_ID:1001 - INFO - No jars found
   2022-08-31 04:22:04,906 - USER_ID:1001 - INFO - The excel report is saved: /opt/x2hce_ca/output/software/batch-20220831042159/batch-2022083104
   2159.xls
   2022-08-31 04:22:05,069 - USER_ID:1001 - INFO - Generate Success! The Archive file results are saved: /opt/x2hce_ca/output/software/batch-2022
   0831042159.tar.gz
   2022-08-31 04:22:05,069 - USER_ID:1001 - INFO - Analyse finished, total 3 items, 3 success, 0 failed
   ```

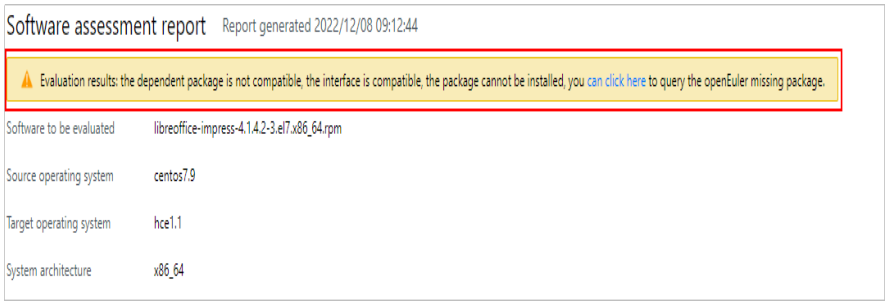
   - Download the compatibility evaluation reports from the **/opt/x2hce-ca/output/software/** directory.
     - If the software package is compatible with the target OS, the following message is displayed.

       Figure 4-3 Evaluation result for software package compatibility

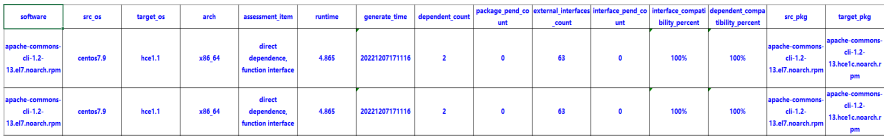
       If the software package is not compatible with the target OS, the following message is displayed.

Figure 4-4 Evaluation result for software package incompatibility



- For details about the compatible dependency packages and APIs, see the .xlsx file with the same name as the software package.

Figure 4-5 Dependency package compatibility and API compatibility



software	src_os	target_os	arch	assessment_item	runtime	generate_time	dependent_count	package_pend_count	external_interfaces_count	interface_pend_count	interface_compatibility_percent	dependent_compatibility_percent	src_pkg	target_pkg
apache-commons-cli-1.2-13.el7.noarch.rpm	centos7.9	hce1.1	x86_64	direct dependence, function interface	4.865	20221207171116	2	0	63	0	100%	100%	apache-commons-cli-1.2-13.el7.noarch.rpm	apache-commons-cli-1.2-13.hce1c.noarch.rpm
apache-commons-cli-1.2-13.el7.noarch.rpm	centos7.9	hce1.1	x86_64	direct dependence, function interface	4.865	20221207171116	2	0	63	0	100%	100%	apache-commons-cli-1.2-13.el7.noarch.rpm	apache-commons-cli-1.2-13.hce1c.noarch.rpm

– If application packages fail the check, view the Excel report in the */opt/x2hce-ca/output/software/* directory.

4.2 Migrating an OS to HCE 2.0

4.2.1 Constraints

- During the OS migration, RPM packages need to be uninstalled, installed, and updated. As a result, the OS may restart unexpectedly. Before the migration, the system disk is automatically backed up. You can also **create a server backup** to manually back up the system disk.
- The available OS memory should be larger than 128 MB, the available system disk space (required for running the migration tool) be larger than 5 GB, and the available space of the boot partition be larger than 200 MB.
- The name of a custom RPM package must be different from that of an OS RPM package. Otherwise, the custom RPM package will be deleted by the migration tool during the migration.
- After the OS migration, the system disk type cannot be changed.
- Packages in the source OS may conflict with the target OS. These packages will be automatically deleted by the migration tool. For the list of conflicting packages, see **Conflicting Packages**.
- DNF is used during OS migration. If the DNF version in the source OS is too early, the migration will be affected. Upgrade it in the source OS before the migration.

4.2.2 Migrating Procedure

Before the Migration

1. Read **OS migration and change** carefully and determine whether to migrate or change an OS. For details about OS change, see .

2. Test the compatibility of your software and HCE and ensure that they are compatible with each other so that services can still run properly after the migration. The compatibility varies depending on the software version.

3. Back up the OS and data to prevent service loss due to unexpected reasons.

Preparing Dependent Packages

Prepare the software packages required by the migration tool:

1. Remotely connect to the source OS.

   Remotely log in to the ECS to be migrated and verify that it can be accessed from the Internet. For details, see **Login Overview**.

2. Check that the migration tool can access the HCE repository to obtain the dependent packages.

   The migration tool can access the HCE repository in the output of **curl https://repo.huaweicloud.com/hce/2.0/os/x86_64/**. If information similar to the following is displayed, the repository can be accessed:

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total   Spent    Left  Speed
100  3417    0  3417    0     0    373      0 --:--:--  0:00:09 --:--:--   696
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title></title>
<link rel="stylesheet"  href="/repository/static/css/style.css" type="text/css"/>
<style>
* {
font-family: 'Verdana', sans-serif;
margin: 0;
padding: 0;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}
......
```

3. Configure the repository of the source OS and ensure that the migration tool can obtain dependent software from this repository.

   The repository address depends on the OS.

4. Install dependent packages.

   a. Install Python software packages.

```
[root@localhost ~]# yum install -y python //Run this command in any directory.
```

   b. (Optional) Create a symbolic link.

      NOTE

      The following steps are only available for CentOS 8 and EulerOS 2.10 and 2.9.

      i. Install Python 3.0 software packages.


```
[root@localhost ~]# yum install -y python3 // Run this command in any directory.
```

ii. Check whether any Python symbolic link exists.

```
[root@localhost]# ll /usr/bin/ | grep python
```

- If **python -> /usr/bin/python3** is returned but **python3 -> python3.9** is not, there is a Python symbolic link but it is not linked to Python 3. Run the following command to delete this link and then **create a Python symbolic link**:

```
[root@localhost]# unlink /usr/bin/python
```

- If both **python -> /usr/bin/python3** and **python3 -> python3.9** are returned, you do not need to create a Python symbolic link. Go to **Installing the Migration Tool and Checking Migration Conditions**.
- If neither **python -> /usr/bin/python3** nor **python3 -> python3.9** is returned, there is no Python symbolic link. You need to **create a Python symbolic link**.

iii. Create a Python symbolic link.

```
[root@localhost]# python
-bash: /usr/bin/python: No such file or directory   //Indicates that the Python symbolic link does not exist.
[root@localhost]# cd /usr/bin/      //Switch to the /usr/bin directory.
[root@localhost bin]# ln -s python3 python   // Create a Python symbolic link.
[root@localhost bin]# python
Python 3.6.8 (default, Apr 16 2020, 01:36:27)
[GCC 8.3.1 20191121 (Red Hat 8.3.1-5)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
//Press Ctrl+D to exit.
```

Installing the Migration Tool and Checking Migration Conditions

1. Download tool package **centos2hce2-*.rpm** of the latest version from the Huawei Cloud open-source image site.
   - If **dnf** can be used in your OS (for example, CentOS 8.2), run the following command to download the latest tool package:

     ```
     [root@localhost test]# yum download centos2hce2 --nogpgcheck --repofrom=centos2hce2_repo,https://repo.huaweicloud.com/hce/2.0/updates/x86_64/  // Download centos2hce2-*.rpm.
     [root@localhost test]#  ls  // Check whether the download is successful. The software version is an example here.
     centos2hce2-1.0.0-0.0.82.hce2.x86_64.rpm
     ```

   - If only **yum** can be used in your OS (for example, CentOS 7.9), run the following command to download the latest tool package:

     ```
     [root@localhost test]# echo -e "[centos2hce2]\nname=centos2hce2_repo\nbaseurl=https://repo.huaweicloud.com/hce/2.0/updates/x86_64/\nenabled=1\ngpgcheck=0" > /etc/yum.repos.d/centos2hce2.repo && yum install centos2hce2 --downloadonly --downloaddir=. && rm -f /etc/yum.repos.d/centos2hce2.repo && yum makecache  // Download centos2hce2-*.rpm.
     [root@localhost test]# ls  // Check whether the download is successful. The software version is an example here.
     centos2hce2-1.0.0-0.0.82.hce2.x86_64.rpm
     ```

2. Install the migration tool.

   ```
   [root@localhost test]# rpm -ivh centos2hce2-1.0.0-0.0.82.hce2.x86_64.rpm --nodeps // Replace the version number with the actual one.
   warning: centos2hce2-1.0.0-0.0.82.hce2.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID a8def926: NOKEY
   Verifying...                          ################################# [100%]
   Preparing...                          ################################# [100%]
   Updating / installing...
   1:centos2hce2-1.0.0-0.0.6.hce2     ################################# [100%]
   ```


3. Configure the backup directory for the system disk of the source OS.

   Before the OS migration, the migration tool automatically backs up all data of the system software to the backup directory.

   You can run **vim /etc/centos2hce2.conf** to configure the **backup_dir** field in the **centos2hce2.conf** file. The default value of **backup_dir** is **/mnt/sdb/.osbak**.

   ```
   # backup dir
   backup_dir = "/mnt/sdb/.osbak"  #Change the backup directory.
   ```

   NOTE

   - To prevent system data from being lost during the migration, configure a backup directory.
   - During OS migration, the migration tool checks the space of the backup directory. To prevent check failures caused by insufficient space, use an independent data disk (for example, /dev/sdb/ mounted to /mnt/sdb/) as the backup directory.
   - Do not use the tmpfs file system (such as **/dev** and **/run**) as the backup directory. If such a tmpfs file system is used as the backup directory, files in the file system will be lost after the system is restarted.

4. Configure the migration parameters.

   a. Set Web migration.

      To perform a Web migration, the system needs to download the RPM package. The network cannot be disconnected during the download.

      In the **centos2hce2.conf** configuration file, configure the parameters by referring to the parameter descriptions.

      ```
      [repo_relation]
      .....
      # default yum source, val: web or iso
      default_yum_source = 'web'
      .....
      # if web as source, web link config as follow
      web_link_dir = "https://repo.huaweicloud.com/hce/2.0/os/x86_64/;https://
      repo.huaweicloud.com/hce/2.0/updates/x86_64/"
      ```

      Table 4-6 Parameter settings for a Web migration

Parameter	Description
default_yum_source	Set this parameter to **web**.

Parameter	Description
web_link_dir	Specifies the addresses of base repository and updates repository for HCE. Separate multiple repositories with semicolons (;). This is an example: https://repo.huaweicloud.com/hce/2.0/os/x86_64/;https://repo.huaweicloud.com/hce/2.0/updates/x86_64/. The HCE repository will automatically replace the repository of the original OS during the migration. After the migration is complete, the original repository will be restored. So, both the new and old repository files will exist in the system. You are advised to delete the old ones from the **/etc/yum.repos.d/** directory and retain only those of HCE.

b. Configure the **isclose_modules** parameter, which is required only for CentOS 8.

CentOS 8 allows you to batch install RPM packages as a module, but HCE does not. Therefore, you need to disable the module function before performing an OS migration.

- **yes** (default): The system closes its modules before the migration.
- **no**: The system does not enable its modules before the migration. If any module is enabled, the migration is interrupted.

```
[system]
# whether close modules, if value is no, system may be not migrate
isclose_modules = "yes"
```

NOTE

- You can run **dnf module list** to view all running modules.
- You can run **dnf module list | grep '\[e\]'** to view enabled modules.

5. Run **centos2hce2.py --check all** to check whether the current OS can be migrated.
   - If the message "Environment check passed!" is displayed, continue with the migration.
   - If the message "call migration failed" is displayed, perform step **6** to handle the exceptions. **Table 4-7** provides the error numbers and the corresponding error messages.

Table 4-7 Error numbers

Error Number	Message
10001	When running migration tool commands as a non-root user, you need to switch to user **root**.

Error Number	Message
10002	The URL is invalid. The **web_link_dir** and **web_link_tar** parameters in the **/etc/centos2hce2.conf** configuration file are configured incorrectly. As a result, the corresponding repo and RPM files cannot be downloaded or connected.
10003	Basic commands, such as **rpm**, **yum**, and **yumdownloader**, are missing.
10004	The space check failed. The disk space or memory size is insufficient.
10005	The source OS does not have a local yum repository or the yum repository is unreachable. You need to reconfigure the yum repository.
10006	The yum source configuration of the source OS is incorrect. Check the **web_link_dir** parameter in the **/etc/centos2hce2.conf** configuration file.
10007	The sut installation failed. Check the **web_link_dir** parameter in the **/etc/centos2hce2.conf** configuration file.
10008	The sut check failed.
10009	The dependency check failed. Run the **centos2hce2.py --install all** command to install the dependencies.
10010	In chroot upgrade, clearing the existing chroot directory failed. Check the **chroot_path** parameter in the **/etc/centos2hce2.conf** configuration file.
10011	The chroot path is configured incorrectly. Check the **chroot_path** parameter in the **/etc/centos2hce2.conf** configuration file.
10012	In chroot upgrade where the address for downloading the TAR package in the pre-built environment is configured, decompressing the TAR package failed. Check the **web_link_tar** parameter in the **/etc/centos2hce2.conf** configuration file.
10013	Checking the **/etc/ld.so.conf** file failed. Clear the parameters other than **include ld.so.conf.d/*.conf** in the **/etc/ld.so.conf** configuration file.
10014	The file system is damaged or abnormal and needs to be repaired.

Error Number	Message
10015	The mount directory of the **/etc/fstab** file does not meet the requirements. You need to mount the file system partitions that are not in the LVM volume format in the **/etc/fstab** file using UUIDs.
10016	After the file attribute check is enabled, files with the **Immutable/Append_Only** attribute exist in the system. Such files must be added to the **exclude_dir** parameter in the **/etc/centos2hce2.conf** configuration file.
10017	The **/etc/sysconfig/ntpd** file contains the **-u ntp:ntp** configuration. You need to delete the **-u ntp:ntp** parameter from the **/etc/sysconfig/ntpd** file.
10018	The **/etc/ssh/sshd_config** configuration file contains algorithms that are not supported by HCE 2.0. Delete these algorithms as prompted.
10019	There are duplicate RPM packages in the system. Uninstall the RPM packages of earlier versions that are no longer used and check again. (If you do not want to uninstall duplicate packages, such as kernel and kernel-devel in multi-kernel scenarios, you can set **extra_check_switch** to **false** in the **/etc/centos2hce2.conf** file to skip extra check.)

6. Install software dependencies for the migration tool.

   Run **centos2hce2.py --install all** to back up data in the system, install software dependencies, and complete preprocessing.

   If the following information is displayed, the software dependencies have been installed and preprocessing has been completed. You need to perform step **5** again to check the environment.

   ```
   2022-08-19 03:12:58,373-INFO-centos2hce2.py-[line:832]: Dependency packages already exist!
   2022-08-19 03:12:58,373-INFO-centos2hce2.py-[line:891]: migrate install depend options finished
   ```

   Modify the **centos2hce2.conf** file as follows:

   ```
   [check_config]
   ...
   # backup switch before upgrade
   backup_available = ""
   ```

   backup_available determines whether to perform backup in the installation phase. The default value is an empty string.

   - If **backup_available** is empty or a non-**false** value, backup will be performed in the installation phase.
   - If **backup_available** is **false** (case-insensitive), backup will not be performed in the installation phase but performed in the upgrade phase.
   - For CentOS 8.0 or later, **centos2hce2.py --install all** will back up data and then disable the package management system of CentOS. If you set **backup_available** to **true**, the package management system will still be in the disabled state after a rollback to the original OS.

7. (Optional) Repeat the backup.

Run **centos2hce2.py --backup force** to back up the files in the system to the directory configured in step **3**.

NOTE

The software dependencies installed in step **6** will also be backed up after this command is executed.

Migrating an OS to HCE

1. Run **centos2hce2.py --upgrade all** to migrate an OS to HCE.

   If **migrate success** is displayed, the OS migration is successful. If the migration fails, perform a rollback by referring to **1**.

   Figure 4-6 Migrating an OS

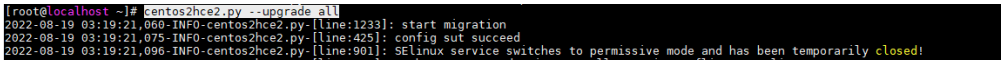
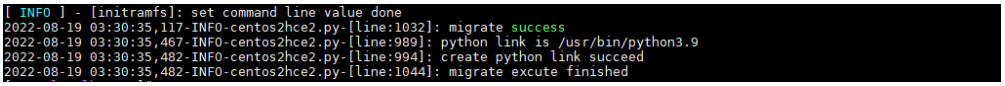


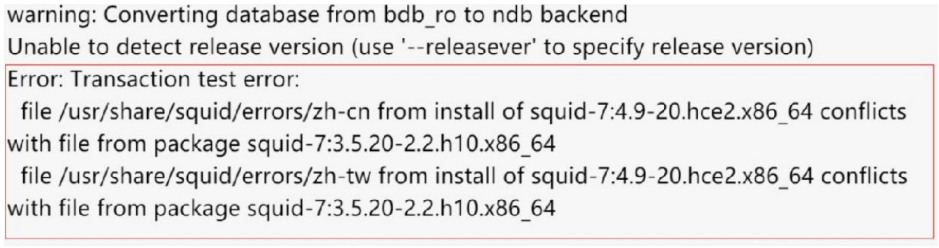
   Figure 4-7 Migration succeeded



   NOTE

   - The migration command cannot be executed in the background in Linux.
   - The **--simple_name** parameter can be added so that the abbreviation of Huawei Cloud EulerOS is displayed in the grub menu after the migration.
   - If an upgrade failed due to network interruption or software package conflicts, run the migration command again.
   - If the error message in **Figure 4-8** is displayed during an upgrade, the upgrade is interrupted due to package conflicts. Handle the conflicts and try the upgrade again. For details about how to handle package conflicts, see **Conflicting Packages**.

   Figure 4-8 An error reported when there is a conflict packet

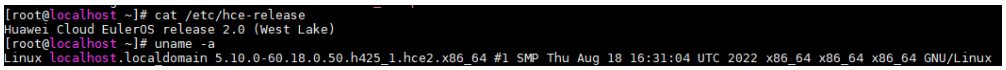


2. Reboot the server by running **reboot**. If the **reboot** command does not respond, run **reboot -f** instead.

   After the system reboot, run **cat /etc/hce-release** to view the OS information and run **uname -a** to view the OS kernel information.

   If **Huawei Cloud EulerOS** is displayed, the OS migration is successful. Otherwise, the migration failed. Contact technical support.

Figure 4-9 Checking the OS and kernel after the migration



NOTE

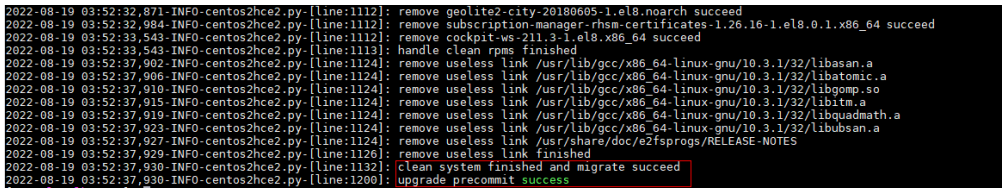
After the OS is migrated to HCE, the name of the original OS is still displayed on the console.

3. Delete the files of the source OS's components.

   After the OS migration, the components of the source OS will be replaced by those of HCE. However, the component files of the source OS are still stored in the system. You need to run **centos2hce2.py --precommit upgrade** to delete such files.

   If the message "upgrade precommit success" is displayed, the files have been successfully deleted.

   Figure 4-10 Deleting component files of the source OS



NOTE

The deletion can be performed for multiple times.

4. (Optional) Modify Cloud-Init configurations.
   - Skip this step if Cloud-Init is running normally in the source OS and Cloud-Init is an RPM package.
   - If Cloud-Init is running normally in the source OS and Cloud-Init is a file (for example, the OS is CentOS 7) other than an RPM package, modify **/etc/cloud/cloud.cfg** as follows:

   a. Enable remote login using the password for user **root** and allow SSH access to **root**.

      Set **disable_root** to **0** to keep **root** enabled. Set **ssh_pwauth** to **1** to allow remote login using a password. Set **lock_passwd** to **False** to not lock the password.

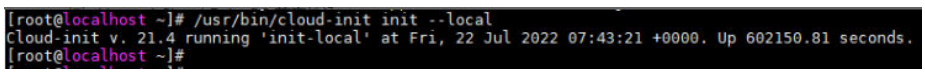
```
users:
 - name: root
   lock_passwd: False

disable_root: 0
ssh_pwauth: 1
```

   b. Run **/usr/bin/cloud-init init --local**.

      If there are no errors and the Cloud-Init version is displayed, Cloud-Init has been correctly configured.

      Figure 4-11 Cloud-Init configured successfully



c. If Cloud-Init is unavailable after the upgrade, reinstall Cloud-Init. For details, see **Installing Cloud-Init**.

5. (Optional) If SELinux service is disabled during the migration but needs to be enabled after the migration, run **centos2hce2.py --precommit upg-selinux** to enable the SELinux service. This command is executed twice. After each execution, the system restarts.

   a. Run **centos2hce2.py --precommit upg-selinux**.

   ```
   [root@localhost ~]# centos2hce2.py --precommit upg-selinux
   2022-08-21 23:46:23,891-INFO-centos2hce2.py-[line:1239]: precommit migration
   2022-08-21 23:46:23,891-INFO-centos2hce2.py-[line:1149]: begin to set selinux
   2022-08-21 23:46:23,892-INFO-centos2hce2.py-[line:1157]: grub path is /boot/grub2/grub.cfg
   2022-08-21 23:46:23,895-INFO-centos2hce2.py-[line:1162]: sed selinux succeed
   2022-08-21 23:46:23,897-INFO-centos2hce2.py-[line:1167]: create autorelabel file succeed
   2022-08-21 23:46:23,901-INFO-centos2hce2.py-[line:1172]: modify selinux config succeed
   2022-08-21 23:46:23,901-INFO-centos2hce2.py-[line:1174]: create phase 1 flag file succeed
   2022-08-21 23:46:23,901-INFO-centos2hce2.py-[line:1184]: selinux has been set, please reboot
   now
   2022-08-21 23:46:23,901-INFO-centos2hce2.py-[line:1206]: upgrade precommit selinux success
   [root@localhost ~]# reboot
   ```

   b. After the system is restarted, run **centos2hce2.py --precommit upg-selinux** again.

   ```
   [root@localhost ~]# centos2hce2.py --precommit upg-selinux
   2022-08-21 23:57:07,576-INFO-centos2hce2.py-[line:1239]: precommit migration
   2022-08-21 23:57:07,576-INFO-centos2hce2.py-[line:1176]: now begin to set selinux phase 2
   2022-08-21 23:57:07,580-INFO-centos2hce2.py-[line:1181]: modify selinux config succeed
   2022-08-21 23:57:07,580-INFO-centos2hce2.py-[line:1183]: create phase 2 flag file succeed
   2022-08-21 23:57:07,580-INFO-centos2hce2.py-[line:1184]: selinux has been set, please reboot
   now
   2022-08-21 23:57:07,580-INFO-centos2hce2.py-[line:1206]: upgrade precommit selinux success
   [root@localhost ~]# reboot
   ```

   c. After the second restart, run **getenforce** to check the SELinux status. If it is **Enforcing**, SELinux has been enabled.

   ```
   [root@localhost ~]# getenforce
   Enforcing
   ```

6. (Optional) After the migration is complete, delete the source OS data.

   After the migration, the system data of the source OS is still stored in the new system and occupies a large amount of memory. You can run **centos2hce2.py --commit all** to clear the data.

   The system will automatically delete the system data of the source OS, including the system data in the backup directory mentioned in step **3**.

   NOTICE

   After the command is executed, the OS cannot be rolled back.

   ```
   [root@localhost ~]# centos2hce2.py --commit all
   2022-08-22 04:45:32,601-INFO-centos2hce2.py-[line:1242]: commit migration
   ```

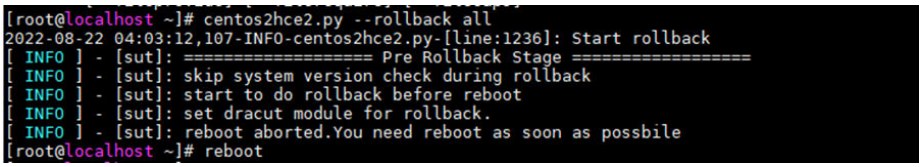
Rolling Back the OS

1. Roll back the OS if needed.

   The migration can be rolled back. You can determine whether to roll back to the original OS as required.

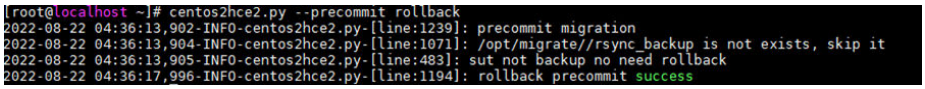
   a. Run **centos2hce2.py --rollback all** to roll back the system. After the rollback, run **reboot** to restart the system.

Figure 4-12 System rollback and restart



b. Run **centos2hce2.py --precommit rollback** to restore the environment.

Figure 4-13 Environment restoration



2. (Optional) If SELinux has been enabled before the migration, the SELinux service will be automatically disabled during the migration. If necessary, manually enable the SELinux status after the rollback.

   a. Run **centos2hce2.py --precommit rbk-selinux**.

   ```
   [root@localhost ~]# centos2hce2.py --precommit rbk-selinux
   2022-09-05 03:58:37,015-INFO-centos2hce2.py-[line:1401]: precommit migration
   2022-09-05 03:58:37,047-INFO-centos2hce2.py-[line:1319]: now begin to set selinux
   2022-09-05 03:58:37,051-INFO-centos2hce2.py-[line:1324]: modify selinux config succeed
   2022-09-05 03:58:37,051-INFO-centos2hce2.py-[line:1325]: selinux has been set, please reboot
   now
   2022-09-05 03:58:37,051-INFO-centos2hce2.py-[line:1340]: set rollback selinux succeed
   2022-09-05 03:58:37,051-INFO-centos2hce2.py-[line:1365]: upgrade precommit selinux success
   ```

   b. Run **reboot** to restart the system.

   ```
   [root@localhost ~]# reboot
   ```

   c. After the system is restarted, you can see that SELinux is enabled.

   ```
   [root@localhost ~]# getenforce
   Enforcing
   ```

3. Clear data from the OS.

   Run **centos2hce2.py --commit all** to clear the data.

   The system will automatically delete the system data of the source and target OSs, including the system data in the backup directory mentioned in step 3.

   ```
   [root@localhost ~]# centos2hce2.py --commit all
   2022-08-22 04:45:32,601-INFO-centos2hce2.py-[line:1242]: commit migration
   ```

4.2.3 Conflicting Packages

NOTE

- Conflicting packages are software packages that conflict with HCE in the source OS, which affects the upgrade.
- Conflicting packages will be automatically uninstalled during the upgrade and will not be installed again. Before the upgrade, check whether the software packages on which the source OS depends are in the conflicting package list to prevent software loss after the upgrade.
- If a software package is lost after the upgrade, run the **yum** command to install the software package of the new version.
- If other conflict problems occur during the upgrade, you can modify the **/etc/centos2hce2.conf** configuration file and add custom conflicting package names by referring to the conflicting package list in this section.

Table 4-8 Conflicting packages in CentOS 8 series

CentOS Version	Conflicting Packages
CentOS 8.0	rust-doc;intel-gpu-tools;netcf-libs;redhat-rpm-config;asciidoc;gnuplot-common;perf;tigervnc-icons;libpq-devel;paratype-pt-sans-caption-fonts;scala-apidoc;java-11-openjdk-devel;java-11-openjdk-headless;java-1.8.0-openjdk-headless;dovecot;systemd-journal-remote;pcp-manager;pcp-webapi;libguestfs-java-devel;libguestfs-javadoc;icedtea-web-javadoc;systemtap-runtime-java;java-1.8.0-openjdk-accessibility;java-1.8.0-openjdk-demo;ant;tigervnc-server-applet;java-atk-wrapper;java-11-openjdk;guava20;javapackages-tools;jboss-jaxrs-2.0-api;maven-shared-utils;tagsoup;cdi-api;libbase;geronimo-annotation;pentaho-reporting-flow-engine;maven-resolver-api;apache-commons-codec;maven-lib;jansi-native;maven-wagon-provider-api;libguestfs-java;apache-commons-cli;istack-commons-tools;jline;plexus-cipher;istack-commons-runtime;jcl-over-slf4j;apache-commons-io;maven-resolver-spi;maven-wagon-file;httpcomponents-core;icedtea-web;glassfish-el-api;aopalliance;hawtjni-runtime;plexus-containers-component-annotations;flute;jboss-annotations-1.2-api;liblayout;java-1.8.0-openjdk;postgresql-jdbc;mariadb-java-client;plexus-sec-dispatcher;google-guice;libformula;jdeparser;ant-lib;maven-wagon-http-shared;jboss-logging;plexus-classworlds;slf4j;librepository;ongres-scram-client;sisu-plexus;libfonts;plexus-interpolation;java-1.8.0-openjdk-src;plexus-utils;scala-swing;maven-wagon-http;ongres-scram;maven-resolver-impl;libloader;httpcomponents-client;atinject;apache-commons-logging;maven-resolver-connector-basic;jansi;jsoup;maven-resolver-util;jboss-interceptors-1.2-api;libreoffice-ure;byteman;sac;apache-commons-lang3;libserializer;scala;maven-resolver-transport-wagon;jboss-logging-tools;sisu-inject;libreoffice-core;java-1.8.0-openjdk-devel

CentOS Version	Conflicting Packages
CentOS 8.1	kernel-rpm-macros;intel-gpu-tools;netcf-libs;redhat-rpm-config;asciidoc;gnuplot-common;perf;tigervnc-icons;libpq-devel;paratype-pt-sans-caption-fonts;java-1.8.0-openjdk-headless;java-11-openjdk-headless;java-11-openjdk-devel;pcp-pmda-rpm;pcp-pmda-podman;scala-apidoc;libguestfs-java-devel;libguestfs-javadoc;icedtea-web-javadoc;systemtap-runtime-java;java-1.8.0-openjdk-accessibility;java-1.8.0-openjdk-demo;ant;tigervnc-server-applet;java-atk-wrapper;java-11-openjdk;jansi-native;hawtjni-runtime;ongres-scram;jboss-annotations-1.2-api;liblayout;atinject;plexus-utils;istack-commons-tools;jline;apache-commons-io;ongres-scram-client;maven-shared-utils;maven-resolver-impl;libfonts;jsoup;apache-commons-codec;glassfish-el-api;jdeparser;maven-resolver-util;scala-swing;tagsoup;google-guice;istack-commons-runtime;jcl-over-slf4j;pentaho-reporting-flow-engine;maven-resolver-api;maven-resolver-connector-basic;libloader;slf4j;apache-commons-cli;maven-wagon-provider-api;maven-resolver-transport-wagon;byteman;httpcomponents-client;jna;java-1.8.0-openjdk-devel;maven-lib;libreoffice-core;java-1.8.0-openjdk-src;javapackages-tools;plexus-cipher;cdi-api;jboss-logging;sisu-inject;httpcomponents-core;guava20;sac;libbase;jboss-jaxrs-2.0-api;java-1.8.0-openjdk;libserializer;plexus-containers-component-annotations;jboss-interceptors-1.2-api;jboss-logging-tools;libguestfs-java;ant-lib;libreoffice-ure;maven-resolver-spi;maven-wagon-file;jansi;maven-wagon-http-shared;apache-commons-lang3;postgresql-jdbc;mariadb-java-client;plexus-sec-dispatcher;sisu-plexus;scala;plexus-classworlds;flute;maven-wagon-http;icedtea-web;libformula;plexus-interpolation;aopalliance;geronimo-annotation;librepository;apache-commons-logging

CentOS Version	Conflicting Packages
CentOS 8.2	python-psycopg2-doc;exiv2;llvm-googletest;adwaita-qt;llvm-static;rust-doc;intel-gpu-tools;netcf-libs;flatpak-session-helper;asciidoc;perf;tigervnc-icons;paratype-pt-sans-caption-fonts;java-1.8.0-openjdk-headless;java-11-openjdk-devel;java-11-openjdk-headless;scala-apidoc;libguestfs-java-devel;libguestfs-javadoc;icedtea-web-javadoc;systemtap-runtime-java;java-1.8.0-openjdk-accessibility;java-1.8.0-openjdk-demo;ant;tigervnc-server-applet;java-atk-wrapper;java-11-openjdk;jboss-annotations-1.2-api;cdi-api;ongres-scram;maven-resolver-util;apache-commons-codec;istack-commons-tools;icedtea-web;plexus-classworlds;plexus-utils;maven-wagon-http-shared;atinject;javapackages-tools;istack-commons-runtime;jline;geronimo-annotation;jansi;jdeparser;byteman;liblayout;maven-resolver-transport-wagon;jmc-core;ant-lib;libreoffice-core;jansi-native;jcl-over-slf4j;slf4j;ee4j-parent;libfonts;maven-wagon-http;jboss-logging;jboss-interceptors-1.2-api;tagsoup;httpcomponents-client;plexus-containers-component-annotations;apache-commons-lang3;jaf;java-1.8.0-openjdk-src;jsoup;guava20;flute;apache-commons-cli;libbase;ongres-scram-client;jboss-logging-tools;plexus-interpolation;libloader;librepository;libreoffice-ure;scala-swing;jboss-jaxrs-2.0-api;maven-resolver-spi;maven-lib;apache-commons-io;hawtjni-runtime;google-guice;aopalliance;libguestfs-java;postgresql-jdbc;jna;glassfish-el-api;maven-resolver-impl;java-1.8.0-openjdk;directory-maven-plugin;mariadb-java-client;httpcomponents-core;maven-wagon-file;maven-wagon-provider-api;owasp-java-encoder;libserializer;maven-shared-utils;plexus-cipher;java-1.8.0-openjdk-devel;plexus-sec-dispatcher;pentaho-reporting-flow-engine;maven-resolver-api;sac;scala;libformula;sisu-inject;apache-commons-logging;maven-resolver-connector-basic;sisu-plexus;centos-logos-httpd;pcp-pmda-rpm

CentOS Version	Conflicting Packages
CentOS 8.3	netcf-libs;rust-doc;git-credential-libsecret;texlive-context;intel-gpu-tools;flatpak-session-helper;asciidoc;perf;tigervnc-icons;paratype-pt-sans-caption-fonts;java-1.8.0-openjdk-headless;java-11-openjdk-devel;java-11-openjdk-headless;libguestfs-java-devel;libguestfs-javadoc;icedtea-web-javadoc;systemtap-runtime-java;java-1.8.0-openjdk-accessibility;java-1.8.0-openjdk-demo;ant;tigervnc-server-applet;java-atk-wrapper;java-11-openjdk;exiv2;llvm-googletest;adwaita-qt;llvm-static;python-psycopg2-doc;scala-apidoc;libXau;libappstream-glib;jmc-core;byteman;libfonts;jaf;jcl-over-slf4j;mariadb-java-client;tagsoup;libguestfs-java;jsoup;apache-commons-cli;sisu-inject;jansi-native;jna;apache-commons-lang3;flute;librepository;javapackages-tools;cdi-api;ongres-scram;java-1.8.0-openjdk-devel;sisu-plexus;istack-commons-runtime;jboss-logging;guava20;java-1.8.0-openjdk-src;maven-resolver-util;geronimo-annotation;hawtjni-runtime;jboss-annotations-1.2-api;ongres-scram-client;maven-resolver-connector-basic;slf4j;sac;apache-commons-codec;atinject;maven-wagon-http;libreoffice-ure;plexus-cipher;jboss-interceptors-1.2-api;jline;pentaho-reporting-flow-engine;httpcomponents-core;liblayout;istack-commons-tools;jdeparser;maven-wagon-provider-api;ee4j-parent;apache-commons-io;maven-resolver-spi;jboss-logging-tools;plexus-sec-dispatcher;plexus-containers-component-annotations;jboss-jaxrs-2.0-api;scala;libbase;libreoffice-core;httpcomponents-client;directory-maven-plugin;java-1.8.0-openjdk;libformula;maven-wagon-file;maven-shared-utils;aopalliance;glassfish-el-api;owasp-java-encoder;postgresql-jdbc;libloader;google-guice;plexus-classworlds;ant-lib;maven-resolver-api;plexus-interpolation;java-1.8.0-openjdk-slowdebug;maven-resolver-impl;java-1.8.0-openjdk-headless-slowdebug;prometheus-jmx-exporter;maven-resolver-transport-wagon;jolokia-jvm-agent;maven-wagon-http-shared;maven-lib;jansi;HdrHistogram;apache-commons-logging;plexus-utils;icedtea-web;libserializer;scala-swing

CentOS Version	Conflicting Packages
CentOS 8.4	python-psycopg2-doc;anaconda-install-env-deps;hwloc-gui;python3-lit;exiv2;cups-filters;cups-filters-libs;gutenprint;adwaita-qt;cups;cups-lpd;hplip-common;hwloc-libs;gutenprint-doc;gutenprint-libs;gutenprint-libs-ui;hwloc;foomatic-db-ppds;foomatic-db;python39-pip;python39-setuptools;python39-numpy;python39-chardet;python39-psutil;python39-urllib3;python39-requests;python39-wheel;libasan6;paratype-pt-sans-caption-fonts;python39-six;python39-idna;python39-ply;python39-pyyaml;python39-pycparser;python39-lxml;python39-pysocks;rust-doc;netcf-libs;git-credential-libsecret;texlive-context;flatpak-session-helper;asciidoc;intel-gpu-tools;tigervnc-icons;jmc-core;byteman;libfonts;jaf;jcl-over-slf4j;mariadb-java-client;tagsoup;libguestfs-java;jsoup;apache-commons-cli;sisu-inject;jansi-native;jna;apache-commons-lang3;flute;librepository;javapackages-tools;cdi-api;ongres-scram;java-1.8.0-openjdk-devel;sisu-plexus;istack-commons-runtime;jboss-logging;guava20;java-1.8.0-openjdk-src;maven-resolver-util;geronimo-annotation;hawtjni-runtime;jboss-annotations-1.2-api;ongres-scram-client;maven-resolver-connector-basic;slf4j;sac;apache-commons-codec;atinject;maven-wagon-http;libreoffice-ure;plexus-cipher;jboss-interceptors-1.2-api;jline;pentaho-reporting-flow-engine;httpcomponents-core;liblayout;istack-commons-tools;jdeparser;maven-wagon-provider-api;ee4j-parent;apache-commons-io;maven-resolver-spi;jboss-logging-tools;plexus-sec-dispatcher;plexus-containers-component-annotations;jboss-jaxrs-2.0-api;scala;libbase;libreoffice-core;httpcomponents-client;directory-maven-plugin;java-1.8.0-openjdk;libformula;maven-wagon-file;maven-shared-utils;aopalliance;glassfish-el-api;owasp-java-encoder;postgresql-jdbc;libloader;google-guice;plexus-classworlds;ant-lib;maven-resolver-api;plexus-interpolation;java-1.8.0-openjdk-slowdebug;maven-resolver-impl;java-1.8.0-openjdk-headless-slowdebug;prometheus-jmx-exporter;maven-resolver-transport-wagon;jolokia-jvm-agent;maven-wagon-http-shared;maven-lib;jansi;HdrHistogram;apache-commons-logging;plexus-utils;icedtea-web;libserializer;scala-swing;java-1.8.0-openjdk-headless;java-11-openjdk-devel;java-11-openjdk-headless;libguestfs-java-devel;libguestfs-javadoc;icedtea-web-javadoc;systemtap-runtime-java;java-1.8.0-openjdk-accessibility;java-1.8.0-openjdk-demo;ant;java-atk-wrapper;java-11-openjdk;scala-apidoc;libappstream-glib;PackageKit-gtk3-module;gnome-software;flatpak-libs;PackageKit-glib;PackageKit-gstreamer-plugin;libpq-devel;poppler;perf

CentOS Version	Conflicting Packages
CentOS 8.5	bluez;python-psycopg2-doc;perl-Devel-Peek;OpenIPMI-libs;anaconda-install-env-deps;postfix-mysql;perl-Devel-SelfStubber;metacity;bluez-libs;libicu;vte-profile;qt5-qttools-examples;exiv2;cups-filters;cups-filters-libs;gutenprint;gnome-session;cups;cups-lpd;hplip-common;hwloc;gnome-session-wayland-session;gutenprint-doc;gutenprint-libs;gutenprint-libs-ui;gnome-session-xsession;foomatic-db-ppds;foomatic-db;gnome-classic-session;gnome-shell-extension-apps-menu;gnome-shell-extension-auto-move-windows;gnome-shell-extension-drive-menu;gnome-shell-extension-launch-new-instance;gnome-shell-extension-native-window-placement;gnome-shell-extension-places-menu;gnome-shell-extension-screenshot-window-sizer;gnome-shell-extension-user-theme;gnome-shell-extension-window-list;gnome-shell-extension-windowsNavigator;gnome-shell-extension-workspace-indicator;python39-six;python39-idna;python39-ply;python39-pyyaml;python39-pycparser;python39-psutil;python39-urllib3;python39-lxml;python39-pysocks;xorg-x11-server-Xwayland;compat-hwloc1;bluez-obexd;bluez-hid2hci;netcf-libs;git-credential-libsecret;texlive-context;flatpak-session-helper;asciidoc;intel-gpu-tools;tigervnc-icons;libasan6;paratype-pt-sans-caption-fonts;pcp-pmda-podman;jmc-core;byteman;libfonts;jaf;jcl-over-slf4j;mariadb-java-client;tagsoup;libguestfs-java;jsoup;apache-commons-cli;sisu-inject;jansi-native;jna;apache-commons-lang3;flute;librepository;javapackages-tools;cdi-api;ongres-scram;java-1.8.0-openjdk-devel;sisu-plexus;istack-commons-runtime;jboss-logging;guava20;java-1.8.0-openjdk-src;maven-resolver-util;geronimo-annotation;hawtjni-runtime;jboss-annotations-1.2-api;ongres-scram-client;maven-resolver-connector-basic;slf4j;sac;apache-commons-codec;atinject;maven-wagon-http;libreoffice-ure;plexus-cipher;jboss-interceptors-1.2-api;jline;pentaho-reporting-flow-engine;httpcomponents-core;liblayout;istack-commons-tools;jdeparser;maven-wagon-provider-api;ee4j-parent;apache-commons-io;maven-resolver-spi;jboss-logging-tools;plexus-sec-dispatcher;plexus-containers-component-annotations;jboss-jaxrs-2.0-api;scala;libbase;libreoffice-core;httpcomponents-client;directory-maven-plugin;java-1.8.0-openjdk;libformula;maven-wagon-file;maven-shared-utils;aopalliance;glassfish-el-api;owasp-java-encoder;postgresql-jdbc;libloader;google-guice;plexus-classworlds;ant-lib;maven-resolver-api;plexus-interpolation;java-1.8.0-openjdk-slowdebug;maven-resolver-impl;java-1.8.0-openjdk-headless-slowdebug;prometheus-jmx-exporter;maven-resolver-transport-wagon;jolokia-jvm-agent;maven-wagon-http-shared;maven-lib;jansi;HdrHistogram;apache-commons-logging;plexus-utils;icedtea-web;libserializer;scala-swing;java-1.8.0-openjdk-headless;java-11-openjdk-devel;java-11-openjdk-

CentOS Version	Conflicting Packages
	headless;libguestfs-java-devel;libguestfs-javadoc;icedtea-web-javadoc;systemtap-runtime-java;java-1.8.0-openjdk-accessibility;java-1.8.0-openjdk-demo;ant;java-atk-wrapper;java-11-openjdk;scala-apidoc;libappstream-glib;PackageKit-gtk3-module;gnome-software;flatpak-libs;PackageKit-glib;PackageKit-gstreamer-plugin;coreos-installer-bootinfra;OpenIPMI;rust;cargo;perf;flatpak;hplip-libs;nautilus;gutenprint-cups;libgtop2;PackageKit;libsane-hpaio;PackageKit-command-not-found;xorg-x11-drv-wacom-serial-support;clutter;clutter-gtk;clutter-gst3;cheese-libs;cheese;gnome-initial-setup;gnome-control-center;clutter-gst2

Table 4-9 Conflicting packages in CentOS 7 series

CentOS Version	Conflicting Packages
CentOS 7.0	texlive-kpathsea-lib;libdhash;libref_array;libbasicobjects;qemu-kvm-tools;texlive-dvipdfm-bin;texlive-dvipdfm;tomcat-servlet-3.0-api;gnuplot-common;postgresql-devel;tigervnc-icons;squid;perf;dovecot;dovecot-mysql;dovecot-pgsql;dovecot-pigeonhole;lvm2-cluster
CentOS 7.1	texlive-kpathsea-lib;libdhash;libref_array;qemu-kvm-tools;texlive-dvipdfm-bin;tomcat-servlet-3.0-api;gnuplot-common;squid;tigervnc-icons;postgresql-devel;perf;dovecot;dovecot-mysql;dovecot-pgsql;dovecot-pigeonhole;lvm2-cluster;texlive-dvipdfm;libcacard
CentOS 7.2	texlive-kpathsea-lib;libdhash;qemu-kvm-tools;rdma-ndd;texlive-dvipdfm;texlive-dvipdfm-bin;dstat;tomcat-servlet-3.0-api;gnuplot-common;perf;squid;tigervnc-icons;tigervnc-icons;postgresql-devel;dovecot;dovecot-pgsql;dovecot-pigeonhole;lvm2-cluster;ipa-server-trust-ad
CentOS 7.3	spice-glib;texlive-kpathsea-lib;libdhash;qemu-kvm-tools;rdma-ndd;texlive-dvipdfm;texlive-dvipdfm-bin;dstat;tomcat-servlet-3.0-api;gnuplot-common;perf;squid;tigervnc-icons;postgresql-devel;dovecot;dovecot-mysql;dovecot-pgsql;dovecot-pigeonhole;lvm2-cluster;pcp-pmda-kvm;pcp-pmda-rpm;spice-gtk3;vinagre;ipa-server;ipa-server-trust-ad
CentOS 7.4	spice-glib;texlive-kpathsea-lib;libdhash;qemu-kvm-tools;texlive-dvipdfm-bin;texlive-dvipdfm;dstat;tomcat-servlet-3.0-api;gnuplot-common;perf;squid;tigervnc-icons;postgresql-devel;lvm2-cluster;spice-gtk3;vinagre

CentOS Version	Conflicting Packages
CentOS 7.5	spice-glib;texlive-kpathsea-lib;qemu-kvm-tools;texlive-dvipdfm-bin;texlive-dvipdfm;dstat;tomcat-servlet-3.0-api;gnuplot-common;perf;squid;tigervnc-icons;postgresql-devel;lvm2-cluster;spice-gtk3;vinagre
CentOS 7.6	shim-x64;spice-glib;adwaita-gtk2-theme;texlive-kpathsea-lib;qemu-kvm-tools;texlive-dvipdfm-bin;texlive-dvipdfm;dstat;tomcat-servlet-3.0-api;gnuplot-common;cockpit-ws;perf;squid;tigervnc-icons;postgresql-devel;java-11-openjdk-headless;lvm2-cluster;spice-gtk3;vinagre
CentOS 7.7	shim-x64;spice-glib;openmpi;adwaita-gtk2-theme;exiv2;texlive-kpathsea-lib;qemu-kvm-tools;texlive-dvipdfm-bin;texlive-dvipdfm;dstat;tomcat-servlet-3.0-api;cockpit-ws;gnuplot-common;perf;squid;tigervnc-icons;postgresql-devel;java-11-openjdk-headless;lvm2-cluster;spice-gtk3;openmpi-devel;vinagre
CentOS 7.8	shim-x64;spice-glib;openmpi;adwaita-gtk2-theme;exiv2;texlive-kpathsea-lib;qemu-kvm-tools;texlive-dvipdfm-bin;texlive-dvipdfm;dstat;tomcat-servlet-3.0-api;cockpit-ws;gnuplot-common;perf;squid;tigervnc-icons;postgresql-devel;java-11-openjdk-headless;lvm2-cluster;spice-gtk3;openmpi-devel;vinagre
CentOS 7.9	spice-glib;openmpi;adwaita-gtk2-theme;exiv2;gnuplot-common;texlive-kpathsea-lib;perf;qemu-kvm-tools;texlive-dvipdfm-bin;texlive-dvipdfm;dstat;tomcat-servlet-3.0-api;cockpit-ws;squid;tigervnc-icons;postgresql-devel;java-11-openjdk-headless;lvm2-cluster;spice-gtk3;openmpi-devel

Table 4-10 Conflicting packages in HCE

HCE	Conflicting Packages
HCE 1.1	spice-glib;openmpi;exiv2;sg3_utils;spice-gtk3;openmpi-devel;kernel-hcek;tomcat-servlet-3.0-api;kernel-hcek-devel;dstat;gnuplot-common;cockpit-ws;perf;squid;postgresql-devel;java-11-openjdk-headless;lvm2-cluster;fcoe-utils;libblockdev;udisks2;python-blivet;device-mapper-multipath;device-mapper-multipath-libs;libblockdev-crypto;libblockdev-fs;libblockdev-loop;libblockdev-mdraid;libblockdev-nvdimm;libblockdev-part;libblockdev-swap;libblockdev-utils;NetworkManager-team;NetworkManager-bluetooth;NetworkManager-wifi;libstorage-uio-static;kiwi-dlimage

Table 4-11 Conflicting packages in EulerOS

EulerOS Version	Conflicting Packages
EulerOS 2.9	euleros-release;euleros-latest-release;kiwi-systemdeps;python3-kiwi;NetworkManager-team;NetworkManager-bluetooth;NetworkManager-wifi;libstorage-uio-static;kiwi-dlimage;systemd-udev-compat
EulerOS 2.10	euleros-release;euleros-latest-release;kiwi-systemdeps;python3-kiwi;NetworkManager-team;NetworkManager-bluetooth;NetworkManager-wifi;libstorage-uio-static;kiwi-dlimage;systemd-udev-compat

Table 4-12 Configuration item for special conflicting packages

Configuration Item	Conflicting Packages
specific_conflict	openssl110f-libs;openssl110h-libs;openssl111d-libs

Software conflicting with HCE is added into the configuration item. Such software will be checked for by running **centos2hce2.py --check all**. If any of the software is detected, the migration tool will display a message, asking you to uninstall the software before the upgrade.

4.3 Migrating an OS to HCE

4.3.1 Constraints

- Only a CentOS 7.9 OS without a GUI installed can be migrated to HCE 1.1.
- During the OS migration, RPM packages need to be uninstalled, installed, and updated. As a result, the OS may restart unexpectedly. Before the migration, the system disk is automatically backed up. You can also **create a server backup** to manually back up the system disk.
- There should be at least 128 MB of available space in the memory and 1 GB on the system disk.

4.3.2 Migration Operations

Migrate CentOS 7.9 to HCE 1.1.

Preparing Dependent Packages

1. Remotely connect to the source OS.

   Remotely log in to the ECS to be migrated and verify that it can be accessed from the Internet. For details, see **Login Overview**.

2. Disable all repository configurations in **/etc/yum.repos.d** of CentOS. This ensures that the repositories of CentOS and HCE do not conflict.

   Figure 4-14 Disabling repository configurations

   ```
   [root@hce-ecs-2d53-gl yum.repos.d]# ls
   CentOS-Base.repo  CentOS-Debuginfo.repo  CentOS-Media.repo    CentOS-Vault.repo         epel.repo         epel-testing.repo
   CentOS-CR.repo    CentOS-fasttrack.repo  CentOS-Sources.repo  CentOS-x86_64-kernel.repo  epel.repo.rpmnew
   [root@hce-ecs-2d53-gl yum.repos.d]#
   ```

   Take **Centos_Base.repo** as an example. Add **enabled=0** under each item, as shown in the following figure.

   Figure 4-15 Adding the configuration item enabled=0

   ```
   [base]
   name=CentOS-$releasever - Base
   #mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
   baseurl=https://repo.huaweicloud.com/centos/$releasever/os/$basearch/
   gpgcheck=1
   enabled=0
   gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

   #released updates
   [updates]
   name=CentOS-$releasever - Updates
   #mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates&infra=$infra
   baseurl=https://repo.huaweicloud.com/centos/$releasever/updates/$basearch/
   gpgcheck=1
   enabled=0
   gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

   #additional packages that may be useful
   [extras]
   name=CentOS-$releasever - Extras
   #mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras&infra=$infra
   baseurl=https://repo.huaweicloud.com/centos/$releasever/extras/$basearch/
   gpgcheck=1
   enabled=0
   gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

   #additional packages that extend functionality of existing packages
   [centosplus]
   ```

3. Configure the repository of HCE.

   Add the following content to **hce.repo** and then store it in the **/etc/yum.repos.d/** directory:

   ```
   [centos7_everything]
   name=centos7_everything
   baseurl=https://repo.huaweicloud.com/hce/1.1/os/x86_64/
   enable=1
   gpgcheck=0
   priority=1

   #released updates
   [updates]
   name=hce1_updates
   baseurl=https://repo.huaweicloud.com/hce/1.1/updates/x86_64/
   gpgcheck=0
   enabled=1
   gpgkey=
   ```

4. Check whether CentOS 7.9 can access the repository of HCE.

   Run the **curl https://repo.huaweicloud.com/hce/1.1/os/x86_64/** command to check whether the repository of HCE can be accessed. If information similar to the following is displayed, the repository can be accessed:

   ```
   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total   Spent    Left  Speed
   100  3417    0  3417    0     0    373      0 --:--:--  0:00:09 --:--:--   696
   <!doctype html>
   <html>
   <head>
   <meta charset="utf-8">
   ```



```
<title></title>
<link rel="stylesheet"  href="/repository/static/css/style.css" type="text/css"/>
<style>
* {
font-family: 'Verdana', sans-serif;
margin: 0;
padding: 0;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}
......
```

5. Install Python 3.0.

   ```
   [root@localhost ~]#  yum install -y python 3 //Run this command in any directory you want.
   ```

   NOTE

   If Python 3 has been installed on CentOS 7.9, skip this step.

6. Disable SELinux.

   To ensure that system configuration files are consistent before and after the migration, SELinux needs to be disabled.

   a. Modify the **/etc/selinux/config** file by changing the value of **SELINUX** to **disabled**.

      ```
      SELINUX=disabled
      ```

   b. Restart the OS to apply the changes.

Installing the Migration Tool

1. Download tool package **centos2hce1-*.rpm** from the Huawei Cloud open-source image site. Contact customer service to obtain the download link from O&M engineers.

   The asterisk (*) indicates the version of the migration tool. In this example, **centos2hce1-1.0.0-0.0.2.x86_64.rpm** is used.

   ```
   [root@localhost test]# wget https://repo.huaweicloud.com/hce/1.1/updates/x86_64/Packages/centos2hce1-1.0.0-0.0.2.x86_64.rpm    //Download the centos2hce1-*.rpm package.
   [root@localhost test]# ls  //Check whether the download is successful.
   centos2hce1-1.0.0-0.0.2.x86_64.rpm
   ```

2. Install the migration tool.

   After the tool has been installed, the system automatically generates the **/etc/centos2hce1.conf** file.

   ```
   [root@localhost ~]# rpm -ivh centos2hce1-1.0.0-0.0.2.x86_64.rpm
   ```

3. Configure the **centos2hce1.conf** file.

   Configure the repository of HCE. It will be used for checking whether the repository can be accessed and updating RPM packages.

   ```
   #iso as yum source link
   [repo_info]
   base_yum_url =https://repo.huaweicloud.com/hce/1.1/os/x86_64/

   #iso as yum source
   repostr_hce1_1 =
      [base]
      name=hceversion
      baseurl=https://repo.huaweicloud.com/hce/1.1/os/x86_64/
      gpgcheck=0
      enabled=1
      #released updates
      [updates]
      name=hce1_updates
      baseurl=https://repo.huaweicloud.com/hce/1.1/updates/x86_64/
   ```



```
gpgcheck=0
enabled=1
gpgkey=
```

NOTE

To learn more about the parameters in the **centos2hce1.conf** file, see **Appendix: Description of the .conf File**.

Migrating the OS

1. Back up the OS.

   The migration to HCE 1.1 cannot be rolled back. Before performing the migration, you should back up CentOS including its system disk and data disks.

2. Run the **centos2hce1.py** command to migrate the OS.

   The migration takes 20 minutes to 1 hour, depending on the number and size of RPM packages to be updated and the download speed of RPM packages from the repository. Reserve sufficient time for the migration based on your environment.

   ```
   [root@localhost home]# centos2hce1.py
   ```

   If the following information is displayed, the migration was complete. If the migration failed, use the backup to restore data.

   Figure 4-16 Command output

   ```
   Complete!
   2023-04-18 15:57:22-centos2hce1.py [327]: redhat-lsb is replaced by system-lsb on hce1_1,centos-logs is replaced by hce-logs
   "gpg-pubkey-f4a80eb5-53a7ff4b"gpg-pubkey-352c64e5-52ae6884package %{PACKAGER}\n" is not installed
   package ! is not installed
   "grep-2.20-3.hce1cpackage CentOS is not installed
   2023-04-18 15:57:27-centos2hce1.py [347]: remove left rpms
   2023-04-18 15:57:27-centos2hce1.py [350]: remove rpm list: kernel-3.10.0-1160.el7.x86_64 kernel-devel-3.10.0-1160.el7.x86_64 ift
   op-1.0-0.21.pre4.el7.x86_64 kernel-devel-3.10.0-1160.53.1.el7.x86_64 centos-release-7-9.2009.0.el7.centos.x86_64 kernel-3.10.0-1
   160.53.1.el7.x86_64
   2023-04-18 15:57:34-centos2hce1.py [357]: Removing yum cache
   2023-04-18 15:57:41-centos2hce1.py [380]: Sync successfully, update grub.cfg.
   [root@centos7 home]# _
   ```

   NOTE

   CentOS contains some RPM packages that are not provided by HCE 1.1. After you run the **centos2hce1.py** command to migrate the OS, these RPM packages are automatically deleted. If you want to retain them, run the **-s skip** command to migrate the OS.

3. (Optional) Delete unnecessary RPM packages.

   The following two RPM packages are not used during the migration and do not affect how the system runs. You can just delete them.

   Figure 4-17 Unused RPM packages

   ```
   [root@localhost home]# ll
   total 24
   -rw-r--r--. 1 root root 15972 Jul  1 06:31 hce-release-1.1-23.hce1c.x86_64.rpm
   -rw-r--r--. 1 root root  5032 Jul  1 06:31 hce-repos-2.10-2.hce1c.x86_64.rpm
   [root@localhost home]#
   ```

   Figure 4-18 Deleting unused RPM packages

   ```
   [root@localhost home]# rm hce-release-1.1-23.hce1c.x86_64.rpm hce-repos-2.10-2.hce1c.x86_64.rpm
   rm: remove regular file ‘hce-release-1.1-23.hce1c.x86_64.rpm’? y
   rm: remove regular file ‘hce-repos-2.10-2.hce1c.x86_64.rpm’? y
   [root@localhost home]#
   ```


4. Run the **reboot** command to restart the OS.
5. Run the **cat /etc/os-release** command to check whether the migration was successful.

   If the following information is displayed, the migration was successful.

   Figure 4-19 Successful migration

   ```
   [root@localhost centos2hce1]# cat /etc/os-release
   NAME="Huawei Cloud EulerOS"
   VERSION="1.1 (x86_64)"
   ID="hce"
   VERSION_ID="1.1"
   PRETTY_NAME="Huawei Cloud EulerOS 1.1 (x86_64)"
   ANSI_COLOR="0;31"

   [root@localhost centos2hce1]#
   ```

6. (Optional) Enable SELinux.

   SELinux was disabled before the OS migration. Enable it if needed.

   a. Modify the **/etc/selinux/config** file by changing the value of **SELINUX** to **enforcing**.

      ```
      SELINUX=enforcing
      ```

   b. Restart the OS to apply the changes.

Appendix: Description of the .conf File

```
#rpm lists for os migration
[rpm_lists]
#origin system must need rpms
baserpms_list = "basesystem initscripts hce-logos plymouth grub2 grubby"  //The RPM packages required for the OS migration.

#old rpm and default conflict rpms  //The conflicting RPM packages that may exist in the source OS during the migration.
oldrpms_list =  centos-backgrounds centos-release-cr desktop-backgrounds-basic \
centos-release-advanced-virtualization centos-release-ansible26 centos-release-ansible-27 \
centos-release-ansible-28 centos-release-ansible-29 centos-release-azure \
centos-release-ceph-jewel centos-release-ceph-luminous centos-release-ceph-nautilus \
centos-release-ceph-octopus centos-release-configmanagement centos-release-dotnet centos-release-fdio \
centos-release-gluster40 centos-release-gluster41 centos-release-gluster5 \
centos-release-gluster6 centos-release-gluster7 centos-release-gluster8 \
centos-release-gluster-legacy centos-release-messaging centos-release-nfs-ganesha28 \
centos-release-nfs-ganesha30 centos-release-nfv-common \
centos-release-nfv-openvswitch centos-release-openshift-origin centos-release-openstack-queens \
centos-release-openstack-rocky centos-release-openstack-stein centos-release-openstack-train \
centos-release-openstack-ussuri centos-release-opstools centos-release-ovirt42 centos-release-ovirt43 \
centos-release-ovirt44 centos-release-paas-common centos-release-qemu-ev centos-release-qpid-proton \
centos-release-rabbitmq-38 centos-release-samba411 centos-release-samba412 \
centos-release-scl centos-release-scl-rh centos-release-storage-common \
centos-release-virt-common centos-release-xen centos-release-xen-410 \
centos-release-xen-412 centos-release-xen-46 centos-release-xen-48 centos-release-xen-common \
python3-syspurpose python-oauth sl-logos yum-rhn-plugin centos-indexhtml \
libreport-centos libreport-web libreport-plugin-mantisbt libreport-plugin-rhtsupport \
libreport hunspell-en-US hunspell-en policycoreutils-gui libcanberra-gtk2 cups \
NetworkManager-libreswan-gnome plymouth-graphics-libs avahi cups-lpd pinentry-qt \
librsvg2-devel libcanberra-gtk3 gnome-themes-standard wodim gsettings-desktop-schemas-devel \
avahi-ui-gtk3 freerdp-libs pulseaudio-utils gstreamer1-plugins-bad-free-gtk ghostscript-cups \
setools-console libxkbcommon-x11 cups plymouth-plugin-two-step pulseaudio-module-x11 ImageMagick-c++ \
cups-devel policycoreutils-sandbox PackageKit-gstreamer-plugin gtk3-immodule-xim avahi-glib avahi-
```



```
autoipd \
mesa-libGLES foomatic libcanberra-devel plymouth-plugin-label PackageKit-gtk3-module colord avahi-
gobject \
pinentry-qt4 avahi-ui-gtk3 plymouth-plugin-two-step ghostscript-cups ImageMagick-perl firewall-config \
plymouth-plugin-label redhat-redhat-lsb-corelsb vim-X11 dbus-x11 pulseaudio PackageKit-command-not-
found libproxy-mozjs \
pinentry-gtk nm-connection-editor gtk2-immodule-xim wireshark-gnome pulseaudio-module-bluetooth
pidgin-sipe freerdp kmod-kvdo  \
redhat-lsb-core

#The following list contains the same symbol as centos/redhat
dstrpms_list = "hce-release hce-repos"

[log_conf]
# migration tool log common dir
migrate_common_dir = "/var/log/migrate-tool/"    //The path for storing logs.

# migration tool classification log dir
migrate_classification_dir = %(migrate_common_dir)s/centos2hce1/

#iso as yum source link
[repo_info]
base_yum_url =https://repo.huaweicloud.com/hce/1.1/os/x86_64/  //The base yum URL used for checking
the network connection.


#iso as yum source
repostr_hce1_1 =     //The source path that provides the migration method.
   [base]
   name=hceversion
baseurl=https://repo.huaweicloud.com/hce/1.1/os/x86_64/   //The base yum URL used for obtaining the
RPM packages.
   gpgcheck=0
   enabled=1
   gpgkey=

   #released updates
   [updates]
   name=hce1_updates
   baseurl=
   gpgcheck=0
   enabled=0
   gpgkey=

   #additional packages that may be useful
   [extras]
   name=hce1_extras
   baseurl=
   gpgcheck=0
   enabled=0
   gpgkey=

   # plus packages provided by Huawei Linux dev team
   [plus]
   name=hce1_plus
   baseurl=
   gpgcheck=0
   enabled=0
   gpgkey=
```


5 Upgrading HCE and RPM Packages

5.1 Upgrade Overview

Updates and maintenance are provided for HCE and the RPM packages, including RPM packages deployed on HCE and those related to security updates for vulnerability fixing. To ensure system security, always install updates in a timely manner.

You can upgrade HCE using either dnf, yum, or OSMT:

- In Linux, you can use dnf or yum to upgrade or roll back RPM packages.
- OSMT is a software tool from Huawei Cloud that you can use to upgrade or roll back HCE and RPM packages. OSMT allows you to customize the upgrade scope and configure scheduled checks and delayed restarts.

Differences between the two methods are described in the following table.

Table 5-1 Differences between two methods

Item	dnf or yum	OSMT
RPM package upgrade	• Upgrading all RPM packages, including those related to security updates for vulnerability fixing • Upgrading only RPM packages related to security updates	• Upgrading all RPM packages, including those related to security updates for vulnerability fixing • Custom upgrades: – Upgrading only RPM packages that do not need an OS restart – Upgrading only RPM packages that need an OS restart – Upgrading the RPM packages defined in a custom blacklist or whitelist – Upgrading RPM packages related to security updates – Fixing vulnerabilities – Upgrading the RPM packages related to new functions – Updating new RPM packages • Automatically updating RPM package and delaying restarts until specified times
OS version upgrade	Not supported	Supported
Version	Upgrade of HCE 1.1	Upgrade of HCE 2.0
Rollback	Rollback to any historical update	Rollback only to the last update

5.2 Using dnf or yum for Upgrade and Rollback

This section describes how to update or roll back HCE 1.1. The methods of using dnf and yum are the same. In this section, dnf is used as an example.

NOTE

- HCE 2.0 and later support both yum and dnf.
- HCE 1.1 supports only yum.

Context

Dandified YUM (DNF) and Yellowdog Updater Modified (YUM) are RPM-based package management tools used to install, update, and remove software

packages. YUM is an earlier tool, while DNF is its successor. DNF aims to provide better performance and more functions, such as modules and parallel download. Both **yum** and **dnf** commands can be used by HCE to achieve better compatibility.

Upgrade Procedure

1. Check what RPM package updates are available.
   - Run **dnf list updates** to see the available updates.

     ```
     [root@localhost bin]# dnf list updates
     Last metadata expiration check: 6:49:11 ago on Tue 28 Jun 2022 01:55:35 PM CST.
     hce-config.x86_64                        3.0-66.hce2
     hce-latest-release.x86_64                  2.0-1656179342.2.0.2206.B032.hce2
     irqbalance.x86_64                        3:1.8.0-7.h9.hce2
     kernel.x86_64                           5.10.0-60.18.0.50.h316_1.hce2
     kernel-tools.x86_64                      5.10.0-60.18.0.50.h316_1.hce2
     kernel-tools-libs.x86_64                  5.10.0-60.18.0.50.h316_1.hce2
     kexec-tools.x86_64                       2.0.23-4.h8.hce2
     libcurl.x86_64                          7.79.1-2.h4.hce2
     libssh.x86_64                           0.9.6-2.h3.hce2
     libstdc++.x86_64                         10.3.1-10.h10.hce2
     libxml2.x86_64                           2.9.12-5.h5.hce2
     openssh.x86_64                           8.8p1-2.h12.hce2
     openssh-clients.x86_64                     8.8p1-2.h12.hce2
     openssh-server.x86_64                      8.8p1-2.h12.hce2
     Obsoleting Packages
     dnf-data.noarch                          4.10.0-3.h6.hce2
     dnf.noarch                              4.10.0-3.h5.hce2
     dnf-data.noarch                          4.10.0-3.h6.hce2
     dnf-data.noarch                          4.10.0-3.h5.hce2
     ```

   - Run **dnf list updates --security** to list only the RPM packages related to security updates.

     ```
     [root@localhost bin]# dnf list updates --security
     Last metadata expiration check: 0:00:03 ago on Fri 08 Jul 2022 04:45:56 PM CST.
     No security updates needed, but 2 updates available
     ```

2. Update the RPM packages.
   - Run **dnf update** to update all the RPM packages, including those related to security updates for vulnerability fixing. The components' target versions are returned to the **Version** column.

     ```
     [root@localhost bin]# dnf update
     Last metadata expiration check: 7:12:18 ago on Tue 28 Jun 2022 01:55:35 PM CST.
     Dependencies resolved.
     ================================================================================
     ====================
      Package                  Arch      Version                          Repo     Size
     ================================================================================
     ====================
     Installing:
      kernel                  x86_64    5.10.0-60.18.0.50.h316_1.hce2          hce2     47 M
     Upgrading:
      hce-config              x86_64    3.0-66.hce2                      hce2     13 k
      hce-latest-release        x86_64    2.0-1656179342.2.0.2206.B032.hce2       hce2    5.2 k
      kernel-tools            x86_64    5.10.0-60.18.0.50.h316_1.hce2          hce2    230 k
      kernel-tools-libs         x86_64    5.10.0-60.18.0.50.h316_1.hce2          hce2     62 k
      kexec-tools             x86_64    2.0.23-4.h8.hce2                 hce2    400 k
      libcurl                 x86_64    7.79.1-2.h4.hce2                 hce2    284 k
      libssh                  x86_64    0.9.6-2.h3.hce2                  hce2    194 k
      libstdc++                 x86_64    10.3.1-10.h10.hce2                hce2    535 k
      libxml2                 x86_64    2.9.12-5.h5.hce2                 hce2    659 k
      logrotate               x86_64    3.18.1-1.h2.hce2                 hce2     60 k
      mdadm                     x86_64    4.1-5.h2.hce2                    hce2    331 k
      nftables                x86_64    1:1.0.0-1.h3.hce2                hce2    303 k
      perl                    x86_64    4:5.34.0-3.h5.hce2               hce2    3.2 M
      perl-libs               x86_64    4:5.34.0-3.h5.hce2               hce2    1.8 M
     Installing dependencies:
     ```



```
 grub2-tools-efi             x86_64    1:2.06-3.h5.hce2                hce2    472 k

 Transaction Summary
================================================================================
====================
Install   2 Packages
Upgrade  72 Packages
Total download size: 105 M
Is this ok [y/N]:
```

- Run **dnf update --security** to update only the RPM packages related to security updates.

```
[root@localhost bin]# dnf update --security
Last metadata expiration check: 7:15:16 ago on Tue 28 Jun 2022 01:55:35 PM CST.
No security updates needed, but 73 updates available Dependencies resolved.
Nothing to do.
Complete!
```

3. After the update is successful, check that services are running properly.

Rollback Procedure

1. Run **dnf history** to query the IDs of historical operations.

   Figure 5-1 Querying the IDs of historical operations

```
[root@localhost ~]# dnf history
ID     | Command line                | Date and time    | Action(s)      | Altered
-------------------------------------------------------------------------------
     5 | upgrade chrony              | 2022-10-09 11:38 | Upgrade        |    1
     4 | history undo 3              | 2022-10-09 11:37 | Downgrade      |    1
     3 | upgrade chrony              | 2022-10-09 11:36 | Upgrade        |    1
     2 | install createrepo          | 2022-10-09 11:30 | Install        |    2
     1 |                             | 2022-10-09 11:15 | Install        |  421 EE
[root@localhost ~]#
```

2. Run **dnf history undo** <*ID*> to roll back to the desired historical operation.

5.3 Upgrade Using OSMT

5.3.1 Overview

OSMT is a tool provided by Huawei Cloud to upgrade or roll back HCE and RPM packages. It also allows you to customize the upgrade scope, configure scheduled checks, perform a single upgrade at the specified time period, and schedule restarts for RPM packages.

- To upgrade or roll back HCE, see **Version Upgrade and Rollback**.
- To upgrade or roll back only RPM packages, see **Updating RPM Packages**.

NOTE

OSMT can only upgrade or roll back HCE 2.0 or later. This tool periodically accesses the repository to obtain software update information, which will generate network traffic. You can run the **systemctl stop osmt-agent** command to stop this tool and run the **systemctl disable osmt-agent** command to disable automatic startup of this tool.

5.3.2 Constraints

- An upgrade or a rollback takes no more than 30 minutes, depending on the number and size of RPM packages to be updated and the download speed of

RPM packages from the repository. Reserve sufficient time based on your environment.

- OSMT can only be used to upgrade RPM packages in official repositories. Ensure that repositories are configured correctly. You must run **systemctl restart osmt-agent** to restart the osmt-agent service after modifying a repository.

  Create the **/etc/yum.repos.d/hce.repo** file and configure it as follows:

```
[base]
name=HCE $releasever base
baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

[updates]
name=HCE $releasever updates
baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2

[debuginfo]
name=HCE $releasever debuginfo
baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
enabled=0
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
```

- Modifying the OSMT configuration file using a method other than the **osmt config** command may lead to abnormal OSMT functions, so you are advised to run **osmt config** to modify the file.
- Upgrades must be performed as user **root**.
- Upgrading or rolling back the OS or RPM packages has the following requirements:
  - Memory: 512 MB
  - Root partition: 1.5 GB
  - Backup storage path (**store_path**): 8 GB
  - **/boot** partition in the OS: 100 MB

    NOTE

    - The required storage space varies depending on the upgrade scope and target version. During the upgrade, OSMT automatically estimates the space required for the upgrade. If the available space is insufficient, an error message is displayed.
    - The memory and storage space requirements are concluded from upgrade best practices. You are not advised to change them. If you change them to smaller values, the upgrade may fail due to insufficient memory or storage space.
- The upgrade and rollback impacts on the SELinux status are as follows:
  - An upgrade has no impact on the SELinux status. The SELinux status before and after the upgrade is the same.
  - If the SELinux status before a rollback is **enforcing**, after the rollback, its status automatically changes to **permissive**.
  - To enable SELinux, manually change the SELinux status to **enforcing** and restart the OS.

  - If the SELinux status before a rollback is **disabled**, the rollback has no impact on the SELinux status. The SELinux status before and after the rollback is the same.
  - To enable SELinux, set the SELinux status to **permissive**, create the **.autorelabel** file in the root directory, restart the OS, change the SELinux status to **enforcing**, and restart the OS.
- OSMT checks the OS health status before the upgrade. If the check fails, resolve the issue based on the information provided. You can also manually perform the check by referring to **OSMT Command Help Information**.
- The upgrade using OSMT depends on the DNF tool. To ensure the stability of the upgrade, OSMT will update the DNF tool and its dependent RPM packages to the latest version. For details about how to roll back the RPM packages, see **Rollback Procedure**.
- If the system configuration (you can run **sysctl -a** to query system configuration) is modified after the RPM packages are updated, the upgrade cannot be performed using OSMT. You can run **sysctl -p** to update the system configuration. You can run **sysctl -p <file>** to specify the configuration file that takes effect. The **sysctl --system** command can be run on the configuration files in all system directories. Before running this command, confirm the kernel configuration files in all system directories.
- The kernel or kernel hot patch cannot have more than five versions. If the kernel or kernel hot patch has more than five versions, the OSMT check will fail. If this happens, uninstall unnecessary versions and perform the check again.
- If chrony and NTP coexist and chrony is in the active state, the OSMT check will fail. If this happens, stop the chrony service or uninstall either chrony or the NTP service and perform the upgrade again.

5.3.3 Version Upgrade and Rollback

This section describes how to upgrade or roll back HCE.

During an OS upgrade or rollback, RPM packages will be updated to the versions of the target OS. The blacklist and whitelist configured in **osmt.conf** will not be applied.

Upgrading the OS Version

1. Confirm that the repository is configured correctly.

   Check whether the parameters in the **/etc/yum.repos.d/hce.repo** file are configured correctly. The correct configuration is as follows:

   ```
   [base]
   name=HCE $releasever base
   baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
   enabled=1
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

   [updates]
   name=HCE $releasever updates
   baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
   enabled=1
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2
   ```



```
[debuginfo]
name=HCE $releasever debuginfo
baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
enabled=0
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
```

NOTE

- Incorrect configuration may result in OSMT upgrade failures or unexpected upgrades.

2. Update the OSMT version.

   There is a mapping between OSMT versions and HCE versions. HCE installs by default the OSMT of the current OS. When upgrading HCE, you also need to update the OSMT to the matched version.

   Run **dnf update osmt -y --releasever** [*Target OS version*] to update the OSMT version.

   NOTE

   You can also run **dnf install osmt -y --releasever** [*Target OS version*] to install OSMT if it is deleted by mistake. For example, if the OS version is HCE 2.0, you can run **dnf install osmt -y --releasever 2.0** to install OSMT 2.0.

3. Upgrade the HCE version.

   osmt update --releasever [*Target version*] **--reboot_config** [*Restart configuration*]

   Choose an appropriate upgrade method. For more upgrade options, see **osmt update -h**.

   – Upgrade HCE 2.0 to a new version, for example, HCE 3.0.

     osmt update --releasever 3.0

     The upgrade is only applied after a reboot.

   – Upgrade HCE 2.0 to a new version, for example, HCE 3.0. Then, restart the OS.

     osmt update --releasever 3.0 --reboot_config always

   – Upgrade HCE 2.0 to a new version, for example, HCE 3.0, and specify the restart time, for example, 2022-12-30 23:00:00.

     osmt update --releasever 3.0 --reboot_config "2022-12-30 23:00:00"

4. Check whether the upgrade was successful.

   Run **cat /etc/hce-latest** and view the **hceversion** field. If the **--releasever** value is the version you specified, the upgrade was successful.

5. (Optional) Delete backup files.

   After verifying the OS functions, run **osmt remove** to delete the backup files.

   NOTE

   The operation of deleting backup files cannot be undone. Ensure that no exception occurs after the upgrade before you run **osmt remove**.

Rolling Back the OS Version

1. Choose an appropriate rollback method.
   – To roll back and not restart the OS, run the following command:

osmt rollback

- To roll back and restart the OS immediately, run the following command: (Then skip step 2.)

  osmt rollback --reboot_config always

2. Run **reboot** to restart the OS.

   The rollback is only applied after a restart.

3. Check whether the rollback was successful.

   Run **cat /etc/hce-latest** and view the **hceversion** field. If the **hceversion** value is the source version, the rollback was successful.

5.3.4 Updating RPM Packages

5.3.4.1 Preparations

RPM packages can be updated manually (using **osmt update**) or automatically (using the background osmt-agent service). You need to perform the following operations for both manual and automatic updates.

1. Confirm that the repository is configured correctly.

   Check whether the parameters in the **/etc/yum.repos.d/hce.repo** file are configured correctly. The correct configuration is as follows:

```
[base]
name=HCE $releasever base
baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

[updates]
name=HCE $releasever updates
baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2

[debuginfo]
name=HCE $releasever debuginfo
baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
enabled=0
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
```

   NOTE

   - Incorrect configuration may result in OSMT upgrade failures or unexpected upgrades.

2. Run **dnf update osmt -y** to update OSMT.
3. Configure the **/etc/osmt/osmt.conf** file.

   OSMT updates RPM packages based on the **osmt.conf** file. Configure the file as required.

```
[auto]
#if auto_upgrade is True, the osmt-agent will auto upgrade rpms use osmt.conf and reboot between time interval we specified
#the value of cycle_time means the  osmt-agent will check upgrade every cycle_time seconds, default 86400s(1 day)
#When a configuration item has a line break, you need to leave a space or tab at the beginning of
```



```
the line
auto_upgrade = False
cycle_time = 3600
minimal_interval = 3600
auto_upgrade_window = "22:00-05:00"
auto_upgrade_interval = 1

[Package]
# There are three rules of filters, all enabled by default. severity will be effect only when the types contain security, it is the subtype of security.
# The following are the three rules:
#    1. white list has the highest priority, if whitelist is configured then ignore other rules and filter out the whitelist packages from the full list of packages to be upgrade
#    2. Filter the update range by types, when the types contain security, further filter the severity of security updates severity, only upgrade the severity level of security.
#    3. Filter blacklist to remove packages in blacklist from types filter results, and packages which depend on packages in blacklist will also be removed.
# filters must contain at least one types rule, if the types rule is empty, the -a option will not upgrade any packages (by default all 3 filters are enabled).

filters = "types, blacklist"
whitelist = ""

# types include: security, bugfix, enhancement, newpackage, unknown/other
# if types is empty, no package will be upgrade
types = "security"
# severity is the subtype of security, include: low, moderate, important, critical
severity = "important, critical"
blacklist = ""
# The rpm package that requires a system reboot to take effect after the upgrade
need_reboot_rpms = "kernel,kernel-debug,kernel-debuginfo,kernel-debuginfo-common,kernel-devel,kernel-headers,kernel-ori,kernel-tools,kernel-tools-libs,glibc,glibc-utils,glibc-static,glibc-headers,glibc-devel,glibc-common,dbus,dbus-python,dbus-libs,dbus-glib-devel,dbus-glib,dbus-devel,systemd,systemd-devel,systemd-libs,systemd-python,systemd-sysv,grub2,grub2-efi,grub2-tools,openssl,openssl-devel,openssl-libs,gnutls,gnutls-dane,gnutls-devel,gnutls-utils,linux-firmware,openssh,openssh-server,openssh-clients,openssh-keycat,openssh-askpass,python-libs,python,grub2-pc,grub2-common,grub2-tools-minimal,grub2-pc-modules,grub2-tools-extra,grub2-efi-x64,grub2-efi-x64-cdboot,kernel-cross-headers,kernel-source,glibc-all-langpacks,dbus-common,dbus-daemon,dbus-tools,systemd-container,systemd-pam,systemd-udev,grub2-efi-aa64,grub2-efi-aa64-cdboot,grub2-efi-aa64-modules,openssl-perl,openssl-pkcs,kernel-tools-libs-devel,glibc-debugutils,glibc-locale-source,systemd-help,grub2-efi-ia32-modules,grub2-efi-x64-modules,grub2-tools-efi,grub2-help,openssl-pkcs11,grub2-efi-ia32-cdboot,osmt"
preinstalled_only = False
# Due to security requirements, the following packages need to be uninstalled during the upgrade
need_uninstall_rpm_list = "elfutils-extra,gcc,make,tcpdump,binutils-extra,strace,gdb,gdb-headless,cpp,rpm-build,cups,ypserv,telnet,ypbind,libtool,appict,kmem_analyzer_tools,mcpp,flex,cmake,llvm,rpcgen,wireshark,netcat,nmap,ethereal"

[backup]
store_path = /var/log
backup_dir = /etc,/usr,/boot,/var,/run
exclude_dir =
recover_service =

#the minimum resources required(MB)
[resource_needed]
#min_req_boot_space = 100
#min_req_backup_space = 8192
#min_req_root_space = 1536
#min_req_memory = 512

[cmdline]
cmdline_value =
skip_swap = True

[conflict]
#conflict_rpm = test1,test2
# These rpms conflict with the upgrade and must be removed if installed.
```



```
conflict_rpms_list = "esc,initial-setup,python3-crypto,setroubleshoot,setroubleshoot-
legacy,setroubleshoot-server,setroubleshoot-plugins,openresty-openssl111,openresty-openssl111-
asan,openresty-openssl111-asan-devel,openresty-openssl111-debug,openresty-openssl111-debug-
devel,openresty-openssl111-devel,openresty-zlib,openresty-zlib-asan,openresty-zlib-asan-
devel,openresty-zlib-devel,dleyna-connector-dbus,dleyna-connector-dbus-devel,dleyna-core,dleyna-
core-devel,dleyna-server,tracker,tracker-devel,tracker-miners,kabi-dw,sblim-
sfcb,apull,elara,secpaver,ksh,python3-sssd,python3-editor,python3-Flask-SQLAlchemy,python3-bind"

[strategy]
timeout_action = "stop"
timeout_action_before = 0

[check]
daemon_whitelist = "sysstat-collect.service, sysstat-summary.service, man-db-cache-update.service,
systemd-tmpfiles-clean.service"
check_systemd_running_jobs = True
# the timeout of query systemd services
query_timeout = 30
check_rpm_packages = True
check_file_attr = True

[chroot_config]
chroot_switch = False
chroot_path = "/root/sut_chroot"
rpm_tar_name = "hce-upgrade_pack"
sut_config_file = "/etc/sut/sut.conf"
web_link_tar =
```


Table 5-2 Major configuration items in **osmt.conf**

Configuration Item	Description
[auto]	• **auto_upgrade**: specifies the RPM package update method. The default value is **False**. – **True**: RPM packages can be updated either manually or automatically. – **False**: RPM packages can only be updated manually. • If **auto_upgrade** is set to **True**, the following parameters are available: – **cycle_time**: defines the interval of checking for available updates, in seconds. The default value is **3600**. – **minimal_interval**: defines the minimum interval (in seconds) between the start time and end time in **osmt update -b**. The default value is **3600**. – **auto_upgrade_window**: defines the start time and end time of automatic updates using the osmt-agent service. The value is in the format of "HH:MM-HH:MM". If the end time is smaller than the start time, the update period covers two dates. For example, **22:00-05:00** indicates an update period from 22:00 on the current day to 05:00 on the next day. – **auto_upgrade_interval**: defines the minimum interval between two automatic updates, in days. • If **auto_upgrade** is set to **False**, only the following parameters are available, and any other [auto] parameter configured will not take effect. – **cycle_time**: defines the interval of checking for available updates, in seconds. The default value is **3600**. – **minimal_interval**: defines the minimum interval (in seconds) between the start time and end time in **osmt update -b**. The default value is **3600**. • **motd_setup**: specifies whether to enable the login prompt. The default value is **True**. – **True**: Enable the login prompt. – **False**: Disable the login prompt. After the setting, the login prompt is deleted immediately and will not be generated again. If the option is enabled again, you need to run the **osmt update -s** command or any upgrade command to trigger the generation again.

Configuration Item	Description
[Package]	• **filters**: specifies the update scope. The value can be **types**, **blacklist**, or **whitelist**. For example, specifying **blacklist** will not update the RPM packages in the blacklist. – **types**: the type of RPM packages to be updated. – **blacklist**: the RPM package blacklist. Packages added to the blacklist will not be updated. If an RPM package depends on a package in the blacklist, this RPM package will also not be updated. – **whitelist**: the RPM package whitelist. If the whitelist and blacklist are not configured, all RPM packages will be updated. The whitelist has a higher priority than the blacklist, meaning that if an RPM package is added to both lists, it will be updated. **NOTE** The blacklist and whitelist specified in the command are applied, and those defined in the configuration file will not. • **need_reboot_rpms**: lists the RPM packages that need an OS restart. Automatic updates using osmt-agent will not update packages in **need_reboot_rpms**. To update these packages during automatic updates, you must run **osmt update --auto --reboot_config always** or **osmt update --auto --reboot_config** *"Restart time"*. • **preinstalled_only**: If this parameter is set to **True**, only the RPM packages in **/etc/osmt/preinstalled.list** are required to be upgraded.

Configuration Item	Description
[backup]	• **store_path**: the directory under which the backup directory is created. During the upgrade, OSMT creates the **.osbak** directory under **store_path**. If **.osbak** already exists, run the **osmt remove** command to delete it first. • **backup_dir**: the directories that need to be backed up. Directories **/etc**, **/usr**, **/boot**, **/var**, and **/run** are backed up by default during an update and cannot be removed. The values must be different from those of **exclude_dir**. Otherwise, the upgrade tool will be terminated due to a conflict when processing these directories. • **exclude_dir**: directories excluded from the backup. By default, this parameter is omitted. You are advised to set this parameter to service data directories. The values must be different from those of **exclude_dir**. Otherwise, the upgrade tool will be terminated due to a conflict when processing these directories. • **recover_service**: OSMT checks whether the status of each service in the list is the same before and after the upgrade. If the status of a service is changed, OSMT will restore the service status. **NOTE** The path in **[backup]** must be an absolute path.
[cmdline]	**cmdline_value**: determines the startup items after an upgrade. Configure correct startup items to ensure that the OS can be started properly. By default, the default startup items of HCE are used.
[conflict]	**conflict_rpm**: specifies the RPM packages that will be deleted if there is a conflict during the upgrade.

Configuration Item	Description
[check]	• **check_systemd_running_jobs**: specifies whether to check for services that are being started or stopped in the system before the upgrade. – **True** (default): Any services that are being started or stopped in the system will be checked before the upgrade. – **False**: Any services that are being started or stopped in the system will not be checked before the upgrade. • **check_rpm_packages**: specifies whether to check the status of RPM packages in the system before the upgrade, including missing package dependencies and whether duplicate packages. – **True** (default): The status of the RPM packages will be checked before the upgrade. – **False**: The status of the RPM packages will not be checked before the upgrade. • **check_file_attr**: specifies whether to check if system files are read-only or not before an upgrade. The default value is **True**. – **True**: Check if system files are read-only or not before an upgrade. – **False**: Do not check if system files are read-only or not before an upgrade.
[chroot_config]	• **chroot_switch**: specifies whether to enable upgrade in turbo mode. The default value is **False**. – **True**: Upgrade in turbo mode is enabled. – **False**: Upgrade in turbo mode is disabled. • **chroot_path**: specifies the directory of the chroot environment for upgrade in turbo mode. • **rpm_tar_name**: specifies the name of the pre-built tar package for upgrade in turbo mode. • **sut_config_file**: specifies the path of the SUT configuration file. The default value is **/etc/sut/sut.conf**. • **web_link_tar**: specifies the URL for downloading the pre-built tar package for upgrade in turbo mode. **NOTE** All configurations in **[chroot_config]** only apply to HCE upgrades on the management plane. The turbo mode means that a pre-built tar package of the target version is used for upgrades using **chroot** in a specified directory.

NOTE

You are advised not to modify other configuration items. For details, see **Description of the /etc/osmt/osmt.conf File**.

5.3.4.2 Manual Update Using osmt update

You can manually update RPM packages in the following ways:

- Update the RPM packages using the **filters** field in the configuration file.

  osmt update --auto --reboot_config [*Restart configuration*]

  Table 5-3 Restart parameter description

Value	Description
never	Does not restart the OS after the update. If the **reboot** parameter is not configured or its value is set to **never**, the OS will also not restart after the update. This way, RPM packages in the **need_reboot_rpms** list will not be updated. To update them, run the following commands to set **filters** to **whitelist** and add the packages to the whitelist: (The update is only applied after a restart.) **osmt config -k filters -v "whitelist"** **osmt config -k whitelist -v** "*rpm1, rpm2, rpm3*"
always	Updates the RPM packages (including the packages in **need_reboot_rpms**) and restart the OS immediately after the update.
<specific time>	Updates the RPM packages (including the packages in **need_reboot_rpms**) and restart the OS at the specified time, for example **"2020-02-02 2:02:02"**.

- Update the RPM packages using the whitelist and blacklist.

  osmt update --pkgs [*rpm1 rpm2 rpm3 ...*] **--exclude_pkgs** [*rpm4 rpm5 rpm6 ...*] --**reboot_config** [*Restart configuration*]

  – **--pkgs**: (optional) specifies the whitelisted packages to be updated. Multiple packages are separated by spaces.

    For example, run the following command to update the **hce-logos**, **hce-lsb**, and **tomcat** whitelisted packages:

    osmt update --pkgs hce-logos hce-lsb tomcat

  – **--exclude_pkgs**: (optional) specifies the blacklisted packages that will not be updated. Multiple packages are separated by spaces.

    For example, run the following command to not update the **ongres-scram** and **llvm-static** blacklisted packages:

    osmt update --exclude_pkgs ongres-scram llvm-static

  – **--reboot_config** [*Restart configuration*]: (optional) configures the restart method. The value can be **always**, **never**, or a specific restart time.

- **always**: restarts the OS after the update if some package updates are only applied after a restart. If there are no such packages, the OS will not restart.
- **never**: does not restart the OS after the update.
- <*specific time*>: specifies a specific restart time. If some package updates are only applied after a restart, the OS will restart at the specified time after the update. The restart time is in the format of "2020-02-02 2:02:02". If there are no such packages, the OS will not restart.

NOTE

- If you update packages using the blacklist or whitelist, specify at least one of **--pkgs** and **--exclude_pkgs**.
- The blacklist and whitelist specified in the command are applied, and those defined in the configuration file will not.

5.3.4.3 Automatic Update Using osmt-agent

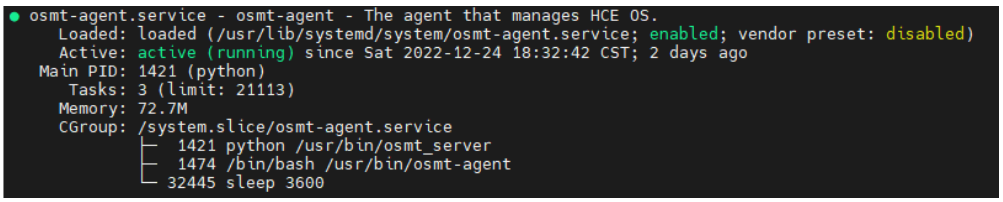
The osmt-agent service periodically checks whether there are available RPM package updates and updates them automatically. You can configure how often to check updates and when to perform the update.

1. Run the following command to ensure that the value of **auto_upgrade** in the **osmt.conf** file is **True**:

   osmt config -k auto_upgrade -v True

2. Run **systemctl status osmt-agent.service** to check whether the osmt-agent service is started.
   - If the **Active** value is **active (running)**, osmt-agent is started.
   - Otherwise, run **systemctl start osmt-agent.service** to start osmt-agent.

   Figure 5-2 Checking whether the osmt-agent service is started

3. Run the desired command to configure when or how often to perform the updates:
   - To configure a time window for automatic updates:

     osmt config -k auto_upgrade_window -v "*auto_upgrade_window*"

     auto_upgrade_window: defines the start time and end time of automatic updates using the osmt-agent service. The value is in the format of "HH:MM-HH:MM".

     If the end time is smaller than the start time, the update period covers two dates. For example, **22:00-05:00** indicates an update period from 22:00 on the current day to 05:00 on the next day.

For example, run the following command to configure a time window starting from 23:00 on the current day to 01:00 on the next day:

osmt config -k auto_upgrade_window -v "23:00-01:00"

– To configure the interval between two automatic updates:

osmt config -k auto_upgrade_interval -v *auto_upgrade_interval*

auto_upgrade_interval: defines the minimum interval between two automatic updates, in days.

For example, run the following command to configure automatic updates every other day:

osmt config -k auto_upgrade_interval -v 1

5.3.5 Follow-up Operations

1. After the update is successful, check that services are running properly. Then run **osmt remove** to delete the backup files when appropriate. Once deleted, the update cannot be rolled back.
2. Based on the requirements of security regulations, the chronyd service will be disabled after Huawei Cloud EulerOS is upgraded from 2.0.2206 to a new version. If required, run **systemctl enable chronyd** to enable the service and run **systemctl start chronyd** to start the service.

5.3.6 Rolling Back RPM Packages

Run **osmt rollback --reboot_config always** to roll back RPM packages. They can only be rolled backup to the last update.

In the command, **--reboot_config always** is optional. You must use it if there are RPM packages in **need_reboot_rpms** updated in the last update.

If **--reboot_config always** is not specified, you need to manually restart the OS so that the rollback of packages in **need_reboot_rpms** can be applied.

NOTE

Use the latest OSMT. You are advised not to roll back OSMT using OSMT.

5.4 Appendixes

5.4.1 OSMT Command Help Information

- Run **osmt -h** to display the OSMT help information.

```
[root@localhost SOURCES]# osmt -h
usage: osmt [-h] {update,rollback,remove,config,job} ...

positional arguments:
  {update,rollback,remove,config,job}
    update              update packages
    rollback            rollback last upgrade
    remove              remove backup files in store path
    config              modify config file by command line
    job                 handle upgrade task.

options:
  -h, --help            show this help message and exit
```


Table 5-4 OSMT parameters

Parameter	Description
update	Upgrades the OS or update the RPM packages.
rollback	Rolls back the OS or RPM packages.
remove	Deletes the backup files from the storage path.
config	Queries or modifies the configuration file.
job	Queries or manages the upgrade tasks.
-h, --help	(Optional) Provides help information of the **osmt** command.

- Run **osmt update -h** to display the help information about the OS or RPM package updates.

```
[root@localhost SOURCES]# osmt update -h
usage: osmt update [-h] [--nosignature] [-s] [--all] [--security] [--version] [-a] [-p  PKGS [PKGS ...]] [-
e  EXCLUDE_PKGS [EXCLUDE_PKGS ...]] [-v  RELEASEVER] [-r  REBOOT_CONFIG]
                   [-b BETWEEN] [-j] [-c]
optional arguments:
  -h, --help            show this help message and exit
  --nosignature         ignore the signature of packages
  -s , --show           show updateinfo
  --all                 show all pkgs which can update, 'osmt update --show --all'
  --security            show security pkgs which can update
  --version             show all version can update to
  -a , --auto           auto update use config file
  -p  PKGS [PKGS ...], --pkgs PKGS [PKGS ...]
                        specify the packages to upgrade
  -e  EXCLUDE_PKGS [EXCLUDE_PKGS ...], --exclude_pkgs EXCLUDE_PKGS [EXCLUDE_PKGS ...]
                        specify the packages not to upgrade
  -v  RELEASEVER, --releasever RELEASEVER
                        specify the release version to upgrade
  -r  REBOOT_CONFIG, --reboot_config REBOOT_CONFIG
                        you can choose between always, never or a specific time. 'always': reboot os after
update ends if need. 'never': never reboot os automatically. '<specific time>':
                        reboot at specified time, format like "2020-02-02 2:02:02".
  -b BETWEEN, --between BETWEEN
                        start upgrade time and end upgrade time, format like: '2020-02-02
2:02:02','2020-02-02 4:02:02'
  -j , --job            run upgrade in background.
  -c, --check           check upgrade task.
  -V, --verbose         show more log to screen
  -o, --preinstalled-only
                        upgrade preinstalled packages only
  -t, --retry           retry previous upgrade action
  --nocheck             do not check before upgrade
  -f, --fix             auto fix some system problems checked out
```

Table 5-5 osmt update

Parameter	Description
-h, --help	Provides help information about the **osmt update** command.
--nosignature	Specifies not to filter the RPM packages to be updated by package signature.

Parameter	Description
-s,--show	Displays available upgrade or update information. • **--all**: displays all RPM packages to be updated. • **--security**: displays the security packages to be updated. • **--version**: displays the version to be upgraded to.
-a,--auto	**Specifies the RPM package update method.** This parameter is mutually exclusive with **-v**, **-p**, and **-e**.
-p,--pkgs	Specifies the whitelisted RPM packages to be updated. This parameter is mutually exclusive with **-v** and **-a**.
-e,--exclude_pkgs	Specifies the blacklisted RPM packages that will not be updated. This parameter is mutually exclusive with **-v** and **-a**.
-v , --releasever	Specifies the target HCE version.
-r, --reboot_config	Specifies the restart configuration. • **always**: restarts the OS after the update if some package updates are only applied after a restart. If there are no such packages, the OS will not restart. • **never**: does not restart the OS after the update. • <*specific time*>: specifies a specific restart time. If some package updates are only applied after a restart, the OS will restart at the specified time after the update. The restart time is in the format of "2020-02-02 2:02:02". If there are no such packages, the OS will not restart.
-b, --between	Specifies the start time and end time of automatic updates using osmt-agent. The value is in the format of "HH:MM-HH:MM". If the end time is smaller than the start time, the update period covers two dates. For example, **22:00-05:00** indicates an update period from 22:00 on the current day to 05:00 on the next day.
-j, --job	Performs the upgrade using background processes.
-c, --check	Checks the system status before the upgrade. This parameter is optional. The system also performs the check before executing the upgrade. You are advised to add **-c** to the update command to perform the check. For example, before running **osmt update -a** to perform the upgrade, run **osmt update -a -c** to check the system status.
-V, --verbose	(Optional) Displays detailed upgrade information.

Parameter	Description
-o, --preinstalled-only	(Optional) Only the RPM package list in **/etc/osmt/preinstalled.list** is upgraded. This parameter is valid only for version upgrade.
-t, --retry	(Optional) Retries the upgrade.
--nocheck	(Optional) No checks will be performed before the upgrade. The upgrade starts directly.
-f, --fix	(Optional) Some environment problems are automatically resolved during the version upgrade.

- Run **osmt rollback -h** to display the help information about the OS or RPM package rollbacks.

```
usage: osmt rollback [-h] [-r  {never,always}] [-V] [-t] [--nocheck]

optional arguments:
  -h, --help            show this help message and exit
  -r  {never,always}, --reboot_config {never,always}
                        whether to reboot after rollback
  -V, --verbose         show more log to screen
  -t, --retry           retry previous upgrade action
  --nocheck             do not check before rollback
```

Table 5-6 osmt rollback

Parameter	Description
-h, --help	Provides help information about the **osmt rollback** command.
-r, --reboot_config	Specifies the restart configuration.
-V, --verbose	(Optional) Displays detailed process logs.
-t, --retry	(Optional) Retries the rollback.
--nocheck	(Optional) No checks will be performed before the rollback. The rollback starts directly.

- Run **osmt config -h** to display the help information about configuration item modification or query.

```
usage: osmt config [-h] [-k KEY] [-v VALUE] [-V]

optional arguments:
  -h, --help            show this help message and exit
  -k KEY, --key KEY     key of config item
  -v VALUE, --value VALUE
                        value of config item
  -V, --verbose         show more log to screen
```


Table 5-7 osmt config

Parameter	Description
-h, --help	Provides help information about the **osmt config** command.
-k, --key	Specifies the keys to be queried or modified.
-v, --value	Specifies the values of the keys to be modified.
-V, --verbose	(Optional) Displays detailed process logs.

NOTE

You are advised to run **osmt config** to modify the configuration file. Any modification made to the file using a method other than **osmt config** may lead to abnormal OSMT functions.

- Run **osmt job -h** to display the help information about task management.

```
usage: osmt job [-h] [-s] [-c] [-d DELAY] [-y]

optional arguments:
  -h, --help            show this help message and exit
  -s, --show            show task info.
  -c, --cancel          cancel current task.
  -d DELAY, --delay DELAY
                        delay task
  -y, --yes             never ask for yes.
  -V, --verbose         show more log to screen
```

Table 5-8 osmt job

Parameter	Description
-h, --help	Provides help information about the **osmt job** command.
-s, --show	Displays background task information.
-d, --delay	Allows for postponing the pending restart task. For example, to postpone the task for 1 hour and 50 minutes, set the value to **"1:50:00"**.
-c, --cancel	Cancels the current job.
-y, --yes	Considers by default that the user agrees the operation.
-V, --verbose	(Optional) Displays detailed process logs.

5.4.2 Description of the /etc/osmt/osmt.conf File

This section describes the OSMT configuration items that you are advised not to modify in the **osmt.conf** file.

```
[auto]
# if auto_upgrade is True, the osmt-agent will auto upgrade rpms use osmt.conf and reboot between time
```

```
interval we specified
# the value of cycle_time means the  osmt-agent will check upgrade every cycle_time seconds, default
86400s(1 day)
# When a configuration item has a line break, you need to leave a space or tab at the beginning of the line
auto_upgrade = False
cycle_time = 3600
minimal_interval = 3600
auto_upgrade_window = "22:00-05:00"
auto_upgrade_interval = 1
[Package]
# There are three rules of filters, all enabled by default. Severity will be effect only when the types contain
security, it is the subtype of security.
# The following are the three rules:
#    1. whitelist has the highest priority, if whitelist is configured then ignore other rules and filter out the
whitelist packages from the full list of packages to be upgrade
#    2. Filter the update range by types, when the types contain security, further filter the severity of
security updates severity, only upgrade the severity level of security.
#    3. Filter blacklist to remove packages in blacklist from types filter results, and packages which depend
on packages in blacklist will also be removed.
# filters must contain at least one types rule, if the types rule is empty, the -a option will not upgrade any
packages (by default all 3 filters are enabled).

filters = "types, blacklist"
whitelist = ""

# types include: security, bugfix, enhancement, newpackage, unknown
# if types is empty, no package will be upgrade
# types = security, bugfix, enhancement, newpackage, unknown
types = "security"
# severity is the subtype of security, include: low, moderate, important, critical
severity = "important, critical"
blacklist = "mysql"
# RPM packages that only take effect after an OS restart
need_reboot_rpms = kernel,kernel-debug,glibc,glibc-utils,dbus,dbus-python...
preinstalled_only = False
[backup]
store_path = /var/log
backup_dir = /etc,/usr,/boot,/var,/run
exclude_dir =
recover_service =
[resource_needed]
#the minimum resources required(MB)
#min_req_boot_space = 100
#min_req_backup_space = 8192
#min_req_root_space = 1536
#min_req_memory = 512
[cmdline]
cmdline_value = crashkernel=512M resume=/dev/mapper/hce-swap rd.lvm.lv=hce/root rd.lvm.lv=hce/swap
crash_kexec_post_notifiers panic=3 nmi_watchdog=1 rd.shell=0
[conflict]
#conflict_rpm = test1,test2

[strategy]
timeout_action = "stop"
timeout_action_before = 0

[check]
daemon_whitelist=sysstat-collect.service, sysstat-summary.service, systemd-tmpfiles-clean.service
# the timeout of query systemd services
check_systemd_running_jobs = True
query_timeout = 30
check_rpm_packages = True
check_file_attr = True

[chroot_config]
chroot_switch = False
chroot_path = "/root/sut_chroot"
rpm_tar_name = "hce-upgrade_pack"
```



```
sut_config_file = "/etc/sut/sut.conf"
web_link_tar =
```

Table 5-9 Configuration items that should not be modified in **osmt.conf**

Configuration Item	Description
types	The parameter that defines the RPM package update scope, including five configuration items **security**, **bugfix**, **enhancement**, **newpackage**, and **unknown**. You are advised not to modify it unless in some special cases.
severity	The system upgrades security updates by default. You are advised not to modify it unless in some special cases.
[resource_needed]	The minimum resource required by the system to perform the update or update check. You are advised not to modify it unless in some special cases.
[chroot_config]	Whether to perform an upgrade in turbo mode. This parameter is used only for HCE upgrades on the management plane and not for the tenant plane. Retain the default value.

5.4.3 FAQ

1. When yum, dnf, or osmt is used for installation or upgrade, an error message " **package** *<a>* **conflicts with** *<b>* **provided by** *<c>*" is displayed.

   Figure 5-3 Error message

   ```
   Error:
    Problem: problem with installed package mysql-8.0.37-1.hce2.x86_64
     - package mysql-8.0.37-1.hce2.x86_64 conflicts with mariadb provided by mariadb-4:10.5.22-1.hce2.x86_64
     - package mysql-8.0.28-1.oe2203.x86_64 conflicts with mariadb provided by mariadb-4:10.5.22-1.hce2.x86_64
     - cannot install the best candidate for the job
   ```

   Cause: Two software packages with the same function are installed in the system. The two software packages conflict with each other. For example, the MySQL and MariaDB software packages conflict with each other.

   Solution: Run **rpm -e/ yum remove/ dnf remove** to delete one of the software packages and retain only the required one.

2. When yum, dnf, or osmt is used for installation or upgrade, an error message "**file** *<x>* **from install of** *<y>* **conflicts with file from** *<z>*" is displayed.

   Figure 5-4 Error message

   ```
   Error: Transaction test error:
     file /etc/my.cnf from install of mariadb-config-4:10.5.22-1.hce2.x86_64 conflicts with file from package mysql-config-8.0.37-1.hce2.x86_64
   ```

   Cause: Two software packages contain files with the same path. For example, both the **mysql-config** and **mariadb-config** packages contain the **/etc/my.cnf** file.

   Solution: Run **rpm -e/ yum remove/ dnf remove** to delete one of the software packages and retain only the required one.

6 Security Updates for HCE

6.1 Security Updates Overview

This section describes how to query and install HCE security updates using yum or dnf.

The support for yum and dnf depends on the OS version. This section uses yum as an example.

NOTE

As a substitute for yum, dnf delivers better performance. The methods for using dnf and yum are the same.

- HCE 2.0 and later support both yum and dnf.
- Versions earlier than HCE 2.0 support only yum.

6.2 About CVE

Common Vulnerabilities and Exposures (CVE) is a list of publicly disclosed vulnerabilities, each with a unique CVE serial number. To ensure HCE security, Huawei Cloud closely follows industry vulnerability warnings and fixes software vulnerabilities in a timely manner. You can view the security updates at HCE security advisories:

- **Huawei Cloud EulerOS 1.1 Security Advisories**
- **Huawei Cloud EulerOS 2.0 Security Advisories**

In accordance with the Common Vulnerability Scoring System (CVSS), HCE security updates are classified into the following levels:

- Critical (high risk, mandatory)
- Important (medium high risk, strongly recommended)
- Moderate (medium risk, recommended)
- Low (low risk, optional)

This section uses HCE 2.0 as an example. Before you install HCEOS security updates, deploy a remote repository, add security advisories, and configure clients to access the repository. For details, see **Configuring an HCE Repository**.

NOTE

The command outputs vary depending on the HCE version.

6.3 Yum Command Parameters

Command format: yum <command> [option]

Table 6-1 Major <command> parameters

Parameter	Description
help	Displays help information.
updateinfo	Displays summary of package updates.
upgrade	Installs package updates.
check-update	Checks for available package updates.

Table 6-2 Major [option] parameters

Parameter	Description
-h, --help, --help-cmd	Displays command help information.
--security	Displays available security updates.
--advisory ADVISORY	Specifies specific advisories. This parameter can be used together with **updateinfo**, **upgrade**, and **check -update**. Multiple packages are separated by commas (,).
--cve CVES	Specifies specific CVEs. This parameter can be used together with **updateinfo**, **upgrade**, and **check -update**. Multiple packages are separated by commas (,).
--sec-severity {Critical,Important,Moderate,Low}	Specifies specific security levels. This parameter can be used together with **updateinfo**, **upgrade**, and **check -update**. Values in the brackets can be any combination of security update levels.

NOTE

Use the **yum --help** command to obtain more information.

6.4 Querying Security Updates

Command format: yum updateinfo <command> [option]

- Run the **yum updateinfo** command to query all available security updates.

```
[root@localhost ~]# yum updateinfo
Last metadata expiration check: 0:03:05 ago on Thu 08 Sep 2022 05:30:23 PM CST.
Updates Information Summary: available
12 Security notice(s)
4 Critical Security notice(s)
6 Important Security notice(s)
2 Moderate Security notice(s)
```

- Major <command> parameters include:
  - **list**: lists the available security updates.

```
[root@localhost ~]# yum updateinfo list
Last metadata expiration check: 0:03:32 ago on Thu 08 Sep 2022 05:30:23 PM CST.
HCE2-SA-2022-0006 Critical/Sec.  curl-7.79.1-2.h6.hce2.x86_64
HCE2-SA-2022-0011 Moderate/Sec.  gnupg2-2.2.32-1.h6.hce2.x86_64
HCE2-SA-2022-0002 Important/Sec. kernel-5.10.0-60.18.0.50.h425_2.hce2.x86_64
```

  - **info** <SA ID>: queries the security updates of a specific advisory.

```
[root@localhost ~]# yum updateinfo info HCE2-SA-2024-0262
Last metadata expiration check: 0:01:07 ago on Wed 26 Mar 2025 11:08:19 AM CST.
================================================================================
  An update for wget is now available for HCE 2.0
================================================================================
  Update ID: HCE2-SA-2024-0262
       Type: security
    Updated: 2024-09-23 18:09:48
       CVEs: CVE-2024-38428
Description: Security Fix(es):
           :
           : url.c in GNU Wget through 1.24.5 mishandles semicolons in the userinfo subcomponent
of a URI, and thus there may be insecure behavior in which data that was supposed to be in the
userinfo subcomponent is misinterpreted to be part of the host subcomponent.
(CVE-2024-38428)
   Severity: Critical
```

- Major [option] parameters include:
  - **--sec-severity={Critical,Important,Moderate,Low}**: queries security updates of a specific level. Values in the brackets can be any combination of security update levels.

    In the following example, **--sec-severity=Critical** is used to query critical security updates.

```
[root@localhost ~]# yum updateinfo list --sec-severity=Critical
Last metadata expiration check: 0:10:15 ago on Thu 08 Sep 2022 05:30:23 PM CST.
HCE2-SA-2022-0006 Critical/Sec. curl-7.79.1-2.h6.hce2.x86_64
HCE2-SA-2022-0003 Critical/Sec. libarchive-3.5.2-1.h2.hce2.x86_64
HCE2-SA-2022-0006 Critical/Sec. libcurl-7.79.1-2.h6.hce2.x86_64
...
```

    In the following example, **--sec-severity={Critical,Moderate}** is used to query critical and moderate security updates.

```
[root@localhost ~]# yum updateinfo list --sec-severity={Critical,Moderate}
Last metadata expiration check: 0:11:07 ago on Thu 08 Sep 2022 05:30:23 PM CST.
HCE2-SA-2022-0006 Critical/Sec. curl-7.79.1-2.h6.hce2.x86_64
```



```
HCE2-SA-2022-0011 Moderate/Sec. gnupg2-2.2.32-1.h6.hce2.x86_64
HCE2-SA-2022-0003 Critical/Sec. libarchive-3.5.2-1.h2.hce2.x86_64
...
```

- **--cve=**<CVE ID>: queries security updates of a specific CVE.

```
[root@localhost ~]# yum updateinfo info --cve=CVE-2024-38428
Last metadata expiration check: 0:11:10 ago on Wed 26 Mar 2025 11:08:19 AM CST.
================================================================================
  An update for wget is now available for HCE 2.0
================================================================================
  Update ID: HCE2-SA-2024-0262
       Type: security
    Updated: 2024-09-23 18:09:48
       CVEs: CVE-2024-38428
Description: Security Fix(es):
           :
           : url.c in GNU Wget through 1.24.5 mishandles semicolons in the userinfo subcomponent
of a URI, and thus there may be insecure behavior in which data that was supposed to be in the
userinfo subcomponent is misinterpreted to be part of the host subcomponent.
(CVE-2024-38428)
   Severity: Critical
```

NOTE

Use the **yum updateinfo --help** command to obtain more information.

6.5 Checking for Security Updates

- Run the **yum check-update --security** command to check for available security updates in the OS.

```
[root@localhost ~]# yum check-update --security
Last metadata expiration check: 0:13:32 ago on Wed 26 Mar 2025 11:08:19 AM CST.

c-ares.x86_64                      1.18.1-1.r5.hce2                   hceversion
curl.x86_64                        7.79.1-2.r34.hce2                  hceversion
dnsmasq.x86_64                       2.86-1.r20.hce2                    hceversion
expat.x86_64                       2.4.1-5.r8.hce2                    hceversion
…
```

- Run the **yum check-update --sec-severity={Critical,Important,Moderate,Low}** command to check for security updates of the specified level.

  Values in the brackets can be any combination of security update levels.

```
[root@localhost ~]#  yum check-update --sec-severity=Moderate
Last metadata expiration check: 0:15:20 ago on Wed 26 Mar 2025 11:08:19 AM CST.

c-ares.x86_64                      1.18.1-1.r5.hce2                   hceversion
curl.x86_64                        7.79.1-2.r34.hce2                  hceversion
expat.x86_64                       2.4.1-5.r8.hce2                    hceversion
gnutls.x86_64                      3.7.2-2.r31.hce2                   hceversion
gnutls-utils.x86_64                  3.7.2-2.r31.hce2                   hceversion
…
```

6.6 Installing Security Updates

- Run the **yum upgrade --security** command to install all security updates.

```
[root@localhost ~]# yum upgrade --security
Last metadata expiration check: 0:16:41 ago on Wed 26 Mar 2025 11:08:19 AM CST.
Dependencies resolved.
================================================================
Package       Arch       Version              Repository    Size
================================================================
```



```
Installing:
Kernel        x86_64        5.10.0-60.18.0.50.h498_2.hce2    hceversion        49 M
Upgrading:
Curl          x86_64        7.79.1-2.h6.hce2           hceversion        147 k
...
Transaction Summary
================================================================
Install   1 Package
Upgrade  22 Packages
Total download size: 69 M
Is this ok [y/N]:
```

- Run the **yum upgrade --sec-severity={Critical,Important,Moderate,Low}** command to install security updates of the specified level.

  Values in the brackets can be any combination of security update levels.

```
[root@localhost ~]# yum upgrade --sec-severity=Moderate
Last metadata expiration check: 0:32:27 ago on Thu 08 Sep 2022 05:30:23 PM CST.
Dependencies resolved.
================================================================
Package                 Architecture          Version             Repository       Size
================================================================
Upgrading:
gnupg2                x86_64               2.2.32-1.h6.hce2      hceversion          2.2 M
python3-unbound       x86_64               1.13.2-3.h2.hce2      hceversion           96 k
unbound-libs          x86_64               1.13.2-3.h2.hce2      hceversion          505 k
Transaction Summary
================================================================Upgrade       3
Packages
Total download size: 2.8 M
Is this ok [y/N]:
```

- Run **yum upgrade --advisory** =*<SA ID>* to install security updates of a specific advisory.

  Multiple packages are separated by commas (,).

```
root@localhost ~]# yum upgrade --advisory=HCE2-SA-2022-0033
Last metadata expiration check: 1:48:44 ago on Tue 13 Sep 2022 09:43:13 AM CST.
Dependencies resolved.
================================================================
Package             Architecture          Version               Repository          Size
================================================================
Upgrading:
python3-rpm           x86_64       4.17.0-8.h13.hce2       hceversion       87 k
rpm           x86_64       4.17.0-8.h13.hce2       hceversion       498 k
rpm-libs      x86_64       4.17.0-8.h13.hce2       hceversion       376 k
Transaction Summary
================================================================
Upgrade  3 Packages
Total download size: 962 k
Is this ok [y/N]:
```

- Run the **yum upgrade --cve=<CVE ID>** command to install security updates of a specific CVE.

  Multiple packages are separated by commas (,).

```
[root@localhost ~]# yum upgrade --cve=CVE-2021-28861
Last metadata expiration check: 5:16:36 ago on Tue 13 Sep 2022 09:43:13 AM CST.
Dependencies resolved.
================================================================
Package               Architecture    Version           Repository    Size
================================================================
Upgrading:
python3               x86_64          3.9.9-7.h10.hce2          hceversion          8.0 M
python3-unversioned-command    x86_64          3.9.9-7.h10.hce2          hceversion          3.9 k
Transaction Summary
================================================================
Upgrade  2 Packages
```



```
Total download size: 8.0 M
Is this ok [y/N]:
```


7 Obtaining the openEuler Extended Software Packages

By default, HCE does not load the openEuler repository to avoid conflicts with HCE software packages.

HCE 2.0 is only compatible with openEuler 22.03 LTS. This section describes how to obtain the extended software packages of openEuler 22.03 LTS.

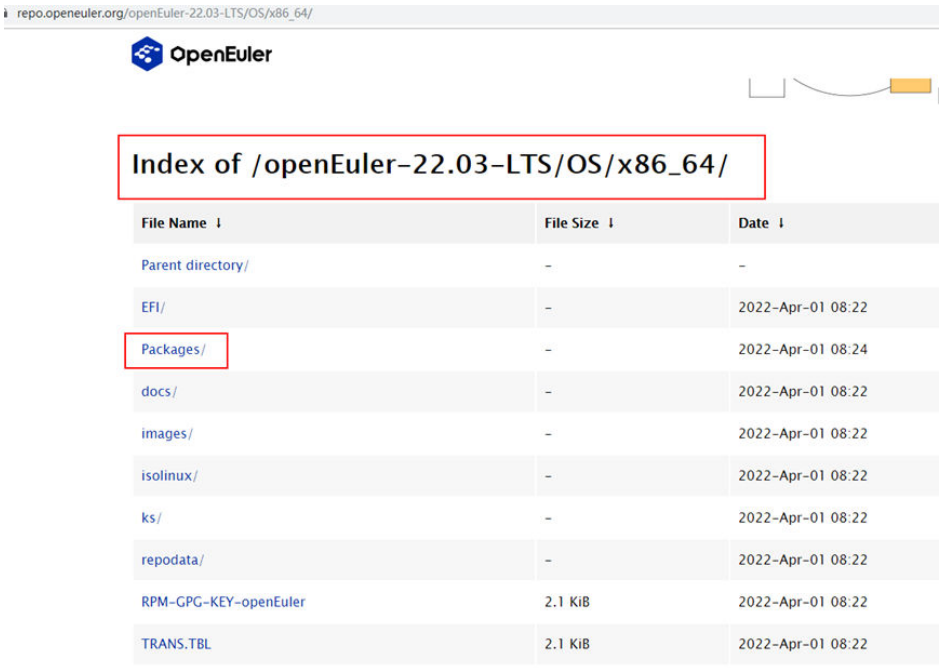
How to Obtain	Application Scenario	How to Install RPM Packages	How to Back Up and Restore Repository Files
Using wget to Download RPM Packages	Installing a few RPM packages	Manually download and install the RPM packages.	N/A
Batch Downloading RPM Packages Using the Repository File	Installing multiple RPM packages	The RPM packages are automatically installed. You do not need to download them again.	Back up the repository files in the **/etc/yum.repos.d** directory first and then delete them. Restore the repository files after the RPM packages are installed.

Using wget to Download RPM Packages

You can run the **wget** command to download the RPM packages. **hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm** is used as an example.

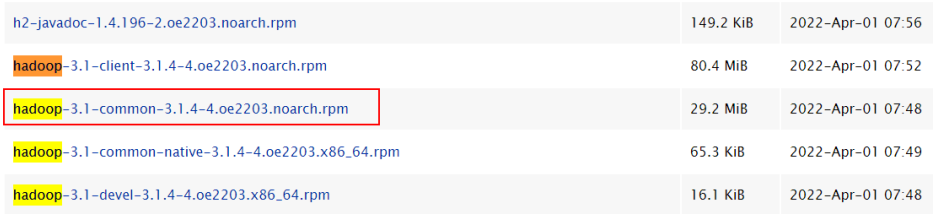
1. **Sign in to the openEuler community**.
2. In the **OS/** or **everything/** directory, select the **aarch64/** or **x86_64/** system architecture directory and open the **Packages/** directory.

Figure 7-1 Example



3. Search for the required RPM package, for example, **hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm**.

Figure 7-2 Example

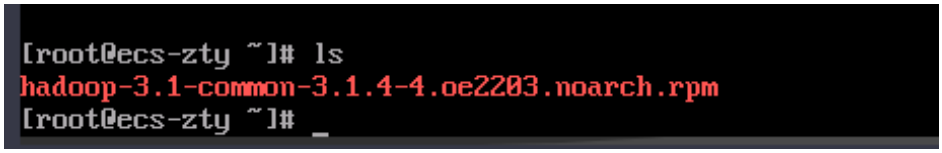


4. Right-click the RPM package, copy the download link, and run the **wget** command to download the RPM package.

Figure 7-3 Example

5. Check whether the download is successful.

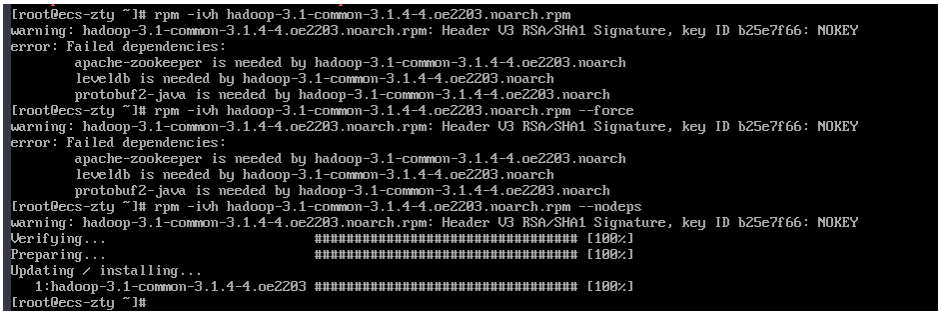
Figure 7-4 Example of successful download



6. Run the **rpm -ivh hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm** command to install the RPM package. If information similar to **Figure 7-5** is displayed, the package is installed.

   If other packages are required during the installation, repeat the above steps to install the dependent packages.

   Figure 7-5 Example of successful RPM package installation



Batch Downloading RPM Packages Using the Repository File

For example, download openEuler-22.03-LTS (x86_64) RPM packages and run **yum** to install them.

1. Ensure that your VM can access **https://repo.openeuler.org/openEuler-22.03-LTS/**.
2. Configure a yum repository.

   Go to the **/etc/yum.repos.d** directory, create an **openEuler.repo** file, and copy and paste the following content to this file.

   NOTE

   - The **openEuler.repo** file in **repo.openeuler.org/openEuler-22.03-LTS/update/x86_64/** conflicts with the HCE repository file. Back up the HCE repository file in the **/etc/yum.repos.d** directory and delete this file. Then, create a new **openEuler.repo** file.

```
[openEuler-everything]
name=openEuler everything repository
baseurl=https://repo.openeuler.org/openEuler-22.03-LTS/everything/x86_64
gpgcheck=1
enabled=1
priority=3
gpgkey=https://repo.openeuler.org/openEuler-22.03-LTS/everything/x86_64/RPM-GPG-KEY-openEuler
[openEuler-update]
name=openEuler update repository
baseurl=https://repo.openeuler.org/openEuler-22.03-LTS/update/x86_64/
gpgcheck=1
enabled=1
priority=3
gpgkey=https://repo.openeuler.org/openEuler-22.03-LTS/everything/x86_64/RPM-GPG-KEY-openEuler
```

3. Run the **yum clean all** command to delete the cache of the yum repository.
4. Run **yum makecache** to connect to the newly configured repository. If information similar to the following is displayed, the repository is connected.

Figure 7-6 Example of yum commands

```
[root@ecs-10ce-5810 ~]# yum clean all
13 files removed
[root@ecs-10ce-5810 ~]# yum makecache
openEuler everything repository                                 2.1 MB/s |  16 MB     00:07
openEuler update repository                                     977 kB/s |  85 MB     01:29
Last metadata expiration check: 0:00:11 ago on Thu 27 Mar 2025 02:12:27 PM CST.
Metadata cache created.
[root@ecs-10ce-5810 ~]# _
```

5. Install the RPM packages. **hadoop-3.1-common** is used as an example here.

   a. Run **yum list | grep hadoop-3.1-common** to check whether the package exists.

   Figure 7-7 Checking whether the hadoop-3.1-common package exists

   ```
   [root@ecs-zty ~]# yum list | grep hadoop-3.1-common
   hadoop-3.1-common.noarch                    3.1.4-4.oe2203                  @System
   hadoop-3.1-common-native.x86_64             3.1.4-4.oe2203                  everything
   [root@ecs-zty ~]#
   ```

   b. Run **yum -y install hadoop-3.1-common** to install the package. If the following information is displayed, the package is installed.

   Figure 7-8 Successful installation of the hadoop-3.1-common package using yum

   ```
   Installed:
     alsa-lib-1.2.5.1-1.oe2203.x86_64                atk-2.36.0-1.oe2203.x86_64
     cairo-1.17.4-1.oe2203.x86_64                    copy-jdk-configs-4.0-1.oe2203.noarch
     dejavu-fonts-2.37-1.oe2203.noarch               fontconfig-2.13.94-1.oe2203.x86_64
     fonts-filesystem-2.0.5-2.oe2203.noarch          fribidi-1.0.10-1.oe2203.x86_64
     gdk-pixbuf2-2.42.6-1.oe2203.x86_64              gdk-pixbuf2-modules-2.42.6-1.oe2203.x86_64
     graphite2-1.3.14-5.oe2203.x86_64                gtk-update-icon-cache-3.24.30-5.oe2203.x86_64
     gtk2-2.24.33-4.oe2203.x86_64                    hadoop-3.1-common-3.1.4-4.oe2203.noarch
     harfbuzz-2.8.2-1.oe2203.x86_64                  hicolor-icon-theme-0.17-4.oe2203.noarch
     java-1.8.0-openjdk-1:1.8.0.312.b07-11.oe2203.x86_64  java-1.8.0-openjdk-headless-1:1.8.0.312.b07-11.oe2203.x86_64
     javapackages-filesystem-5.3.0-6.oe2203.noarch   jbigkit-libs-2.1-17.oe2203.x86_64
     leveldb-1.20-4.oe2203.x86_64                    libX11-1.7.2-2.oe2203.x86_64
     libXau-1.0.9-2.oe2203.x86_64                    libXcomposite-0.4.5-1.oe2203.x86_64
     libXcursor-1.2.0-1.oe2203.x86_64                libXdamage-1.1.5-1.oe2203.x86_64
     libXext-1.3.4-2.oe2203.x86_64                   libXfixes-6.0.0-1.oe2203.x86_64
     libXft-2.3.4-1.oe2203.x86_64                    libXi-1.8-1.oe2203.x86_64
     libXinerama-1.1.4-5.oe2203.x86_64               libXrandr-1.5.2-1.oe2203.x86_64
     libXrender-0.9.10-10.oe2203.x86_64              libXtst-1.2.3-10.oe2203.x86_64
     libdatrie-0.2.13-1.oe2203.x86_64                libfontenc-1.1.4-2.oe2203.x86_64
     libjpeg-turbo-2.1.1-1.oe2203.x86_64             libthai-0.1.28-4.oe2203.x86_64
     libtiff-4.3.0-9.oe2203.x86_64                   libxcb-1.14-1.oe2203.x86_64
     lksctp-tools-1.0.19-1.oe2203.x86_64             lua-posix-35.0-1.oe2203.x86_64
     nspr-4.32.0-1.oe2203.x86_64                     nss-3.72.0-2.oe2203.x86_64
     nss-help-3.72.0-2.oe2203.x86_64                 nss-softokn-3.72.0-2.oe2203.x86_64
     nss-util-3.72.0-2.oe2203.x86_64                 pango-1.49.3-2.oe2203.x86_64
     pixman-0.40.0-1.oe2203.x86_64                   protobuf2-java-2.5.0-3.oe2203.x86_64
     tzdata-java-2021e-2.oe2203.noarch               xorg-x11-font-utils-1:7.5-43.oe2203.x86_64
     xorg-x11-fonts-others-7.5-24.oe2203.noarch      zookeeper-3.6.2-2.4.oe2203.noarch

   Complete!
   ```

6. Restore the repository file.

   After installing the required openEuler packages, delete the openEuler.repo file and restore the repository file deleted in step 2.

8 Creating a Docker Image and Starting a Container

This section uses HCE 2.0 as an example to describe how to create a Docker image of HCE and start the container on HCE.

Constraints

- The version of HCE running the container image must be the same as that of the created container image.

Creating an Image Archive File

1. Confirm that the repository is configured correctly.

   Check whether the parameters in the **/etc/yum.repos.d/hce.repo** file are configured correctly. The correct configuration is as follows:

   ```
   [base]
   name=HCE $releasever base
   baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
   enabled=1
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

   [updates]
   name=HCE $releasever updates
   baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
   enabled=1
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2

   [debuginfo]
   name=HCE $releasever debuginfo
   baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
   enabled=0
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
   ```

2. Create a new directory as the root file system of Docker images, for example, **/tmp/docker_rootfs**. Install the software package in this directory.

   ```
   mkdir -p /tmp/docker_rootfs
   yum --setopt=install_weak_deps=False --installroot /tmp/docker_rootfs --releasever 2.0 install bash yum coreutils security-tool procps-ng vim-minimal tar findutils filesystem hce-repos hce-rootfiles cronie -y
   ```


CAUTION

By default, the **yum** command installs the software package of the current HCE version. To install the software package of other HCE version, you can use **--releasever** to specify the version. For example, the command above is used to install the software package of HCE 2.0.

3. Use **chroot** to go to the temporary directory.
```
chroot /tmp/docker_rootfs
```
4. Configure the temporary directory.
   a. Execute **security-tool.sh** to disable unnecessary services.
```
export EULEROS_SECURITY=0
echo "export TMOUT=300" >> /etc/bashrc
/usr/sbin/security-tool.sh -d / -c /etc/hce_security/hwsecurity/hce_security_install.conf -u /etc/hce_security/usr-security.conf -l /var/log/hce-security.log -s
```
   During the execution, it is normal if the errors similar to **Figure 8-1** are displayed. The errors can be:
   - The service file was not found. The service is not started in the chroot file system.
   - The **/etc/sysconfig/init** file for booting the system was not found. The tool disables services during system startup. The image rootfs is not involved in system startup.
   - The **/proc/sys/kernel/sysrq** file was not found. This file is used for calling after the system is started and does not exist in the chroot file system.

Figure 8-1 Error messages

```
[root@localhost /]# /usr/sbin/security-tool.sh -d / -c /etc/hce_security/hwsecurity/hce_
security_install.conf -u /etc/hce_security/usr-security.conf -l /var/log/hce-security.lo
g -s
Failed to disable unit, unit avahi-daemon.service does not exist.
Failed to disable unit, unit avahi-daemon.socket does not exist.
Failed to disable unit, unit snmpd.service does not exist.
Failed to disable unit, unit squid.service does not exist.
Failed to disable unit, unit smb.service does not exist.
Failed to disable unit, unit vsftpd.service does not exist.
Failed to disable unit, unit tftp.service does not exist.
Failed to disable unit, unit nfs-server.service does not exist.
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
Failed to disable unit, unit rpcbind.service does not exist.
Failed to disable unit, unit slapd.service does not exist.
Failed to disable unit, unit dhcpd.service does not exist.
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
Failed to enable unit, unit haveged.service does not exist.
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
Failed to disable unit, unit postfix.service does not exist.
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
Failed to disable unit, unit chronyd.service does not exist.
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
tail: cannot open '//etc/sysconfig/init' for reading: No such file or directory
package at is not installed
[security-tool.sh:839] [error] package at does not exist
cronie-1.5.7-1.r2.hce2.x86_64
sysctl: cannot stat /proc/sys/kernel/sysrq: No such file or directory
Removed /etc/systemd/system/multi-user.target.wants/hce-security.service.
```


b. Uninstall the security-tool, cronie, and systemd software packages and their dependent software packages.

```
cp -af /etc/pam.d /etc/pam.d.bak
rm -f /etc/yum/protected.d/sudo.conf /etc/yum/protected.d/systemd.conf
yum remove -y security-tool cronie systemd
rpm -e --nodeps logrotate crontabs
rm -rf /etc/pam.d
mv /etc/pam.d.bak /etc/pam.d
sh -c 'shopt -s globstar; for f in $(ls /**/*.rpmsave); do rm -f $f; done' &> /dev/null
[ -d /var/lib/dnf ] && rm -rf /var/lib/dnf/*
[ -d /var/lib/rpm ] && rm -rf /var/lib/rpm/__db.*
```

c. Remove the **/boot** directory.

```
rm -rf /boot
```

d. Set the container image language to en_US.

```
cd /usr/lib/locale;rm -rf $(ls | grep -v en_US | grep -vw C.utf8 )
rm -rf /usr/share/locale/*
```

e. Remove shared files **man**, **doc**, **info**, and **mime**.

```
rm -rf /usr/share/{man,doc,info,mime}
```

f. Remove the cached log files.

```
rm -rf /etc/ld.so.cache
[ -d /var/cache/ldconfig ] && rm -rf /var/cache/ldconfig/*
[ -d /var/cache/dnf ] && rm -rf /var/cache/dnf/*
[ -d /var/log ] && rm -rf /var/log/*.log
```

g. Remove the Java security certificate.

```
rm -rf /etc/pki/ca-trust/extracted/java/cacerts /etc/pki/java/cacerts
```

h. Remove **/etc/machine-id**.

```
rm -rf /etc/machine-id
```

i. Remove **/etc/mtab**.

```
rm -rf /etc/mtab
```

5. Exit from the chroot file system.

```
exit
```

6. Compress the temporary directory and generate the Docker image archive file **hce-docker.x86_64.tar.xz**.

   The archive path is **/tmp/docker_rootfs/hce-docker.x86_64.tar.xz**.

```
pushd /tmp/docker_rootfs/
tar cvf hce-docker.x86_64.tar .
xz hce-docker.x86_64.tar
popd
```

Starting a Container Using an Image Archive File

1. Confirm that the repository is configured correctly.

   Check whether the parameters in the **/etc/yum.repos.d/hce.repo** file are configured correctly. The correct configuration is as follows:

```
[base]
name=HCE $releasever base
baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

[updates]
name=HCE $releasever updates
baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2

[debuginfo]
```



```
name=HCE $releasever debuginfo
baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
enabled=0
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
```

2. Install the Docker software package.

```
yum install docker -y
```

3. Use the image archive file to create a container image.

```
mv /tmp/docker_rootfs/hce-docker.x86_64.tar.xz .
docker import hce-docker.x86_64.tar.xz
```

Run **docker images** to check the container image ID. In this example, the container image ID is **6cfefae3a541**.

Figure 8-2 Checking the container image ID

```
[root@localhost /]# mv /tmp/docker_rootfs/hce-docker.x86_64.tar.xz .
[root@localhost /]# docker import hce-docker.x86_64.tar.xz
sha256:6cfefae3a5410e3b48208f8a9e0a28fc08fa1b3a62ad39b27196a742969d5bfc
[root@localhost /]#
[root@localhost /]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>              6cfefae3a541        6 seconds ago       181MB
```

NOTE

To create an image, you can run the following command to specify the **REPOSITORY** and **TAG** parameters:

docker import [OPTIONS] file|URL|- [REPOSITORY[:TAG]]

4. Use the image to run containers and enter the bash environment.

If the shell view changes after you run the following command, you have entered the bash environment of the containers: **6cfefae3a541** is the image ID.

```
docker run -it 6cfefae3a541 bash
```


9 Tools

9.1 BiSheng Compiler

BiSheng compiler is a high-performance, high-reliability, and easy-to-expand compiler developed by Huawei. BiSheng compiler has introduced multiple compilation technologies and supports programming languages C, C++, and Fortran.

Constraints

The HCE native Clang compiler cannot work with the Clang compiler of BiSheng. If you have installed the native Clang compiler, do not install the BiSheng compiler anymore.

If you have installed the BiSheng compiler but want to use the native Clang compiler, run **rpm -e bisheng-compiler** to delete the BiSheng compiler, and then open a new terminal to use the native Clang compiler.

Installing BiSheng Compiler

1. Confirm that the repository is configured correctly.

   Check whether the parameters in the **/etc/yum.repos.d/hce.repo** file are configured correctly. The correct configuration is as follows:

   ```
   [base]
   name=HCE $releasever base
   baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
   enabled=1
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

   [updates]
   name=HCE $releasever updates
   baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
   enabled=1
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2

   [debuginfo]
   name=HCE $releasever debuginfo
   baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
   enabled=0
   ```



```
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
```

2. Run **yum -y install bisheng-compiler** to install BiSheng compiler.
3. Run **source /usr/local/bisheng-compiler/env.sh** to import environment variables.

   NOTE

   If you have opened a new terminal, import the environment variables into the new terminal.

4. Check whether BiSheng compiler has been installed.

   Run **clang –v** to view the version number. If the command output contains the BiSheng compiler version, the BiSheng compiler has been successfully installed.

Using BiSheng Compiler

1. Compile and run a C/C++ program. A C program is used as an example.
   a. Add the following content to the **hello.c** file:
   ```
   #include <stdio.h>

   int main() {
       printf("Hello, World! This is a C program.\n");
       return 0;
   }
   ```
   b. Compile **hello.c**.
   ```
   clang hello.c -o hello
   ```
   c. Run **hello.o**.
   ```
   ./hello
   ```
2. Compile and run a Fortran program.
   a. Add the following content to the **hello.f90** file:
   ```
   program hello
       print *, "Hello, World! This is a Fortran program."
   end program hello
   ```
   b. Compile **hello.f90**.
   ```
   flang hello.f90 -o hello.o
   ```
   c. Run **hello.o**.
   ```
   ./hello.o
   ```
3. Specify a linker.

   Specify the LLVM lld for BiSheng compiler. If you do not specify the LLVM lld, the default linker ld will be used.

   ```
   clang -fuse-ld=lld hello.c -o hello.o
   ./hello.o
   ```

Upgrade and Rollback

1. Upgrade
   ```
   yum upgrade bisheng-compiler
   ```
2. Rollback
   ```
   yum downgrade bisheng-compiler
   ```

   For a cross-version upgrade of the BiSheng compiler, the rollback may fail due to file conflicts. You can manually delete the conflicting files and then run the rollback command again.

9.2 Workload Accelerator

9.2.1 Overview

Workload Accelerator is a tool provided by Huawei Cloud for application optimization.

It works in two ways:

- Static acceleration

  Static acceleration collects the PMU monitoring information on the CPU when an application is running, and statically builds a new high-performance binary based on the collected information. It may require only adjustments to compiler parameters, without involving any other code modifications. There are two optimization methods in static acceleration.

  - **Use the native BOLT tool**: Only fixed parameter combinations can be used to optimize applications.
  - **Run the hce-wae-auto commands**: Different parameter combinations can be generated based on the user-defined parameter range for application optimization.
- Dynamic acceleration

  Dynamic acceleration directly accelerates the application process without interrupting services.

Table 9-1 Advantages and disadvantages of the optimization methods

Optimization Method	Pros	Cons
Static acceleration	Applications are optimized on the basis of binary executable files, and the program code does not need to be modified.	After the optimization, you need to restart the applications.
Dynamic acceleration	Application processes are optimized directly. Applications do not need to be restarted. Optimization results can be saved using application snapshots. In addition, binary file source tracing can be ensured, and application processes can be continuously optimized for iteration until the performance improvement reaches the bottleneck.	Data can only be collected using instrumentation, and optimization can be performed only once.

In addition, a CPU feature setting tool is provided for Kunpeng 920 V200 chips. You can configure chip features based on your service requirements. This way, you can optimize the system through both hardware and software improvements.

Constraints

- Only x86 HCE can use static acceleration and dynamic acceleration.
- Only Arm HCE can use the CPU feature setting tool on Kunpeng 920 V200.
- Only the **root** user can use Workload Accelerator.

Process for Using Workload Accelerator to Optimize Applications

1. **Install the Workload Accelerator**.
2. Optimize the application through **static acceleration** or **dynamic acceleration**.

9.2.2 Installing Workload Accelerator

1. Confirm that the repository is configured correctly.

   Check whether the parameters in the **/etc/yum.repos.d/hce.repo** file are configured correctly. The correct configuration is as follows:

   ```
   [base]
   name=HCE $releasever base
   baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
   enabled=1
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

   [updates]
   name=HCE $releasever updates
   baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
   enabled=1
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2

   [debuginfo]
   name=HCE $releasever debuginfo
   baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
   enabled=0
   gpgcheck=1
   gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
   ```

2. Run **yum -y install hce-wae** to install Workload Accelerator.
3. Check whether Workload Accelerator has been installed.

   If information similar to **Figure 9-1** is displayed after you run the following command, Workload Accelerator is installed successfully:

   ```
   llvm-bolt --help | more
   ```

   Figure 9-1 Example output

   ```
   [root@localhost sdb]# llvm-bolt --help | more
   OVERVIEW: BOLT - Binary Optimization and Layout Tool

   USAGE: llvm-bolt [options] <executable> <executable>

   OPTIONS:

   BOLT generic options:

     --bolt-id=<string>                                - add any string to tag this execution in the output binary via bolt info section
     --data=<string>                                   - <data file>
     --data2=<string>                                  - <data file>
     --deterministic-debuginfo                         - disables parallel execution of tasks that may producenondeterministic debug info
     --dwarf-output-path=<string>                      - Path to where .dwo files or dwp file will be written out to.
     --dyno-stats                                      - print execution info based on profile
     --enable-bat                                      - write BOLT Address Translation tables
     --hot-data                                        - hot data symbols support (relocation mode)
     --hot-functions-at-end                            - if reorder-functions is used, order functions putting hottest last
   ```


NOTE

The output may vary depending on the tool version.

9.2.3 Static Acceleration

Preparations

1. Run the following command to check whether the binary file to be optimized can be located again:

   application can be replaced with the binary file to be checked.

   ```
   readelf -a application | grep .rela.text
   ```

   Figure 9-2 Example of a readelf command

   ```
   [root@hce2 shard]# readelf -a gemini-redis-server | grep .rela.text
     [15] .rela.text         RELA            0000000000000000  0750a490
   Relocation section '.rela.text' at offset 0x750a490 contains 241690 entries:
   [root@hce2 shard]#
   ```

   - If the binary file contains **.rela.text**, the file can be located again. In this case, the application can be optimized.
   - If the binary file does not contain **.rela.text**, to allow BOLT to re-arrange functions in your program, you need to add **--emit-relocs** or **-q** to the command that was used to create the binary file.

2. Collect the logs when the application is running.

   After deploying and preheating the application, you can run **llvm-bolt -instrument -o -instrumentation-file** to configure the way how the logs are collected.

   For example, to collect logs every 30 seconds after the **test.so** file is executed, run the following command to save the logs to the **test.log** file:

   ```
   llvm-bolt tests.so -instrument -o testd.so -instrumentation-file=test.log -instrumentation-sleep-time=30 -instrumentation-no-counters-clear
   ```

   - **instrument**: indicates the new dynamic library file generated after the log collection way is configured. In this example, the newly generated dynamic library is **testd.so**.
   - **instrumentation-file**: indicates the name of the file where the logs are saved. In this example, the log file is **test.log**.
   - **instrumentation-sleep-time**: indicates the interval for collecting logs, in seconds. In this example, logs are collected every 30 seconds.
   - **instrumentation-no-counters-clear**: indicates that the log counter information is not cleared after each log collection to ensure that the log context is continuous.

3. Run the application corresponding to **testd.so**. The run logs of the application are automatically saved in the **test.log** file.

   Workload Accelerator will optimize the application based on the dynamic data in the **test.log** file.

Procedure

- Use the native BOLT tool to optimize the application.

Once you have **test.log** ready, you can use it for optimizations with BOLT. For example, you can run the following command:

```
llvm-bolt <executable> -o <executable>.bolt -data=test.log -reorder-blocks=ext-tsp -reorder-functions=hfsort -split-functions -split-all-cold -split-eh -dyno-stats
```

NOTE

The parameters in the command are for illustration purposes and are not the optimal combination of parameters.

- Run the **hce-wae-auto** command to optimize the application.

  The **hce-wae-auto** command optimizes the application based on user-defined **configuration file**.

  The command format is **hce-wae-auto [-h] [-c <Path>] [-s <Keyword>] [-e <Pattern>] [-l] [--free]** *Application*.

  NOTE

  - After you run the **hce-wae-auto command**, the path of the generated parameter set is the same as the binary output path (/data/hce-wae-auto/hce-wae-auto.data) in the configuration file by default.
  - After you run the **hce-wae-auto** command, a log file is automatically generated, and its default path is **/var/log/hce-wae-auto/hce-wae-auto.log**.

Table 9-2 Parameter description

Parameter	Value Required	Value Type	Mandatory	Description
-h	No	/	No	Displays command parameter help information.
-c	Yes	String	No	Defines the configuration file path. The default path is **/etc/hce-wae-auto.conf**.
-s	Yes	String	No	Defines the keyword for fuzzy search of the last parameter set. If there is no saved parameter set, this parameter will not take effect.

Parameter	Value Required	Value Type	Mandatory	Description
-e	Yes	String	No	Executes the selected parameter combination based on the content of the last parameter set. A sequence number must be specified. ● A single number specifies the parameter combination corresponding to the sequence number. ● The colon (:) indicates the sequence number range. For example: – **4:6** indicates sequence numbers 4 to 6. – **:7** indicates sequence numbers 1 to 7. – **5:** indicates sequence numbers from 5 to the end. – **:** indicates all sequence numbers. ● A single number can be used together with a sequence number range (:) and separated using a comma (,). For example: – **1,4:6** indicates sequence numbers 1, 4, 5, and 6.
-l	No	/	No	Generates and prints the parameter set based on the configuration file. The command is not automatically executed and the parameter set is not saved.
--free	No	/	No	Indicates that the validity of parameters in the configuration file is not verified. Workload Accelerator performs simple verification on the parameters in the configuration file to determine whether the parameters are used for optimization using BOLT. If you run the **--free** command, the verification is canceled and parameters that are not for optimization can be entered.

Parameter	Value Required	Value Type	Mandatory	Description
application	No	/	Yes	Specifies the binary file to be executed. You can add a relative path or an absolute path. Example: • mysqld • /usr/bin/mysqld • ../bin/mysqld

NOTE

If the **-c**, **-s**, **-e** and **-l** parameters conflict, **-s** has the highest priority, followed by **-l** and then **-e**, and **-c** has the lowest priority.

The following table describes some example commands.

Table 9-3 hce-wae-auto commands

Example Command	Description
hce-wae-auto mysqld	Reads the configuration from the default path, generates a parameter set based on configuration parameters, automatically executes the command, and saves the parameter set after the command is executed.
hce-wae-auto mysqld -c /etc/my.conf -l	Reads configuration parameters from the **/etc/my.conf** path, generates a parameter set, prints the generated command, but does not execute the command or save the parameter set.
hce-wae-auto mysqld -c /etc/my.conf	Reads configuration parameters from the **/etc/my.conf** path, generates a parameter set, automatically executes the command, and saves the parameter set after the command is executed.
hce-wae-auto mysqld -s align	Searches for the keyword **align** in fuzzy mode from the parameter set generated last time and prints the matched parameter combination.
hce-wae-auto mysqld -e 4:6	Selects the parameter combinations of No.4, No.5, and No.6 from the parameter set generated last time to generate a binary file.
hce-wae-auto mysqld -e 1,4:6	Selects the parameter combinations of No.1, No.4, No.5, and No.6 from the parameter set generated last time to generate a binary file.

Example Command	Description
hce-wae-auto mysqld -e :	Selects all parameter combinations from the parameter set generated last time to generate a binary file.
hce-wae-auto mysqld -e 4:6 -l	Selects the parameter combinations of No.4, No. 5, and No. 6 from the parameter set generated last time, prints the generated command, but does not execute the command.
hce-wae-auto mysqld -e : -l	Selects all parameter combinations from the parameter set generated last time, prints the generated command, but does not execute the command.
hce-wae-auto mysqld --free	Reads the configuration from the default path but does not verify parameters, generates a parameter set based on configuration parameters, automatically executes the command, and saves the parameter set after the command is executed.
hce-wae-auto -h	Displays command help information.

9.2.4 Dynamic Acceleration (Only for HCE 2.0)

Preparations

Before dynamic acceleration, you need to perform two checks. Applications can be dynamically accelerated only when both conditions are met.

1. Run the following command to check whether the binary file to be optimized can be located again:

   application can be replaced with the binary file to be checked.

   ```
   readelf -a application | grep .rela.text
   ```

   Figure 9-3 Example of a readelf command

   ```
   [root@hce2 shard]# readelf -a gemini-redis-server | grep .rela.text
     [15] .rela.text         RELA            0000000000000000  0750a490
   Relocation section '.rela.text' at offset 0x750a490 contains 241690 entries:
   [root@hce2 shard]#
   ```

   - If the binary file contains **.rela.text**, the file can be located again. In this case, the application can be optimized.
   - If the binary file does not contain **.rela.text**, to allow BOLT to re-arrange functions in your program, you need to add **--emit-relocs** or **-q** to the command that was used to create the binary file.

2. Run **hce-wae --check /data/apps/mysql-8.0.28/bin/mysqld** to check whether the application supports dynamic acceleration.

   If **3** is displayed, dynamic acceleration is supported.

Figure 9-4 Example of the check result 3

```
[root@localhost ~]# hce-wae --check /data/apps/mysql-8.0.28/bin/mysqld
2023-09-14 20:41:21,968-INFO: Log level: INFO
2023-09-14 20:41:38,425-INFO: check /data/apps/mysql-8.0.28/bin/mysqld result: 3
```

Procedure

In the following operations, the mysqld application in the **/data/apps/mysql-8.0.28/bin** directory is used as an example.

1. Generate an instrumentation application and run it.
   a. Run the **/data/hce-wae/dbo/gen_instrumentation /data/apps/mysql-8.0.28/bin/mysqld** command to generate an instrumentation application.

      Command format: **/data/hce-wae/dbo/gen_instrumentation***Application path*

      Figure 9-5 Example of /data/hce-wae/dbo/gen_instrumentation

```
[root@localhost ~]# /data/hce-wae/dbo/gen_instrumentation /data/apps/mysql-8.0.28/bin/mysqld
BOLT-INFO: Target architecture: x86_64
BOLT-INFO: BOLT version: b93bff771fd4
BOLT-INFO: first alloc address is 0x400000
BOLT-INFO: creating new program header table at address 0x4200000, offset 0x3e00000
BOLT-WARNING: debug info will be stripped from the binary. Use -update-debug-sections to keep it.
BOLT-INFO: enabling relocation mode
BOLT-INFO: forcing -jump-tables=move for instrumentation
BOLT-INFO: enabling -align-macro-fusion=all since no profile was specified
BOLT-INFO: enabling lite mode
BOLT-WARNING: split function detected on input : _ZL28delete_dictionary_tablespacev.cold/1. The support is limited in re
location mode.
BOLT-INFO: 0 out of 69377 functions in the binary (0.0%) have non-empty execution profile
BOLT-INFO: the input contains 6637 (dynamic count : 0) opportunities for macro-fusion optimization that are going to be
fixed
BOLT-INSTRUMENTER: Number of indirect call site descriptors: 37513
BOLT-INSTRUMENTER: Number of indirect call target descriptors: 87025
BOLT-INSTRUMENTER: Number of function descriptors: 68951
BOLT-INSTRUMENTER: Number of branch counters: 686408
BOLT-INSTRUMENTER: Number of ST leaf node counters: 481684
BOLT-INSTRUMENTER: Number of direct call counters: 248150
BOLT-INSTRUMENTER: Total number of counters: 1416242
BOLT-INSTRUMENTER: Total size of counters: 11329936 bytes (static alloc memory)
BOLT-INSTRUMENTER: Total size of string table emitted: 7829557 bytes in file
BOLT-INSTRUMENTER: Total size of descriptors: 70240472 bytes in file
BOLT-INSTRUMENTER: Profile will be saved to file /data/hce-wae/dbo/tmp/perf.fdata
BOLT-INFO: 425568 instructions were shortened
BOLT-INFO: removed 11295 empty blocks
BOLT-INFO: UCE removed 143092 blocks and 8769293 bytes of code.
BOLT-INFO: SCTC: patched 0 tail calls (0 forward) tail calls (0 backward) from a total of 0 while removing 0 double jump
s and removing 0 basic blocks totalling 0 bytes of code. CTCs total execution count is 0 and the number of times CTCs ar
e taken is 0.
BOLT-INFO: output linked against instrumentation runtime library, lib entry point is 0xd0616e0
BOLT-INFO: clear procedure is 0xd05fe10
BOLT-INFO: setting _end to 0x3c83b18
BOLT-INFO: setting _end to 0x3c83b18
BOLT-INFO: patched build-id (flipped last bit)
[root@localhost ~]#
```

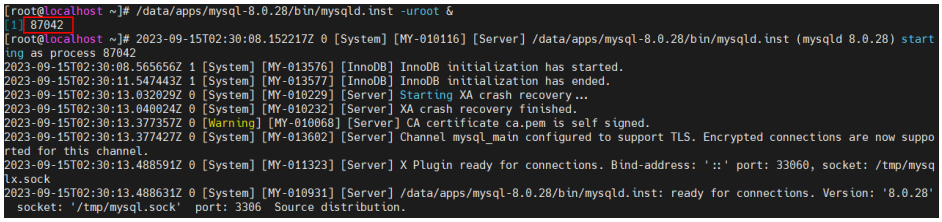
      The **mysqld.inst** instrumentation file is generated in the current directory.

      Figure 9-6 Example of querying an instrumentation file with the .inst suffix

```
[root@localhost ~]# ls -al *.inst
-rwxrwxrwx. 1 root root 304995968 Sep 15 10:24 mysqld.inst
[root@localhost ~]#
```

   b. Run the instrumentation file to obtain the PID of the application process. In this example, the PID is 87042.

Figure 9-7 Example of running an instrumentation file to obtain a PID



2. Create a dynamic acceleration configuration file for mysqld.

   Each application to be optimized must have a configuration file. Workload Accelerator dynamically accelerates the application based on the configuration file.

   a. Run the following command to copy the default configuration file **/data/hce-wae/config/mysqld.conf**:

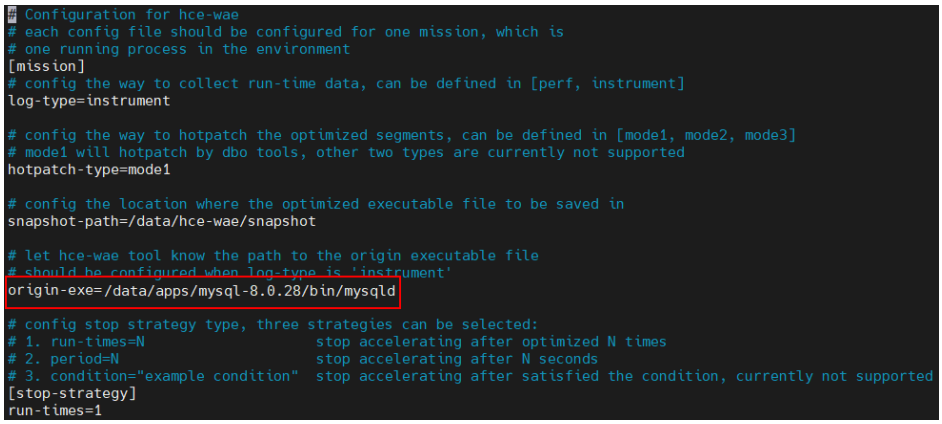
   ```
   [root@localhost]# cp /data/hce-wae/config/hce-wae-tmp.conf /data/hce-wae/config/mysqld.conf
   ```

   b. Set the **origin-exe** field in the **/data/hce-wae/config/mysqld.conf** configuration file.

   origin-exe indicates the location of the application to be optimized. In this example, the location is **/data/apps/mysql-8.0.28/bin/mysqld**.

   ```
   [root@localhost]# vim /data/hce-wae/config/mysqld.conf
   ```

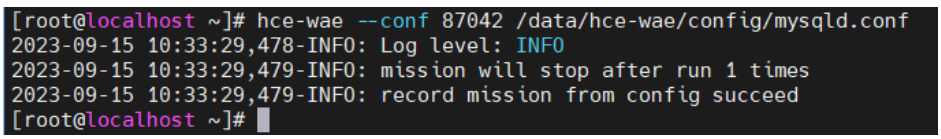
Figure 9-8 Example of the origin-exe field



3. Use the configuration file and the PID to configure dynamic acceleration.

   Command format: **hce-wae --conf [PID] [/path/to/config]**

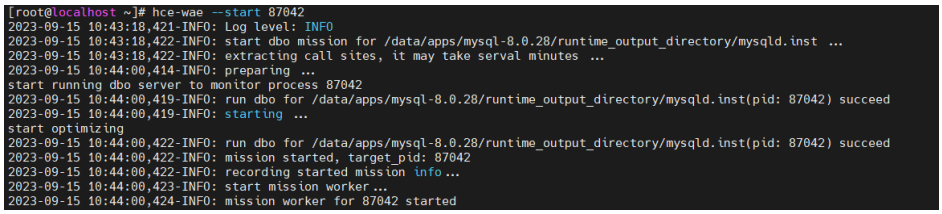
Figure 9-9 Example of configuring dynamic acceleration



4. Enable dynamic acceleration to optimize the instrumentation application.

   Command format: **hce-wae --start [PID]**

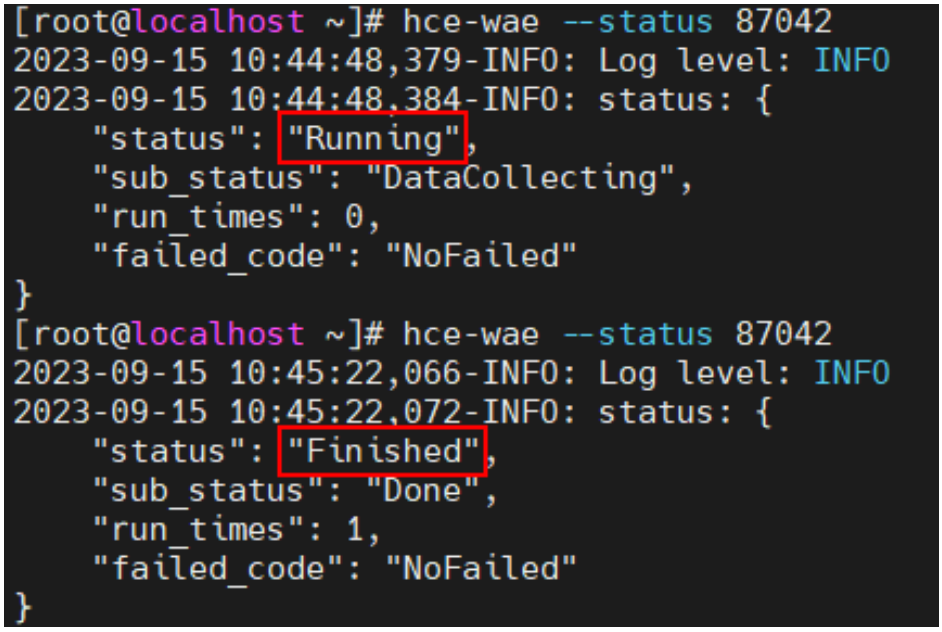
Figure 9-10 Enabling dynamic acceleration



You can view the optimization status from the **--status** parameter. If the status is **Running**, the process is being optimized. If the status is **Finished**, the process has been optimized.

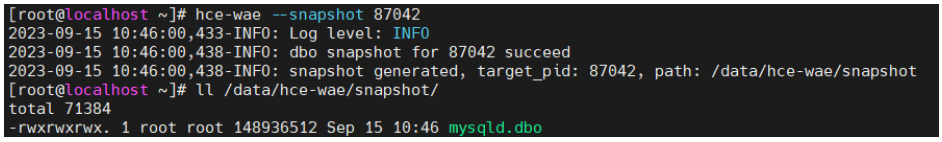
Command format: **hce-wae --status [PID]**

Figure 9-11 Viewing the optimization status



5. Use the **--snapshot** parameter to generate an optimized **.dbo** binary snapshot file. In this example, the file is **mysqld.dbo**.

Figure 9-12 Example of generating an optimized .dbo binary snapshot file



The default snapshot path is **/data/hce-wae/snapshot/**. You can change the path in the configuration file as needed. You can use this snapshot file to run the application without repeated optimization.

6. Terminate dynamic acceleration to stop application optimization.

   Command format: **hce-wae --stop [PID]**

Figure 9-13 Example of stopping dynamic acceleration

```
[root@localhost ~]# hce-wae --stop 87042
2023-09-15 10:46:20,867-INFO: Log level: INFO
stop dbo server successfully!
2023-09-15 10:46:20,871-INFO: dbo stop for 87042 succeed
2023-09-15 10:46:20,871-INFO: mission stopped, target_pid: 87042
```

CLI for Dynamic Acceleration

A CLI is provided for dynamic acceleration. **Table 9-4** lists the supported commands.

Figure 9-14 Startup page of dynamic acceleration

```
[root@localhost ~]# hce-wae --console

Welcome to hce-wae!

Version     : 1.0.0
Release     : 0.0.16.hce2
Architecture: x86_64


Type:  'h' or 'help' for help with commands
       'quit' to quit

hce-wae>
```

Figure 9-15 Help page of dynamic acceleration

```
hce-wae> help

h, help                   |    Show this help message and exit

list                      |    List the process information of the current environment,
                          |  including PID, PPID, and command

check <Binary file path> |    Check if the application support static or dynamic
                          |  acceleration. 0: both not support; 1: static only; 2:
                          |  dynamic only; 3: both are supported.

show <Pid>                |    Show all non-kernel processes which contain one of the
                          |  shared-object libraries that the target process
                          |  depends on.

conf <Pid> <Config path> |    Config the target process by pid

conf <Process>            |    Set the Config of the target process

start <Pid>               |    Start dynamic acceleration for process by its pid

stop <Pid>                |    Stop dynamic acceleration for process by its pid

status <Pid>              |    Show the status of the dynamic accelerating process by
                          |    its pid

snapshot <Pid>            |    Make a snapshot for the dynamic accelerating process
                          |    by its pid

quit                      |    Exit console
hce-wae>
```


Table 9-4 CLI commands supported for dynamic acceleration

Command	How to Use	Description
list	list	Obtains the process information in the current environment, including the PID, PPID, and command information.
status	status <PID>	Queries ongoing dynamic acceleration tasks and their status. The following information is returned: • PID • Process name • Optimization count • Acceleration status
check	check <PID>	Checks whether the current process supports static or dynamic acceleration and displays the text result.
show	show <PID>	Queries the process dependencies.
conf	conf <PID>	Configures the dynamic acceleration capability for the target process. After you run the command, configuration options are displayed in sequence. Configure the options as prompted.
start	start <PID>	Enables dynamic acceleration.
stop	stop <PID>	Stops dynamic acceleration.
snapshot	snapshot <PID>	Generates a dynamic acceleration snapshot.
quit	quit	Exits the CLI.
h/help	h/help	Shows help information.

9.2.5 Configuration File

This section describes each configuration item in the **static acceleration configuration file** and **dynamic acceleration configuration file**.

Static Acceleration Configuration File

The following figure shows the default configuration of static acceleration. You can customize a configuration file to optimize applications.

Figure 9-16 Example of default configuration for static acceleration

```
# Section `binary` defined options associated with binary file
[binary]
# Output path for generated binary files, absolute path is recommended
# default: /data/hce-wae-auto/
binary_out_path = "/data/hce-wae-auto"

# Threshold for the number of generated binary file, currently not in use
# default value: 100
binary_num_threshold = 100

# Name suffix for auto-generated binary files
# default value: `blot.auto`
binary_file_suffix = "blot.auto"

# Section `parameter` defined option associated with user defined parameter collection
[parameter]
# User defined parameter collection, parameters in this option will automatically be separated to different
# combination parameter groups, which are used as the parameter input for `llvm-blot` respectively, and
# generate different binary files.
# For those parameters which can be assigned values, use `=` connect parameter name and parameter value.
# Each line should contain one parameter and should not configure the same parameters in this option.
# example:
#   --plt=all
parameters =
    align-blocks
    frame-opt
    align-functions=1

# Section `include` defined include filter for auto-generated parameter collections
[include]
# options here should be the parameters combination that should be included for the auto-generated
# parameter group.
# Each line should contain one parameter, if the parameter is not assigned a value,
# it must end with `=`
# example:
#   frame-opt=none
align-blocks=

# Section `exclude` defined exclude filter for auto-generated parameter collections
[exclude]
# Options here should be the parameters combination that should be excluded for the auto-generated
# parameter group.
# Options has the same formate rule with the `include` section
frame-opt=none
```

Table 9-5 Configuration information

Module	Description
binary	Binary file attributes. For details about the binary parameters, see Table 9-6.
parameter	A collection of user-defined parameters. A parameter set is generated based on this collection of parameters. At least one parameter must be defined. You can run the **llvm-bolt -h** command to view all parameters.
include	Parameters that must be contained in the collection of user-defined parameters. Multiple parameters are allowed. The logical relationship between parameters is **AND**. The key of the configuration item is the parameter name, and the value is specified, for example, **frame-opt=none**. **NOTE** • If the value cannot be specified or does not need to be specified, end the key with an equal sign (=), for example, **frame-opt=**. • If **include** and **exclude** contain the same parameter, **exclude** precedes **include**.

Module	Description
exclude	Parameters that do not need to be contained in the collection of user-defined parameters. Multiple parameters are allowed. The logical relationship between parameters is **AND**. **NOTE** • If the value cannot be specified or does not need to be specified, end the key with an equal sign (=), for example, **frame-opt=**. • If **include** and **exclude** contain the same parameter, **exclude** precedes **include**.

Table 9-6 binary parameters

Item	Value Type	Default Value	Description
binary_out_path	String	"/data/hce-wae-auto"	Defines the path for saving the binary file that is automatically generated.
binary_file_suffix	String	"blot.auto"	Defines the suffix of the binary file name that is automatically generated.

Example Static Acceleration Configuration File

```
[binary]
binary_out_path = "/data/llvm_auto"
binary_num_threshold = 1000
binary_file_suffix = "blot.auto"

[parameter]
parameters =
   --align-blocks            # Parameter prefix -- can be added.
   frame-opt                 # If you do not need to specify the value of a parameter in the collection of user-defined parameters, you do not need to end the key with an equal sign (=).
   align-functions=1         # If the value of a parameter in the collection of user-defined parameters is specified, the value of the parameter is 1 in all parameter combinations in the generated parameter set.

[include]
align-blocks=                # If the parameter value cannot be specified, the configuration item is still ended with an equal sign (=).
[exclude]
frame-opt=none               # The parameter and its value are specified. Parameter combinations whose parameter is frame-opt and value is none are filtered out from the generated parameter set.
indirect-call-promotion=     # The parameter is specified and is of the enumeration type. All parameter combinations whose parameter is frame-opt are filtered out from the generated parameter set.
```

Dynamic Acceleration Configuration File

The following figure shows the default configuration of dynamic acceleration. You can customize a configuration file to optimize applications.

Figure 9-17 Example of default configuration for dynamic acceleration

```
# Configuration for hce-wae
# each config file should be configured for one mission, which is
# one running process in the environment
[mission]
# config the way to collect run-time data, can be defined in [perf, instrument]
log-type=instrument

# config the way to hotpatch the optimized segments, can be defined in [mode1, mode2, mode3]
# mode1 will hotpatch by dbo tools, other two types are currently not supported
hotpatch-type=mode1

# config the location where the optimized executable file to be saved in
snapshot-path=/data/hce-wae/snapshot

# let hce-wae tool know the path to the origin executable file
# should be configured when log-type is 'instrument'
origin-exe=/path/to/origin/executable/file

# config stop strategy type, three strategies can be selected:
# 1. run-times=N                  stop accelerating after optimized N times
# 2. period=N                     stop accelerating after N seconds
# 3. condition="example condition"  stop accelerating after satisfied the condition, currently not supported
[stop-strategy]
run-times=10
```

The modules in the configuration file are described as follows:

[mission]: parameters to be configured for the application.

[stop-strategy]: policy for stopping application optimization. Select one of the configurations.

Table 9-7 Configuration information

Module	Parameter	Description
[mission]	log-type	Indicates the way to collect logs during runtime. Only the instrumentation is supported
	hotpatch-type	Indicates the way to hotpatch the optimized segments. Only mode1 (ptrace) is supported.
	snapshot-path	Indicates the path for storing the optimized binary snapshot file.
	origin-exe	Indicates the location of the original application. This parameter must be specified when logs are collected using instrumentation.
[stop-strategy]	run-times	Specifies how many times the application is optimized. Optimization stops when the number is reached. Currently, an application can be optimized only once.
	period	Specifies the optimization period, in seconds. Optimization stops when the period expires. The value ranges from **1** to **600**.
	condition	Specifies the optimization condition. Optimization stops when this condition is met. This parameter is reserved.

9.2.6 Setting CPU Features

You can use the user-mode tool UcpuHconfig to set CPU features. The table below lists the configuration items and examples.

The settings will be applied and retained after UcpuHconfig commands are executed and will not be removed until the OS is restarted.

Table 9-8 How to use UcpuHconfig

Function	How to Use	Configuration Item	Value	Example	Remarks
Whether to enable L3 cache spills	Observe the traffic of each L3 cache. If the traffic is evenly distributed to all L3 caches, you are advised to disable cache spills. If some L3 caches are busy but others are idle, you are advised to enable cache spills.	spill	[on|off]	UcpuHconfig --spill on	-
Whether to enable write_unique_share for CPU caches	Observe service hotspots and data flows. If data is written into multiple cache lines continuously (for example, **memcpy**), enable this feature to optimize the bandwidth.	write_unique_share	[on|off]	UcpuHconfig --write_unique_share on	-

Function	How to Use	Configuration Item	Value	Example	Remarks
Algorithm for replacing cache lines	If Top-Down analysis shows that the backend bound is higher than 50% and L3 bound stalls are further detected, try to change the algorithm of cache line replacement and compare the effectiveness of different algorithms.	cacheline	Integer (representing a policy)	UcpuHconfig --cacheline 0	**0**: random algorithm **1**: DRRIP algorithm 2: plru **3**: random algorithm
Whether to enable L3 prefetch from DDR	If Top-Down analysis shows that the backend bound is higher than 40%, L3 bound stalls are further detected, and the DDR bandwidth of a single die is higher than 20 Gbit/s, disable prefetch to reduce the DDR bandwidth.	prefetchtgt_en	[on|off]	UcpuHconfig --prefetchtgt_en on	-

Function	How to Use	Configuration Item	Value	Example	Remarks
Whether to notify DDR during cache eviction	If Top-Down analysis detects backend bound stalls and some L3 caches are busy but others are idle, enable the notification.	reg_evict_disable	[on|off]	UcpuHconfig --reg_evict_disable on	-
Whether to notify other CPUs during cache eviction	If Top-Down analysis shows that the backend bound is higher than 40% and L3 bound stalls are further detected, enable the notification.	reg_evict_selfsnp_disable	[on|off]	UcpuHconfig --reg_evict_selfsnp_disable on	-
Whether to enable CPU prefetch from L3	If Top-Down analysis detects backend and then L3 bound stalls and the DDR bandwidth is high, disable the prefetch.	prefetch_l3	[on|off]	UcpuHconfig --prefetch_l3 on	-
Viewing	You can check special hardware settings.	show	[Function name| all]	UcpuHconfig --show spill UcpuHconfig --show all	-

9.3 Pod Bandwidth Management Tool

In hybrid service deployments, the pod bandwidth management function schedules resources based on QoS levels to improve network bandwidth utilization. HCE provides oncn-tbwm for you to manage the pod bandwidth. You

can run the **tbwmcli** commands to limit the network rate in packet sending and receiving.

Prerequisites

Before using the bandwidth management tool, ensure that the virtual NIC ifb0 is not in use, and load the ifb driver.

Constraints

- Only x86 HCE 2.0 supports **tbwmcli** commands.
- Only the root user is allowed to run the **tbwmcli** commands.
- The **tbwmcli** commands can be used to enable the QoS function for only one NIC at a time.
- After the NIC is removed and then inserted, the original QoS rules will be lost. In this case, you need to manually reconfigure the QoS function.
- **tbwmcli** commands are not supported for cgroup v2.
- Upgrading the oncn-tbwm software package does not affect the enabling status before the upgrade. Uninstalling the oncn-tbwm software package will disable the QoS function for all devices.
- Only NIC names containing digits, letters, hyphens (-), and underscores (_) can be identified.
- Bandwidth limiting may cause protocol stack memory overstock. If this happens, the transport layer protocol will perform backpressure. For UDP and other protocols that do not have backpressure mechanisms, packet loss, ENOBUFS, and inaccurate traffic limiting may occur.
- Network rate limiting for packet receiving depends on the TCP backpressure capability. In scenarios where TCP is not used, network packets have been received by the target NIC, and network rate limiting is not supported.
- The **tbwmcli**, **tc**, and NIC commands cannot be used together. You can only run the **tbwmcli** commands for rate limiting. For example, if the tc qdisc rule has been configured for a NIC, enabling the QoS function for the NIC may fail.

How to Use

1. Install the oncn-tbwm software package.
   a. Confirm that the repository is configured correctly.

      Check whether the parameters in the **/etc/yum.repos.d/hce.repo** file are configured correctly. The correct configuration is as follows:

      ```
      [base]
      name=HCE $releasever base
      baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
      enabled=1
      gpgcheck=1
      gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

      [updates]
      name=HCE $releasever updates
      baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
      enabled=1
      gpgcheck=1
      gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2
      ```



```
[debuginfo]
name=HCE $releasever debuginfo
baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
enabled=0
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
```

b. Run **yum install oncn-tbwm** to install the oncn-tbwm software package.

c. Verify the oncn-tbwm software package.

- Run the **tbwmcli -v** command. If the installation is successful, the version number will be displayed. The following is an example only.
  ```
  version: 1.0
  ```
- Check whether the following oncn-tbwm service components exist.
  ```
  /usr/bin/tbwmcli
  /usr/share/tbwmcli
  /usr/share/tbwmcli/README.md
  /usr/share/tbwmcli/bwm_prio_kern.o
  /usr/share/tbwmcli/tbwm_tc.o
  ```

2. Run the **tbwmcli** command as required.

Table 9-9 tbwmcli commands

Command	Description
tbwmcli -e ethx tbwmcli -d ethx egress	Enables or disables the packet sending QoS for a NIC. Example: Enabling the packet sending QoS for eth0 **tbwmcli -e eth0** enable eth0 egress success Example: Disabling the packet sending QoS for eth0 **tbwmcli -d eth0 egress** disable eth0 egress success
tbwmcli -i ethx online/ offline tbwmcli -d ethx ingress	Enables or disables the packet receiving QoS for a NIC. Example: Enabling the packet receiving QoS for eth0 and setting eth0 as an online NIC **tbwmcli -i eth0 online** enable eth0 ingress success, dev is online Example: Enabling the packet receiving QoS for eth0 and setting eth0 as an offline NIC **tbwmcli -i eth0 offline** enable eth0 ingress success, dev is offline **NOTE** Packet receiving QoS cannot be set for multiple offline NICs at the same time. Packet receiving QoS can only be set for one offline NIC but multiple online NICs. Example: Disabling the packet receiving QoS for eth0 **tbwmcli -d eth0 ingress** disable eth0 ingress success

Command	Description
tbwmcli -d ethx	Forcibly disables QoS for a NIC and disables ifb. Example: Forcibly disabling QoS for eth0 and disabling ifb **tbwmcli -d eth0** disable eth0 success
tbwmcli -p istats/estats	Displays the internal statistics of the packet sender and receiver. Example: Displaying the internal statistics of the packet receiver **tbwmcli -p istats** offline_target_bandwidth: 94371840online_pkts: 3626190offline_pkts: 265807online_rate: 0offline_rate: 13580offline_prio: 0 Example: Displaying the internal statistics of the packet sender **tbwmcli -p estats** offline_target_bandwidth: 94371840online_pkts: 4805452offline_pkts: 373961online_rate: 0offline_rate: 19307offline_prio: 1
tbwmcli -s path <prio> tbwmcli -p path	Sets or queries the QoS priority of a cgroup. Currently, only two QoS priorities can be set. ● **0**: Sets the cgroup online. ● **-1**: Sets the cgroup offline. Example: Setting the priority of the test_online cgroup to **0** **tbwmcli -s /sys/fs/cgroup/test_online 0** set prio success Querying the priority of the test_online cgroup **tbwmcli -p /sys/fs/cgroup/test_online** prio is 0
tbwmcli -s bandwidth <low, high> tbwmcli -p bandwidth	Sets or queries the offline bandwidth range. Example: Setting the offline bandwidth range to 30 Mbit/s to 100 Mbit/s **tbwmcli -s bandwidth 30mb,100mb** set bandwidth success Example: Querying the offline bandwidth range **tbwmcli -p bandwidth** bandwidth is 31457280(B),104857600(B)
tbwmcli -s waterline <val> tbwmcli -p waterlin	Sets or queries the online network bandwidth threshold. Example: Setting the online network bandwidth threshold to 20 Mbit/s **tbwmcli -s waterline 20mb** set waterline success Example: Querying the online network bandwidth threshold **tbwmcli -p waterline** waterline is 20971520 (B)

Command	Description
tbwmcli -p devs	Checks the enabling status of all NICs in the system. **tbwmcli -p devs** lo Egress : disabled lo Ingress : disabled eth0 Egress : disabled eth0 Ingress : enabled, it's offline ifb0 Egress : enabled
tbwmcli -c	Forcibly deletes the QoS settings of all network interfaces.
modprobe ifb numifbs=1	Loads ifb.
rmmod ifb	Uninstalls ifb.

9.4 Hardware Compatibility Test Tool

Overview

oec-hardware is a hardware compatibility test tool provided by HCE. It verifies the compatibility between servers, boards, and HCE. The verification covers only basic functions.

Compatibility Conclusion Inheritance

- **Servers**

  If the servers to be verified use the same motherboard and are in the same CPU generation, the compatibility conclusion can be inherited.

- **Boards**

  Generally, the board model is determined based on the following quadruple information:

  - **vendorID**: Chip vendor ID
  - **deviceID**: Chip model ID
  - **svID**: Board vendor ID
  - **ssID**: Board model ID

  Whether the board compatibility conclusion can be inherited is determined by the following:

  - The value of **vendorID** is different from that of **deviceID**.

    The compatibility conclusion cannot be inherited.
  - The value of **vendorID** is the same as that of **deviceID**, but different from that of **svID**.

    The compatibility conclusion cannot be inherited because the chip models are the same but the board vendors are different.
  - The values of **vendorID**, **deviceID**, and **svID** are the same.

Different boards that use the same chip from the same vendor can inherit the compatibility conclusion.

– The values of **vendorID**, **deviceID**, **svID**, and **ssID** are the same.

  Boards of the same series that use the same chip from the same vendor can inherit the compatibility conclusion. Vendors can assess the boards of the same series and use the typical board name.

Environment Requirements

- **Requirements for the server test environment**

  Table 9-10 Requirements for the server test environment

Item	Requirements
Server quantity	Two servers are required, and their service network ports can communicate with each other.
Hardware	At least one RAID controller card and one NIC (including the hardware integrated on the mainboard) are required.
Memory	Maximum memory is recommended.

- **Requirements for the board test environment**

  Table 9-11 Requirements for the board test environment

Item	Requirements
Server model	TaiShan 200 (Model 2280), 2288H V5, or equivalent servers should be used. For x86_64 servers, you can select Ice Lake, Cooper Lake, or Cascade Lake. Ice Lake is preferred.
RAID controller card	At least RAID 0 is required.
NIC/IB card	A board of the same type should be inserted into the server and the test machine, respectively. IP addresses on the same network segment are required to ensure direct communication.
FC card	The disk array needs to be connected, and at least two LUNs need to be created.

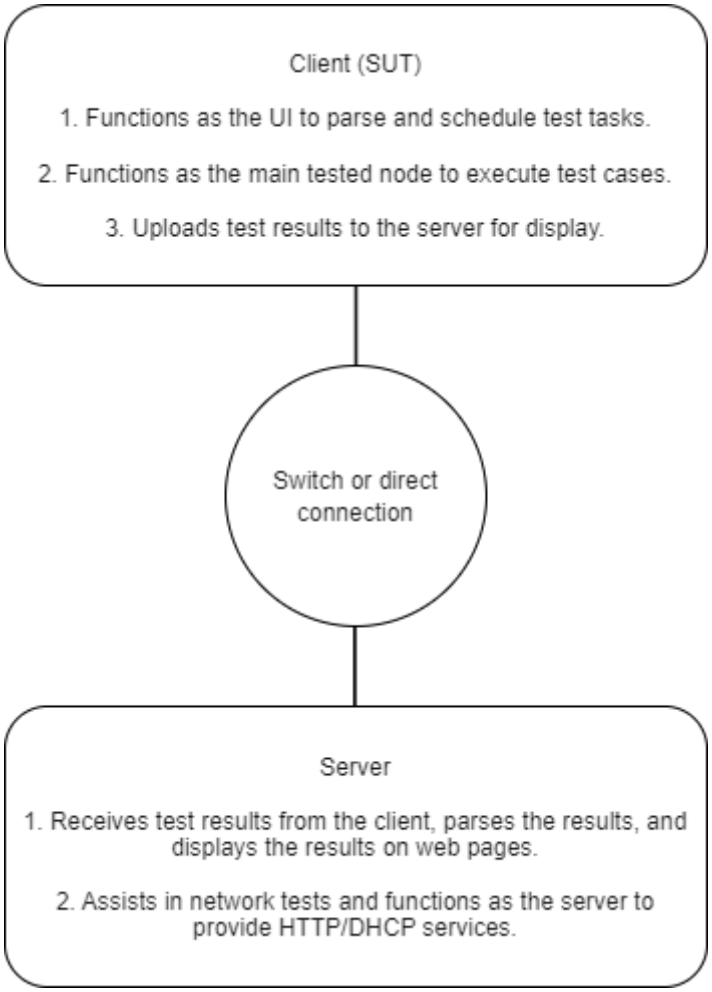
NOTE

To test an external driver, install the driver and configure the test environment in advance.

For items that need to be tested, such as GPU and keycards, install external drivers in advance. Then, use the test tool to perform the tests.

- **Operating environment networking**

Figure 9-18 Networking



Installing the Tool

oec-hardware can run in HCE 2.0 or later. For details about the supported OS versions, see the **/usr/share/oech/kernelrelease.json** file.

Step 1 Obtain the installation package.

- **Online installation**

Use an official HCE repository of the matched version, and use DNF to obtain the software package.

- **Offline installation**

1. Mount the HCE image locally and configure repositories to obtain dependencies.

2. Obtain the latest software package from the **updates** directory of the official repository of HCE.

Step 2 Install the tool.

- **Client**

  a. Install oec-hardware using DNF.
  ```
  dnf install oec-hardware
  ```

  b. Run **oech**. If the tool runs normally, the installation is successful.

- **Server**

1. Install oec-hardware-server using DNF.
   ```
   dnf install oec-hardware-server
   ```
2. Start services. oec-hardware works with Nginx to provide a web service. By default, port 80 is used. You can change the port in the Nginx configuration file. Before starting the services, ensure that their ports are not occupied.
   ```
   systemctl start oech-server.service
   systemctl start nginx.service
   ```
   Enable the services to automatically start upon server startup.
   ```
   systemctl enable oech-server.service
   systemctl enable nginx.service
   ```
3. Disable the firewall and SELinux.
   ```
   systemctl stop firewalld
   iptables -F
   setenforce 0
   ```

----End

Test Items

- **Test Introduction**
  - oec-hardware automatically restarts when kdump and watchdog tests are being performed. You are advised to perform the kdump and watchdog tests separately from other tests.
  - The keycard test depends on the specified open-source OpenSSL version. Perform the test when the environment can access the public network, or download the following content to the /**opt** directory in advance:

    openssl.tar.gz
  - The GPU test depends on some open-source tools. Perform the test when the environment can access the public network, or download the following content to the **/opt** directory in advance:

    https://github.com/NVIDIA/cuda-samples/archive/refs/heads/master.zip

    radeontop

    glmark2

    clpeak

    gpu-burn

    VulkanSamples
  - If the tests involve interaction between two nodes, disable the firewalls on the two nodes to prevent the test data from being filtered out, such as the Ethernet and DPDK tests.
    ```
    systemctl stop firewalld
    ```


  - Currently, the USB test is used to check whether the USB device can be identified. During the test, you need to manually insert or remove the USB device in different phases as prompted.
  - **/usr/share/oech/lib/config/test_config.yaml** is the configuration file template for hardware tests. Before performing FC, RAID, disk, Ethernet, and InfiniBand tests, edit the configuration file based on the actual environment and specify the hardware to be tested. For other hardware tests, you do not need to edit the configuration file.

- **Test Strategies**

  Table 9-12 Test strategies

Test	Mandatory for Servers	Mandatory for Boards
system	√	√
ACPI	√	-
clock	√	-
cpufreq	√	-
cdrom	-	-
disk	√	-
dpdk	-	-
Ethernet	√	√
FC	-	√
GPU	-	√
IPMI	√	-
InfiniBand	-	√
kABI	√	√
kdump	√	-
keycard	-	√
memory	√	-
NVMe	-	√
perf	√	-
RAID	√	√
USB	-	-
watchdog	√	-

Using the Tool

Prerequisites

- The **/usr/share/oech/kernelrelease.json** file lists all supported system versions. Run **uname -a** to check whether the current system kernel version is supported by the framework.
- By default, the framework scans all NICs. Before testing NICs, list the NICs to be tested. The test port must be connected and in the up state. You are advised not to use the service network port to perform the NIC test.
- **/usr/share/oech/lib/config/test_config.yaml** is the configuration file template for hardware tests. Before performing FC, RAID, disk, Ethernet, and InfiniBand tests, edit the configuration file based on the actual environment. For other hardware tests, you do not need to edit the configuration file. For the NIC test, if the IP address is automatically added by the tool, you need to manually delete the IP address of the server for security after the test is complete.

Procedure

Step 1 Start the test framework on the client.

```
# oech
```

Step 2 Set **Compatibility Test ID**, **Product URL**, and **Compatibility Test Server**.

Set a custom compatibility test ID (which cannot contain special characters), set **Product URL** to the product URL, and set **Compatibility Test Server** to the domain name or IP address of the server that can be directly accessed by the client and is used to display test reports and perform network tests. The default Nginx port number on the server is 80. If the port number is not changed after the server is installed, set **Compatibility Test Server** to the service IP address of the server. Otherwise, set it to the IP address and port number, for example, 172.167.145.2:90.

```
The HCE Hardware Compatibility Test Suite
Please provide your Compatibility Test ID:
Please provide your Product URL:
Please provide the Compatibility Test Server (Hostname or Ipaddr):
```

Step 3 Go to the test suite selection page. On the test case selection page, the framework automatically scans hardware and selects the test suite that can be tested in the current environment. You can enter **edit** to go to the test suite selection page.

```
These tests are recommended to complete the compatibility test:
No. Run-Now? status    Class       Device          driverName      driverVersion     chipModel       boardModel
1     yes     NotRun    acpi
2     yes     NotRun    clock
3     yes     NotRun    cpufreq
4     yes     NotRun    disk
5     yes     NotRun    ethernet    enp3s0          hinic           2.3.2.17          Hi1822          SP580
6     yes     NotRun    ethernet    enp4s0          hinic           2.3.2.17          Hi1822          SP580
7     yes     NotRun    ethernet    enp125s0f0      hns3                              HNS GE/10GE/25GE    TM210/
TM280
8     yes     NotRun    ethernet    enp125s0f1      hns3                              HNS GE/10GE/25GE    TM210/
TM280
9     yes     NotRun    raid        0000:04:00.0    megaraid_sas    07.714.04.00-rc1  SAS3408         SR150-M
10    yes     NotRun    gpu         0000:03:00.0    amdgpu                            Navi            Radeon PRO
W6800
11    yes     NotRun    ipmi
12    yes     NotRun    kabi
13    yes     NotRun    kdump
```



```
14    yes     NotRun    memory
15    yes     NotRun    perf
16    yes     NotRun    system
17    yes     NotRun    usb
18    yes     NotRun    watchdog
Ready to begin testing? (run|edit|quit)
```

Step 4 Select a test suite. The options (**all** and **none**) are used to select all and cancel all (system is a mandatory test and cannot be canceled). Enter a number to select a test suite. Only one number can be entered at a time. After you press **Enter**, **no** changes to **yes**, indicating that the test suite is selected.

```
Select tests to run:
No. Run-Now? status    Class        Device        driverName     driverVersion     chipModel
boardModel
1     no      NotRun    acpi
2     no      NotRun    clock
3     no      NotRun    cpufreq
4     no      NotRun    disk
5     yes     NotRun    ethernet     enp3s0        hinic          2.3.2.17          Hi1822            SP580
6     no      NotRun    ethernet     enp4s0        hinic          2.3.2.17          Hi1822            SP580
7     no      NotRun    ethernet     enp125s0f0     hns3                            HNS GE/10GE/25GE    TM210/
TM280
8     no      NotRun    ethernet     enp125s0f1     hns3                            HNS GE/10GE/25GE    TM210/
TM280
9     yes     NotRun    raid         0000:04:00.0   megaraid_sas   07.714.04.00-rc1   SAS3408           SR150-M
10    yes     NotRun    gpu          0000:03:00.0   amdgpu                           Navi              Radeon PRO
W6800
11    yes     NotRun    ipmi
12    yes     NotRun    kabi
13    yes     NotRun    kdump
14    yes     NotRun    memory
15    yes     NotRun    perf
16    yes     NotRun    system
17    yes     NotRun    usb
18    yes     NotRun    watchdog
Selection (<number>|all|none|quit|run):
```

Step 5 Start the test. After selecting a test suite, enter **run** to start the test.

Step 6 Upload the test results. After a test is complete, you can upload the test results to the server for display and log analysis. If the upload fails, check the network configuration and upload the test results again.

```
...
------------  Summary  ------------
ethernet-enp3s0                PASS
system                         PASS
Log saved to /usr/share/oech/logs/oech-20240928210118-TnvUJxFb50.tar succ.
Do you want to submit last result? (y|n) y
Uploading...
Successfully uploaded result to server X.X.X.X.
```

----End

Obtaining the Results

- **Viewing Test Logs**

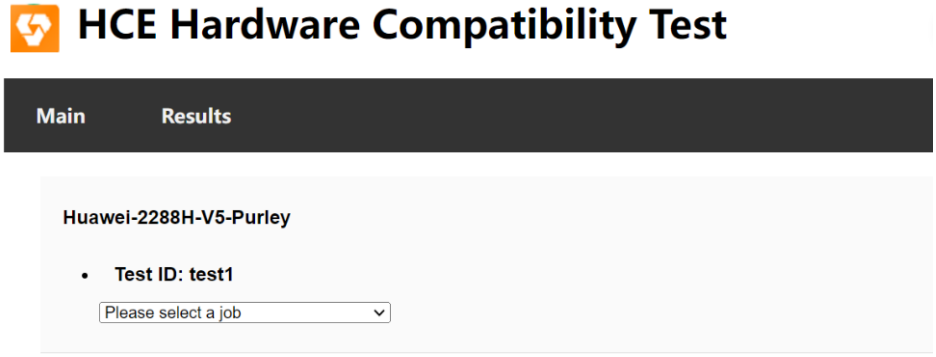
  After the test is complete, the test logs are saved in the **/usr/share/oech/logs/** directory. You can export and decompress the test logs to view them.

- **Viewing the Test Report Using a Browser**

  View the test report using a browser. You need to configure the server in advance receive the test results.

  a. Open the browser, enter the server IP address, click **Results**, and find the corresponding test IDs.

Figure 9-19 Viewing the test report using a browser



b. View the detailed test results on each page, including the environment information and execution results.

- **Summary**: View all test results.
- **Devices**: View information about all hardware devices.
- **Runtime**: View the test runtime and general task execution logs.
- **Attachment**: Download the test log attachment.

9.5 A-Tune

9.5.1 About A-Tune

An operating system (OS) is base software that connects applications and hardware. It is critical for users to adjust OS and application configurations and make full use of software and hardware capabilities to achieve better service performance. However, numerous workload types and varied applications run on an OS, and the requirements on resources are different. Currently, the application environment composed of hardware and software involves more than 7,000 configuration objects. When the service complexity and optimization objects increase, the time required for optimization increases exponentially. As a result, optimization efficiency decreases sharply. Optimization becomes complex and brings great challenges to users.

As infrastructure software, an OS provides a large number of software and hardware management capabilities. The capabilities required vary in different scenarios and need to be enabled or disabled as needed. A combination of capabilities will maximize the optimal performance of applications.

In addition, the actual services embrace hundreds and thousands of scenarios, and each scenario involves a wide variety of hardware configurations for compute, network, and storage. The lab cannot list all applications, service scenarios, or hardware combinations.

To address the preceding challenges, HCE integrates A-Tune launched by openEuler.

A-Tune is an AI-powered engine that optimizes the system to ensure that services can run at optimal performance. It uses AI technologies to precisely profile service

scenarios, discover and infer service characteristics, so as to make intelligent decisions, match the optimal combination of system parameter settings, and give recommendations.

9.5.2 Installation and Deployment

Installing A-Tune

This section describes how to install A-Tune.

Installation modes

A-Tune can be installed in single-node, distributed, or cluster mode.

- Single-node

  The client and server are installed on the same node.
- Distributed

  The client and server are installed on different nodes.
- Cluster

  A cluster consists of one client and more than one server.

Installation operations

To install A-Tune, perform the following steps:

1. Install the A-Tune server.
   ```
   # yum install atune -y
   # yum install atune-engine -y
   ```
2. In distributed mode, you also need to install the A-Tune client.
   ```
   # yum install atune-client -y
   ```
3. Check whether the installation is successful. If the following information is displayed, A-Tune is installed successfully:
   ```
   # rpm -qa | grep atune
   atune-client-xxx
   atune-db-xxx
   atune-xxx
   atune-engine-xxx
   ```

Deploying A-Tune

This section describes how to deploy A-Tune.

Configuration

The A-Tune configuration file **/etc/atuned/atuned.cnf** contains the following items:

- A-Tune startup (the values can be changed as needed)
  - **protocol**: protocol used by the system service gRPC. The value can be **unix** or **tcp**. **unix** indicates local socket communications and **tcp** indicates communications through socket listening ports. The default value is **unix**.
  - **address**: listening address of gRPC. The default value is **unix socket**. If A-Tune is deployed in distributed mode, change the value to the listening IP address.

  - **port**: listening port of gRPC. The value must be an idle port ranging from **0** to **65535**. If **protocol** is set to **unix**, this parameter does not need to be configured.
  - **connect**: IP address list of the nodes where A-Tune is deployed in a cluster. Separate the IP addresses by commas (,).
  - **rest_host**: listening address of the system service REST. The default value is **localhost**.
  - **rest_port**: listening port of the system service REST. The value must be an idle port ranging from **0** to **65535**. The default value is **8383**.
  - **engine_host**: address for connecting to the system service A-Tune engine.
  - **engine_port**: port for connecting to the system service A-Tune engine.
  - **sample_num**: number of samples collected by the system for data analysis. The default value is **20**.
  - **interval**: interval for the system to collect samples. The default value is 5s.
  - **grpc_tls**: whether to enable SSL/TLS certificate verification for gRPC. By default, it is disabled. If **grpc_tls** is enabled, the following environment variables need to be set before **atune-adm** is run to communicate with the server:
    - export ATUNE_TLS=yes
    - export ATUNED_CACERT=*<CA-certificate-path-of-the-client>*
    - export ATUNED_CLIENTCERT=*<client-certificate-path>*
    - export ATUNED_CLIENTKEY=*<client-key-path>*
    - export ATUNED_SERVERCN=server
  - **tlsservercafile**: CA certificate path of the gRPC server.
  - **tlsservercertfile**: certificate path of the gRPC server.
  - **tlsserverkeyfile**: key path of the gRPC server.
  - **rest_tls**: whether to enable SSL/TLS certificate verification for REST. By default, it is enabled.
  - **tlsrestcacertfile**: CA certificate path of the REST server.
  - **tlsrestservercertfile**: certificate path of the REST server.
  - **tlsrestserverkeyfile**: key path of the REST server.
  - **engine_tls**: whether to enable SSL/TLS certificate verification for A-Tune engine. By default, it is enabled.
  - **tlsenginecacertfile**: CA certificate path of the A-Tune engine client.
  - **tlsengineclientcertfile**: certificate path of the A-Tune engine client.
  - **tlsengineclientkeyfile**: key path of the A-Tune engine client.
- System information

  These are parameters required for system optimization. Configure these parameters based on your system requirements.
  - **disk**: disk whose information is to be collected for analysis or who is to be optimized.

  - **network**: NIC whose information is to be collected for analysis or who is to be optimized.
  - **user**: username used for **ulimit** optimization. Currently, only **root** is available.
- Log information

  You can change the log level as needed. The default value is **info**. Logs are recorded in the **/var/log/messages** file.
- Monitoring information

  These are system hardware information collected by default during system startup.
- Tuning information

  These parameters are used for offline tuning.
  - **noise**: estimated value of Gaussian noise.
  - **sel_feature**: whether to output the ranking of offline tuning parameter importance. It is disabled by default.

Configuration example

```
################################### server ################################
# atuned config
[server]
# the protocol grpc server running on
# ranges: unix or tcp
protocol = unix

# the address that the grpc server to bind to
# default is unix socket /var/run/atuned/atuned.sock
# ranges: /var/run/atuned/atuned.sock or ip address
address = /var/run/atuned/atuned.sock

# the atune nodes in cluster mode, separated by commas
# it is valid when protocol is tcp
# connect = ip01,ip02,ip03

# the atuned grpc listening port
# the port can be set between 0 to 65535 which not be used
# port = 60001

# the rest service listening port, default is 8383
# the port can be set between 0 to 65535 which not be used
rest_host = localhost
rest_port = 8383

# the tuning optimizer host and port, start by engine.service
# if engine_host is same as rest_host, two ports cannot be same
# the port can be set between 0 to 65535 which not be used
engine_host = localhost
engine_port = 3838

# when run analysis command, the numbers of collected data.
# default is 20
sample_num = 20

# interval for collecting data, default is 5s
interval = 5

# enable gRPC authentication SSL/TLS
# default is false
# grpc_tls = false
# tlsservercafile = /etc/atuned/grpc_certs/ca.crt
# tlsservercertfile = /etc/atuned/grpc_certs/server.crt
# tlsserverkeyfile = /etc/atuned/grpc_certs/server.key
```



```
# enable rest server authentication SSL/TLS
# default is true
rest_tls = true
tlsrestcacertfile = /etc/atuned/rest_certs/ca.crt
tlsrestservercertfile = /etc/atuned/rest_certs/server.crt
tlsrestserverkeyfile = /etc/atuned/rest_certs/server.key

# enable engine server authentication SSL/TLS
# default is true  engine_tls = true
tlsenginecacertfile = /etc/atuned/engine_certs/ca.crt
tlsengineclientcertfile = /etc/atuned/engine_certs/client.crt
tlsengineclientkeyfile = /etc/atuned/engine_certs/client.key    #

################################### log ##############################
[log]
# either "debug", "info", "warn", "error", "critical", default is "info"
level = info

################################### monitor ##############################
[monitor]
# with the module and format of the MPI, the format is {module}_{purpose}
# the module is Either "mem", "net", "cpu", "storage"
# the purpose is "topo"
module = mem_topo, cpu_topo

################################### system ##############################
# you can add arbitrary key-value here, just like key = value
# you can use the key in the profile
[system]
# the disk to be analysis
disk = sda

# the network to be analysis
network = enp189s0f0
user = root

################################### tuning ##############################
# tuning configs
[tuning]
noise = 0.000000001
sel_feature = false
```

The configuration file **/etc/atuned/engine.cnf** of A-Tune engine contains the following items:

- A-Tune engine startup (the values can be changed as needed)
  - **engine_host**: listening address of the system service A-Tune engine. The default value is **localhost**.
  - **engine_port**: listening port of the system service A-Tune engine. The value must be an idle port ranging from **0** to **65535**. The default value is **3838**.
  - **engine_tls**: whether to enable SSL/TLS certificate verification for A-Tune engine. By default, it is enabled.
  - **tlsenginecacertfile**: CA certificate path of the A-Tune engine server.
  - **tlsengineservercertfile**: certificate path of the A-Tune engine server.
  - **tlsengineserverkeyfile**: key path of the A-Tune engine server.
- Log information

  You can change the log level as needed. The default value is **info**. Logs are recorded in the **/var/log/messages** file.

Configuration example


```
 ################################### engine ##################################
[server]
# the tuning optimizer host and port, start by engine.service
# if engine_host is same as rest_host, two ports cannot be same
# the port can be set between 0 to 65535 which not be used
engine_host = localhost
engine_port = 3838

# enable engine server authentication SSL/TLS
# default is true
engine_tls = true
tlsenginecacertfile = /etc/atuned/engine_certs/ca.crt
tlsengineservercertfile = /etc/atuned/engine_certs/server.crt
tlsengineserverkeyfile = /etc/atuned/engine_certs/server.key

################################### log ##################################
[log]
# either "debug", "info", "warn", "error", "critical", default is "info"
level = info
```

Starting A-Tune

After A-Tune is installed, configure and start it.

- Configuring the A-Tune service: Modify the NIC and disk information in the **atuned.cnf** configuration file.

  Set **network** in **/etc/atuned/atuned.cnf** to the NIC whose information needs to be collected or the NIC to be optimized. You can run the following command to query the NIC:

  ```
  ip addr
  ```

  Set **disk** in **/etc/atuned/atuned.cnf** to the disk whose information needs to be collected or the disk to be optimized. You can run the following command to query the disk:

  ```
  fdisk -l | grep dev
  ```

- Certificate: The A-Tune engine and client use the gRPC protocol for communication. To ensure system security, you need to configure certificates. For information security purposes, A-Tune does not provide the method for generating a certificate. You need to configure system certificates by yourself. If security is not considered, set **rest_tls** and **engine_tls** in **/etc/atuned/atuned.cnf** to **false**, and **engine_tls** in **/etc/atuned/engine.cnf** to **false**. A-Tune is not responsible for any consequences caused by the lack of security certificates.
- Start the atuned service.

  ```
  # systemctl start atuned
  ```

- Check the atuned status.

  ```
  # systemctl status atuned
  ```

  If the following information is displayed, the service is started successfully.

  ```
  ● atuned.service - A-Tune Daemon
     Loaded: loaded (/usr/lib/systemd/system/atuned.service; disabled; vendor preset: disabled)
     Active: active (running) since Sat 2019-12-21 19:28:47 CST; 7s ago
   Main PID: 94790 (atuned)
      Tasks: 19 (limit: 104856)
     Memory: 132.2M
     CGroup: /system.slice/atuned.service
             ├─94790 /usr/bin/atuned
             ├─94797 python3 /usr/libexec/atuned/analysis/app.py /etc/atuned/atuned.cnf
             └─94801 /usr/bin/python3 -c from multiprocessing.semaphore_tracker import main;main(3)
  ```


Starting A-Tune engine

To use AI functions, start the A-Tune engine service.

- Start the atune-engine service.
  ```
  # systemctl start atune-engine
  ```
- Check the atune-engine status.
  ```
  # systemctl status atune-engine
  ```
  If the following information is displayed, the service is started successfully.

```
[root@localhost misc]# systemctl status atune-engine
● atune-engine.service - A-Tune AI service
   Loaded: loaded (/usr/lib/systemd/system/atune-engine.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2020-09-05 04:24:12 EDT; 33min ago
Main PID: 476292 (python3)
    Tasks: 3
   Memory: 102.5M
   CGroup: /system.slice/atune-engine.service
           ├─476292 /usr/bin/python3 /usr/libexec/atuned/analysis/app_engine.py /etc/atuned/atuned.cnf
           └─476312 /usr/bin/python3 -c from multiprocessing.semaphore_tracker import main;main(3)
```

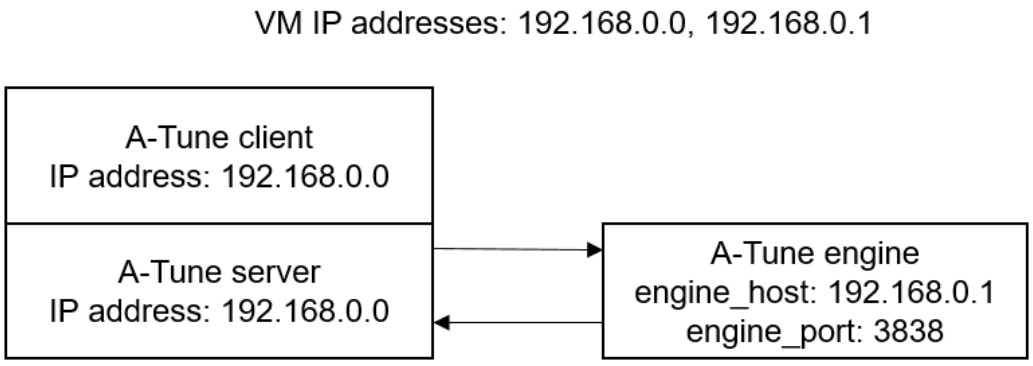
Distributed Deployment

Purpose

Distributed deployment allows A-Tune to be deployed on demand and run in a distributed architecture. Three components can be deployed separately. The lightweight deployment has little impacts on services and prevents too much software dependency, reducing the system load.

There is more than one method of distributed deployment. This section only describes the common one. That is, deploy the server and client on the same node and the engine on another node. For other deployment methods, contact technical support.

Deployment relationships



Configuration files

In distributed deployment, you need to write the IP address and port of the engine into configuration files so that other components can access the engine over the IP address.

1. Modify the **/etc/atuned/atuned.cnf** file on the A-Tune server node.
   - Change the values of **engine_host** and **engine_port** in line 34 to the IP address and port of the engine node. As shown in the figure above, change the values to **engine_host = 192.168.0.1** and **engine_port = 3838**.

  - Change the values of **rest_tls** and **engine_tls** in lines 49 and 55 to **false**. Otherwise, you will be required to apply for and configure certificates. In a testing environment, you do not need to configure the SSL certificate. In a production environment, you need to configure it to prevent security risks.

2. Modify the **/etc/atuned/engine.cnf** file on the engine node.
   - Change the values of **engine_host** and **engine_port** in lines 17 and 18 to the IP address and port of the engine node. As shown in the figure above, change the values to **engine_host = 192.168.0.1** and **engine_port = 3838**.
   - Change the value of **engine_tls** in line 22 to **false**.
3. Restart A-Tune and the engine to make the configurations take effect.
   - Run **systemctl restart atuned** on the A-Tune server node.
   - Run **systemctl restart atune-engine** on the engine node.
4. (Optional) Run a **tuning** command in the **A-Tune/examples/tuning/compress** directory to check whether the distributed deployment is successful.
   a. Perform a preprocess by referring to **A-Tune/examples/tuning/compress/README**.
   b. Run **atune-adm tuning --project compress --detail compress_client.yaml**.

Precautions

1. Details about how to configure an authentication certificate is not provided here. If necessary, you can set **rest_tls/engine_tls** in **atuned.cnf** and **engine.cnf** to **false**.
2. After modifying the configuration files, restart A-Tune and the engine. Otherwise, the modifications will not be valid.
3. Do not enable the network proxy when using the A-Tune service.
4. In the **atuned.cnf** file, set **disk** and **network** in **[system]** to the actual disk and network interface.

Example

atuned.cnf

```
#...

# the tuning optimizer host and port, start by engine.service
# if engine_host is same as rest_host, two ports cannot be same
# the port can be set between 0 to 65535 which not be used
engine_host = 192.168.0.1
engine_port = 3838

#...
```

engine.cnf

```
[server]
# the tuning optimizer host and port, start by engine.service
# if engine_host is same as rest_host, two ports cannot be same
# the port can be set between 0 to 65535 which not be used
engine_host = 192.168.0.1
engine_port = 3838
```


Cluster Deployment

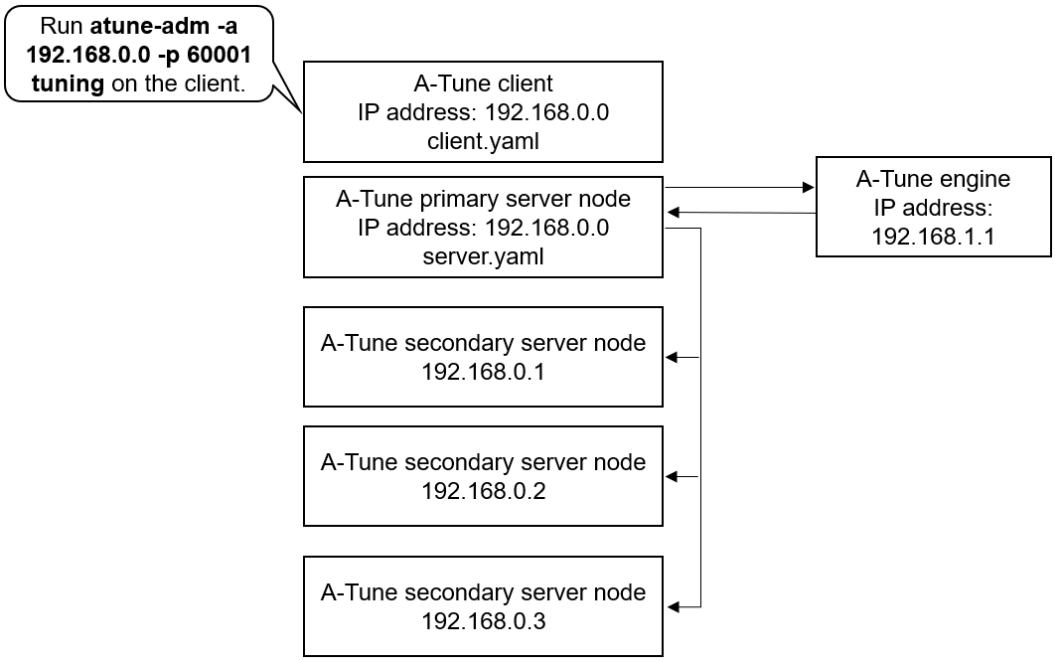
Purpose

In an A-Tune cluster, parameters of multiple nodes can be dynamically tuned at the same time. This avoids repeated tunning on each node and improves tunning efficiency.

Method

An A-Tune cluster consists of one primary node and multiple secondary nodes. The client and server are deployed on the primary node to receive commands and interact with the engine. Other nodes receive instructions from the primary node to tune their parameters.

Deployment relationships



As shown in the figure above, the A-Tune server and client are deployed on the same node (IP address: 192.168.0.0). Project files are stored on this node and do not need to be stored on other nodes. The primary node and secondary nodes communicate with each other through TCP. You need to modify the configuration file for each node.

Modifying atuned.cnf

1. Set **protocol** to **tcp**.
2. Set **address** to the IP address of the current node.
3. Set **connect** to the IP addresses of all nodes. The first is the IP address of the primary node and the others are IP addresses of secondary nodes. Use commas (,) to separate the IP addresses.
4. During debugging, you can set **rest_tls** and **engine_tls** to **false**.
5. Repeat the steps to modify the **atuned.cnf** file of all primary and secondary nodes.

Precautions

1. Set **engine_host** and **engine_port** in **engine.cnf** to the same values of **engine_host** and **engine_port** in **atuned.cnf**.
2. Details about how to configure an authentication certificate is not provided here. If necessary, you can set **rest_tls/engine_tls** in **atuned.cnf** and **engine.cnf** to **false**.
3. After modifying the configuration files, restart A-Tune and the engine. Otherwise, the modifications will not be valid.
4. Do not enable the network proxy when using the A-Tune service.

Example

atuned.cnf

```
#...

[server]
# the protocol grpc server running on
# ranges: unix or tcp
protocol = tcp

# the address that the grpc server to bind to
# default is unix socket /var/run/atuned/atuned.sock
# ranges: /var/run/atuned/atuned.sock or ip address
address = 192.168.0.0

# the atune nodes in cluster mode, separated by commas
# it is valid when protocol is tcp
connect = 192.168.0.0,192.168.0.1,192.168.0.2,192.168.0.3

# the atuned grpc listening port
# the port can be set between 0 to 65535 which not be used
port = 60001

# the rest service listening port, default is 8383
# the port can be set between 0 to 65535 which not be used
rest_host = localhost
rest_port = 8383

# the tuning optimizer host and port, start by engine.service
# if engine_host is same as rest_host, two ports cannot be same
# the port can be set between 0 to 65535 which not be used
engine_host = 192.168.1.1
engine_port = 3838

#...
```

engine.cnf

```
[server]
# the tuning optimizer host and port, start by engine.service
# if engine_host is same as rest_host, two ports cannot be same
# the port can be set between 0 to 65535 which not be used
engine_host = 192.168.1.1
engine_port = 3838
```

Note: Configure **engine.cnf** by referring the configuration for distributed deployment.

9.5.3 How to Use

You can use A-Tune through the CLI client **atune-adm**. This section describes how to use the A-Tune client.

Overview

- To use A-Tune, you must have the **root** permissions.
- You can run **atune-adm help/--help/-h** to query **atune-adm** commands.
- **define**, **update**, **undefine**, **collection**, **train**, and **upgrade** cannot be executed remotely.
- In the command syntax, a parameter enclosed within square brackets ([]) is optional, and a parameter enclosed within angle brackets (<>) is mandatory.

Querying Workload Types

list

Description

This command is used to query the profiles supported by the system and the profiles in active state.

Command format

atune-adm list

Example

```
# atune-adm list
Support profiles:
+-------------------------------------------------+-----------+
| ProfileName                                     | Active    |
+=================================================+===========+
| arm-native-android-container-robox              | false     |
+-------------------------------------------------+-----------+
| basic-test-suite-euleros-baseline-fio           | false     |
+-------------------------------------------------+-----------+
| basic-test-suite-euleros-baseline-lmbench       | false     |
+-------------------------------------------------+-----------+
| basic-test-suite-euleros-baseline-netperf       | false     |
+-------------------------------------------------+-----------+
| basic-test-suite-euleros-baseline-stream        | false     |
+-------------------------------------------------+-----------+
| basic-test-suite-euleros-baseline-unixbench     | false     |
+-------------------------------------------------+-----------+
| basic-test-suite-speccpu-speccpu2006            | false     |
+-------------------------------------------------+-----------+
| basic-test-suite-specjbb-specjbb2015            | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-hdfs-dfsio-hdd                  | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-hdfs-dfsio-ssd                  | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-spark-bayesian                  | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-spark-kmeans                    | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-spark-sql1                      | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-spark-sql10                     | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-spark-sql2                      | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-spark-sql3                      | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-spark-sql4                      | false     |
+-------------------------------------------------+-----------+
| big-data-hadoop-spark-sql5                      | false     |
+-------------------------------------------------+-----------+
```



```
| big-data-hadoop-spark-sql6                    | false    |
+-----------------------------------------------+----------+
| big-data-hadoop-spark-sql7                    | false    |
+-----------------------------------------------+----------+
| big-data-hadoop-spark-sql8                    | false    |
+-----------------------------------------------+----------+
| big-data-hadoop-spark-sql9                    | false    |
+-----------------------------------------------+----------+
| big-data-hadoop-spark-tersort                 | false    |
+-----------------------------------------------+----------+
| big-data-hadoop-spark-wordcount               | false    |
+-----------------------------------------------+----------+
| cloud-compute-kvm-host                        | false    |
+-----------------------------------------------+----------+
| database-mariadb-2p-tpcc-c3                   | false    |
+-----------------------------------------------+----------+
| database-mariadb-4p-tpcc-c3                   | false    |
+-----------------------------------------------+----------+
| database-mongodb-2p-sysbench                  | false    |
+-----------------------------------------------+----------+
| database-mysql-2p-sysbench-hdd                | false    |
+-----------------------------------------------+----------+
| database-mysql-2p-sysbench-ssd                | false    |
+-----------------------------------------------+----------+
| database-postgresql-2p-sysbench-hdd           | false    |
+-----------------------------------------------+----------+
| database-postgresql-2p-sysbench-ssd           | false    |
+-----------------------------------------------+----------+
| default-default                               | false    |
+-----------------------------------------------+----------+
| docker-mariadb-2p-tpcc-c3                     | false    |
+-----------------------------------------------+----------+
| docker-mariadb-4p-tpcc-c3                     | false    |
+-----------------------------------------------+----------+
| hpc-gatk4-human-genome                        | false    |
+-----------------------------------------------+----------+
| in-memory-database-redis-redis-benchmark      | false    |
+-----------------------------------------------+----------+
| middleware-dubbo-dubbo-benchmark              | false    |
+-----------------------------------------------+----------+
| storage-ceph-vdbench-hdd                      | false    |
+-----------------------------------------------+----------+
| storage-ceph-vdbench-ssd                      | false    |
+-----------------------------------------------+----------+
| virtualization-consumer-cloud-olc             | false    |
+-----------------------------------------------+----------+
| virtualization-mariadb-2p-tpcc-c3             | false    |
+-----------------------------------------------+----------+
| virtualization-mariadb-4p-tpcc-c3             | false    |
+-----------------------------------------------+----------+
| web-apache-traffic-server-spirent-pingpo      | false    |
+-----------------------------------------------+----------+
| web-nginx-http-long-connection                | true     |
+-----------------------------------------------+----------+
| web-nginx-https-short-connection              | false    |
+-----------------------------------------------+----------+
```

If the value of **Active** is **true**, the profile is activated. In this example, the activated profile is **web-nginx-http-long-connection**.

Analyzing Workload Types and Performing Automated Tuning

analysis

Description

This command is used to collect real-time statistics of the system to identify workload types and then perform automated tuning.

Command format

atune-adm analysis [OPTIONS]

Parameters

- OPTIONS

Parameter	Description
--model, -m	New model trained by a user
--characterization, -c	Application identification using the default model, without automated tuning
--times value, -t value	Data collection duration
--script value, -s value	File to be executed

Example

- Use the default model for application identification.
  ```
  # atune-adm analysis --characterization
  ```
- Use the default model for application identification and perform automated tuning.
  ```
  # atune-adm analysis
  ```
- Use a user-trained model for application identification.
  ```
  # atune-adm analysis --model /usr/libexec/atuned/analysis/models/new-model.m
  ```

Custom Models

A-Tune allows users to define a new model for learning. To define a new model, perform the following steps:

1. Run **define** to define a profile for a new application.
2. Run **collection** to collect system data of the application.
3. Run **train** to train a model.

define

Description

This command is used to add a user-defined application scenario and tuning items to a profile.

Command format

atune-adm define <service_type> <application_name> <scenario_name> <profile_path>

Example

Add a profile. Set **service_type** to **test_service**, **application_name** to **test_app**, **scenario_name** to **test_scenario**, and tuning item configuration file to **example.conf**.

```
# atune-adm define test_service test_app test_scenario ./example.conf
```

You can write the **example.conf** file as below (the tuning items are optional and for reference only). Alternatively, run **atune-adm info** to see how existing profiles are written.

```
 [main]
# list its parent profile
[kernel_config]
# to change the kernel config
[bios]
# to change the bios config
[bootloader.grub2]
# to change the grub2 config
[sysfs]
# to change the /sys/* config
[systemctl]
# to change the system service status
[sysctl]
# to change the /proc/sys/* config
[script]
# the script extension of cpi
[ulimit]
# to change the resources limit of user
[schedule_policy]
# to change the schedule policy
[check]
# check the environment
[tip]
# the recommended optimization, which should be performed manunaly
```

collection

Description

This command is used to collect resource usages of the entire system and the OS status when services are running and save the collected information into a CSV file as the input dataset for model training.

Note:

- This command depends on **perf**, **mpstat**, **vmstat**, **iostat**, and **sar**.
- Currently, only Kunpeng 920 is supported. You can run **dmidecode -t processor** to check the CPU model.

Command format

atune-adm collection <OPTIONS>

Parameters

- OPTIONS

Parameter	Description
--filename, -f	Name of the generated CSV file used for training (format: *Name-Timestamp*.**csv**)

Parameter	Description
--output_path, -o	Absolute path for storing the generated CSV file
--disk, -b	Disks used for running services, for example, **/dev/sda**
--network, -n	Network interface used for running services, for example, **eth0**
--app_type, -t	Application type of a service, which is used as a label during training
--duration, -d	Data collection duration when services are running, in seconds (default: 1,200s)
--interval, -i	Interval for collecting data, in seconds (default: 5s)

Example

```
# atune-adm collection --filename name --interval 5 --duration 1200 --output_path /home/data --disk sda --network eth0 --app_type test_service-test_app-test_scenario
```

Note:

In the example, data is collected every 5s, and the collection lasts for 1,200s. The collected data is stored in the **name** file in the **/home/data** directory. The service application type is specified by **atune-adm define** and the value is **test_service-test_app-test_scenario**.

train

Description

This command is used to use the collected data to train a model. Use data of at least two application types for training. Otherwise, an error may occur during training.

Command format

atune-adm train <OPTIONS>

Parameters

- OPTIONS

Parameter	Description
--data_path, -d	Directory for storing the CSV file required for model training
--output_file, -o	Trained model

Example

Use the CSV file in the **data** directory as the training input and save the generated model **new-model.m** in the **model** directory.


```
# atune-adm train --data_path /home/data --output_file /usr/libexec/atuned/analysis/models/new-model.m
```

undefine

Description

This command is used to delete a user-defined profile.

Command format

atune-adm undefine <profile>

Example

Delete a user-defined profile.

```
# atune-adm undefine test_service-test_app-test_scenario
```

info

Description

This command is used to view information about a profile.

Command format

atune-adm info <profile>

Example

View the **web-nginx-http-long-connection** profile.

```
# atune-adm info web-nginx-http-long-connection
*** web-nginx-http-long-connection:
#
# nginx http long connection A-Tune configuration
#
[main]
include = default-default

[kernel_config]
#TODO CONFIG

[bios]
#TODO CONFIG

[bootloader.grub2]
iommu.passthrough = 1

[sysfs]
#TODO CONFIG

[systemctl]
sysmonitor = stop
irqbalance = stop

[sysctl]
fs.file-max = 6553600
fs.suid_dumpable = 1
fs.aio-max-nr = 1048576
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.ip_local_port_range = 1024     65500
net.ipv4.tcp_max_tw_buckets = 5000
```



```
net.core.somaxconn = 65535
net.core.netdev_max_backlog = 262144
net.ipv4.tcp_max_orphans = 262144
net.ipv4.tcp_max_syn_backlog = 262144
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_synack_retries = 1
net.ipv4.tcp_syn_retries = 1
net.ipv4.tcp_fin_timeout = 1
net.ipv4.tcp_keepalive_time = 60
net.ipv4.tcp_mem =  362619      483495   725238
net.ipv4.tcp_rmem = 4096         87380   6291456
net.ipv4.tcp_wmem = 4096         16384   4194304
net.core.wmem_default = 8388608
net.core.rmem_default = 8388608
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216

[script]
prefetch = off ethtool =  -X

{network} hfunc toeplitz

[ulimit]
{user}.hard.nofile = 102400
{user}.soft.nofile = 102400

[schedule_policy]
#TODO CONFIG

[check]
#TODO CONFIG

[tip] SELinux provides extra control and security features to linux kernel. Disabling SELinux will improve the performance but may cause security risks. = kernel disable the nginx log = application
```

Updating a Profile

You can update a profile as needed.

update

Description

This command is used to update the tuning items in an existing profile to those in the **new.conf** file.

Command format

atune-adm update <profile> <profile_path>

Example

Update the tuning items in the **test_service-test_app-test_scenario** profile to those in the **new.conf** file.

```
# atune-adm update test_service-test_app-test_scenario ./new.conf
```

Activating a Profile

profile

Description

This command is used to manually activate a profile.

Command format

atune-adm profile <profile>

Description

For the profile name, see the query result of the **list** command.

Example

Activate the **web-nginx-http-long-connection** profile.

```
# atune-adm profile web-nginx-http-long-connection
```

Rolling Back a Profile

rollback

Description

This command is used to roll back a profile to the initial settings.

Command format

atune-adm rollback

Example

```
# atune-adm rollback
```

Updating the Database

upgrade

Description

This command is used to update the system database.

Command format

atune-adm upgrade <DB_FILE>

Description

- DB_FILE
  New database file path

Example

Update the database to **new_sqlite.db**.

```
# atune-adm upgrade ./new_sqlite.db
```

Querying System Information

check

Description

This command is used to query system information such as CPU, BIOS, OS, and NICs.

Command format

atune-adm check

Example

```
# atune-adm check
cpu information:
   cpu:0   version: Kunpeng 920-6426  speed: 2600000000 HZ   cores: 64
   cpu:1   version: Kunpeng 920-6426  speed: 2600000000 HZ   cores: 64
system information:
   DMIBIOSVersion: 20.47
   OSRelease: 5.10.0-182.0.0.95.r2055_140.hce2.aarch64
network information:
   name: eth0          product: HNS GE/10GE/25GE RDMA Network Controller
   name: eth1          product: HNS GE/10GE/25GE Network Controller
   name: eth2          product: HNS GE/10GE/25GE RDMA Network Controller
   name: eth3          product: HNS GE/10GE/25GE Network Controller
   name: eth4          product: HNS GE/10GE/25GE RDMA Network Controller
   name: eth5          product: HNS GE/10GE/25GE Network Controller
   name: eth6          product: HNS GE/10GE/25GE RDMA Network Controller
   name: eth7          product: HNS GE/10GE/25GE Network Controller
```

Performing Automated Parameter Tuning

A-Tune automatically searches for the optimal settings. This saves time and efforts by eliminating the need of manual adjustments and performance evaluations.

tuning

Description

This command is used to specify a project file to dynamically search for the optimal settings for the current environment.

Command format

Before running the command, ensure that the following conditions are met:

1. The YAML configuration file of the A-Tune server has been edited and stored in the ****/etc/atuned/tuning/**** directory of the server.
2. The YAML configuration file of the A-Tune client has been edited and stored in any directory of the client.

atune-adm tuning [OPTIONS] <PROJECT_YAML>

Parameters

- OPTIONS

Parameter	Description
--restore, -r	Initial settings before tuning
--project, -p	Name of the project to be restored in the YAML file
--restart, -c	Tuning based on historical tuning results
--detail, -d	Details about the tuning process

-p must be followed by a specific project name. The YAML file of the project must also be specified.

- PROJECT_YAML: YAML configuration file on the client

Configuration

Table 9-13 YAML configuration file on the server

Parameter	Description	Type	Value
project	Project name.	String	-
startworkload	Script for starting the service to be tuned.	String	-
stopworkload	Script for stopping the service to be tuned.	String	-
maxiterations	Maximum number of tuning iterations, which is used to limit the number of iterations on the client. Generally, more iterations can lead to better tuning results, but they also require more time. You can configure it based on service requirements.	Integer	>10
object	Parameters to be adjusted and related information. For details, see **Table 9-14**.	-	-

Table 9-14 object parameters

Parameter	Description	Type	Value
name	Name of the parameter to be tuned.	String	-
desc	Description of the parameter to be tuned.	String	-

Parameter	Description	Type	Value
get	Script for querying the parameter value.	-	-
set	Script for setting the parameter value.	-	-
needrestart	Whether to restart the service for the parameter to take effect.	Enumerated	"true", "false"
type	Parameter type, which can be **discrete** or **continuous**.	Enumerated	"discrete", "continuous"
dtype	This parameter needs to be configured only when **type** is set to **discrete**. The value can be **int**, **float**, and **string**.	Enumerated	"int","float","string"
scope	Parameter value range. This parameter is valid only when **type** is set to **discrete** and **dtype** is set to **int** or **float**, or when **type** is set to **continuous**.	Integer/Float	User-defined. It specifies the valid value range of the tunned parameter.
step	Parameter value step, which is used when **dtype** is set to **int** or **float**.	Integer/Float	User-defined
items	Enumerated parameter values beyond the range defined by **scope**. It is used when **dtype** is set to **int** or **float**.	Integer/Float	User-defined. It specifies the valid value range of the tunned parameter.

Parameter	Description	Type	Value
options	Enumerated value range of the tunned parameter. It is used when **dtype** is set to **string**.	String	User-defined. It specifies the valid value range of the tunned parameter.

Table 9-15 YAML configuration file on the client

Parameter	Description	Type	Value
project	Project name, which must be the same as that in the configuration file of the server.	String	-
engine	Tunning algorithm	String	"random", "forest", "gbrt", "bayes", "extraTrees"
iterations	Tuning iterations.	Integer	>=10
random_starts	Number of random iterations.	Integer	<iterations
feature_filter_engine	(Optional) Parameter search algorithm. It is used to select important parameters.	String	"lhs"
feature_filter_cycle	Number of parameter search rounds. It is used to select important parameters and must be used together with **feature_filter_engine**.	Integer	-

Parameter	Description	Type	Value
feature_filter_iters	Number of iterations for each cycle of parameter search. It is used to select important parameters and must be used together with **feature_filter_engine**.	Integer	-
split_count	Number of evenly selected parameters in the value range of the tuned parameters. It is used to select important parameters and must be used together with **feature_filter_engine**.	Integer	-
benchmark	Performance test script.	-	-
evaluations	Performance evaluation metrics. For details, see **Table 9-16**.	-	-

Table 9-16 evaluations parameters

Parameter	Description	Type	Value
name	Evaluation metric name.	String	-
get	Script for obtaining performance evaluation results.	-	-

Parameter	Description	Type	Value
type	How evaluation results are judged. For **positive**, a larger number indicates better performance. For **negative**, a smaller number indicates better performance.	Enumerated	"positive","negative"
weight	Metric weight.	Integer	0-100
threshold	Minimum performance required by a metric.	Integer	User-defined

Configuration examples

YAML configuration file on the server:

```
project: "compress"
maxiterations: 500
startworkload: ""
stopworkload: ""
object :
  -
    name : "compressLevel"
    info :
        desc : "The compresslevel parameter is an integer from 1 to 9 controlling the level of compression"
        get : "cat /root/A-Tune/examples/tuning/compress/compress.py | grep 'compressLevel=' | awk -F '='
'{print $2}'"
        set : "sed -i 's/compressLevel=\\s*[0-9]*/compressLevel=$value/g' /root/A-Tune/examples/tuning/
compress/compress.py"
        needrestart : "false"
        type : "continuous"
        scope :
          - 1
          - 9
        dtype : "int"
  -
    name : "compressMethod"
    info :
        desc : "The compressMethod parameter is a string controlling the compression method"
        get : "cat /root/A-Tune/examples/tuning/compress/compress.py | grep 'compressMethod=' | awk -F '='
'{print $2}' | sed 's/\"//g'"
        set : "sed -i 's/compressMethod=\\s*[0-9,a-z,\"]*/compressMethod=\"$value\"/g' /root/A-Tune/
examples/tuning/compress/compress.py"
        needrestart : "false"
        type : "discrete"
        options :
          - "bz2"
          - "zlib"
          - "gzip"
        dtype : "string"
```

YAML configuration file on the client:

```
project: "compress"
engine : "gbrt"
```



```
iterations : 20
random_starts : 10

benchmark : "python3 /root/A-Tune/examples/tuning/compress/compress.py"
evaluations :
 -
   name: "time"
   info:
       get: "echo '$out' | grep 'time' | awk '{print $3}'"
       type: "positive"
       weight: 20
 -
   name: "compress_ratio"
   info:
       get: "echo '$out' | grep 'compress_ratio' | awk '{print $3}'"
       type: "negative"
       weight: 80
```

Usage examples

- Download test data.
  ```
  wget http://cs.fit.edu/~mmahoney/compression/enwik8.zip
  ```
- Create a **prepare.sh** file for preparing the tunning environment.
  ```
  #!/usr/bin/bash
  if [ "$#" -ne 1 ]; then
    echo "USAGE: $0 the path of enwik8.zip"
    exit 1
  fi
  path=$(
    cd "$(dirname "$0")"
    pwd
  )
  echo "unzip enwik8.zip"
  unzip "$path"/enwik8.zip

  echo "set FILE_PATH to the path of enwik8 in compress.py"
  sed -i "s#compress/enwik8#$path/enwik8#g" "$path"/compress.py

  echo "update the client and server yaml files"
  sed -i "s#python3 .*compress.py#python3 $path/compress.py#g" "$path"/compress_client.yaml
  sed -i "s# compress/compress.py# $path/compress.py#g" "$path"/compress_server.yaml

  echo "copy the server yaml file to /etc/atuned/tuning/"
  cp "$path"/compress_server.yaml /etc/atuned/tuning/
  ```

  Run a script.
  ```
  sh prepare.sh enwik8.zip
  ```
- Perform tuning.
  ```
  atune-adm tuning --project compress --detail compress_client.yaml
  ```
- Restore to the initial settings before the tuning. **compress** is the project name in the YAML file.
  ```
  atune-adm tuning --restore --project compress
  ```


10 Kernel Functions and Interfaces

10.1 OOM Process Control Policy

Background

Both offline and online services can be configured in an OS. When Out Of Memory (OOM) occurs, the system preferentially ends the process that consumes the most memory in the offline service control group to reclaim the memory. However, some core services are often running offline. If the memory consumed by such services is reclaimed, the OS will be greatly affected.

To solve this problem, HCE adjusts the memory reclamation policy during OOM and adds the function of configuring cgroups priority. When the memory is insufficient, the kernel traverses cgroups, ends the processes for cgroups with low priorities, and reclaims the memory so that important offline services can keep running.

Prerequisites

vm.panic_on_oom is disabled (value: **0**) by default, which means if the system OOM occurs, kernel panic will not be triggered. Before you use memcg OOM for priority configuration (**memcg_qos_enable** is set to **1** or **2**), if the value of **vm.panic_on_oom** is not **0**, run **sysctl -w vm.panic_on_oom=0** first.

Interface Description

Table 10-1 Interface details

Interface	Description	Example Value
memcg_qos_enable	Specifies whether to enable memcg OOM priority configuration. • **0**: Priority configuration is disabled. When OOM occurs, the process that consumes the most memory is ended based on the original OOM operation, and the memory is reclaimed. • **1**: Priority configuration is enabled, and priorities are configured by cgroup. When OOM occurs, all processes in the cgroup with a lower priority are ended and the memory is reclaimed. • **2**: Priority configuration is enabled, and priorities are configured by process. When OOM occurs, the largest process in the cgroup with a lower priority is ended and the memory is reclaimed.	The value is an integer ranging from **0** to **2**. The default value is **0**.
memory.qos_level	Specifies how to configure the priorities of cgroups. A smaller value indicates a lower priority. • When OOM occurs, the current cgroup is used as the parent cgroup, the process with the highest memory usage in the child cgroup with the lowest priority is ended, and the memory is reclaimed. • When OOM occurs, cgroups with the same priority will be sorted based on their memory usage, and the cgroup with the largest memory usage is ended. **NOTE** • Before using **memory.qos_level**, ensure that **memcg_qos_enable** is set to **1** or **2**. • By default, the value of **memory.qos_level** of a newly created cgroup inherits the value of **memory.qos_level** of the parent cgroup. The priority of a child cgroup is not restricted by the parent cgroup. • If the priority of the parent cgroup is changed, the priorities of the child cgroups are automatically changed to be the same as that of the parent cgroup.	The value is an integer ranging from **-1024** to **1023**. The default value is **0**.

Interface Configuration Example

Create six cgroups A, B, C, D, E and F, configure the **memcg_qos_enable** interface, and set the memcg OOM priorities by specifying **memory.qos_level**.

Figure 10-1 Priorities

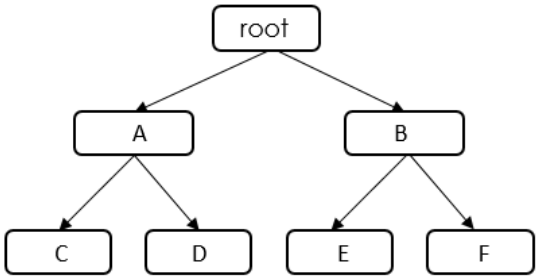


Table 10-2 Data planning

cgroup	Value of memory.qos_level	Description
A	**-8**	When the OOM operation is performed in the root cgroup, the kernel traverses all cgroups in the root cgroup and finally selects cgroups A and E, both with the lowest priority. Because A and E have the same priority, the kernel continues to compare the memory used by A and E. • If you set **memcg_qos_enable** to **1**, the system preferentially ends all processes in the cgroup with a large memory usage and reclaims the memory. • If you set **memcg_qos_enable** to **2**, the system ends the process with the largest memory usage in two cgroups and reclaims the memory.
B	10	
C	1	
D	2	
E	**-8**	
F	3	

1. Disable **vm.panic_on_oom**.
   ```
   sysctl -w vm.panic_on_oom=0
   ```
2. Enable memcg OOM priority configuration.
   ```
   echo 1 > /proc/sys/vm/memcg_qos_enable
   ```
3. Create cgroups A and B and set their memcg OOM priorities to **-8** and **10**.
   ```
   mkdir /sys/fs/cgroup/memory/A
   mkdir /sys/fs/cgroup/memory/B
   cd /sys/fs/cgroup/memory/A
   echo -8 > memory.qos_level
   cd /sys/fs/cgroup/memory/B
   echo 10 > memory.qos_level
   ```


4. Create child cgroups C and D under cgroup A and child cgroups E and F under cgroup B, and set the memcg OOM priorities of cgroups C, D, E, and F to **1**, **2**, **-8**, and **3**.

```
mkdir /sys/fs/cgroup/memory/A/C
mkdir /sys/fs/cgroup/memory/A/D
mkdir /sys/fs/cgroup/memory/B/E
mkdir /sys/fs/cgroup/memory/B/F
cd /sys/fs/cgroup/memory/A/C
echo 1 > memory.qos_level
cd /sys/fs/cgroup/memory/A/D
echo 2 > memory.qos_level
cd /sys/fs/cgroup/memory/B/E
echo -8 > memory.qos_level
cd /sys/fs/cgroup/memory/B/F
echo 3 > memory.qos_level
```

10.2 Multi-level Memory Reclamation Policy

Background

In high-density hybrid container deployments, offline services with a large number of I/O reads and writes consume a large amount of page cache. As a result, the idle memory of the system decreases, and memory reclamation is triggered when the idle memory watermark is reached. Online tasks enter the slow path for memory reclamation when applying for memory, causing latency and jitter.

To solve this problem, multi-level memory reclamation is provided by HCE 2.0. You can set a memory warning value to trigger a memory reclamation task, which ensures available memory space. For memory reclamation, you can set multiple levels of memory protection watermarks to protect task availability.

Constraints

memory.min and memory.low take effect only on leaf cgroups. When a memory cgroup is created, memory.min and memory.low of the parent cgroup are cleared.

Interface Description

The memory.min, memory.low, and memory.high interfaces exist in the non-root memory cgroup by default. You can write values to the files or read the current configuration. The proper value sequence is memory.min ≤ memory.low < memory.high. The three values can be used independently or together.

The following figure shows the memory reclamation mechanism.

Figure 10-2 Memory reclamation mechanism

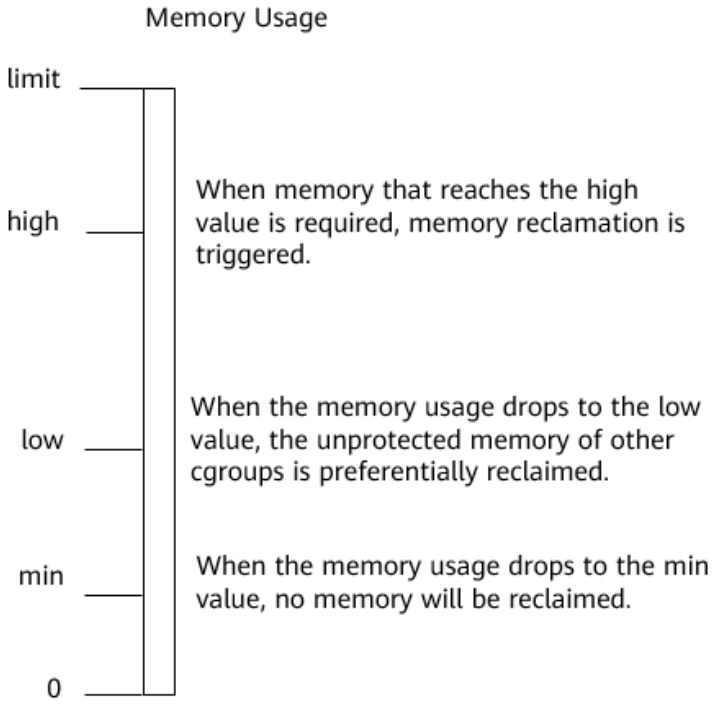


Table 10-3 Interface parameters

Interface	Description
memory.min	Specifies the minimum amount of memory the cgroup must always retain. The default value is **0**. Even if there is no memory that can be reclaimed, the system will not reclaim the memory that is less than or equal to the value of this parameter. The read and write operations are described as follows: • Reading this interface can view the size (in byte) of the protection memory. • Writing to this interface can set the size of the protection memory. The unit is not limited. • The value ranges from **0** to **memory.limit_in_bytes**.
memory.low	Specifies the best-effort memory protection. The default value is **0**. The system preferentially reclaims the memory of unprotected cgroups. If the memory is still insufficient, the system reclaims the memory between **memory.min** and **memory.low**. The read and write operations are described as follows: • Reading this interface can view the best-effort memory protection value, in bytes. • Writing to this interface can set the best-effort memory protection value. The unit is not limited. • The value ranges from **0** to **memory.limit_in_bytes**.

Interface	Description
memory.high	Specifies the memory reclamation warning. The default value is **max**. When the memory usage of a cgroup reaches the high value, a synchronous memory reclamation task is triggered for the cgroup and its child cgroups. The memory is limited to a value lower than the high value to prevent OOM caused by the memory limit. The read and write operations are described as follows: • Reading this interface can view the Throttle limit, in bytes. • Writing to this interface can set the Throttle limit. The unit is not limited. • The value ranges from **0** to **memory.limit_in_bytes**.

Interface Configuration Example

Create cgroups A, B, C, D, E and F and configure the memory.min interface.

Figure 10-3 Multi-level memory reclamation

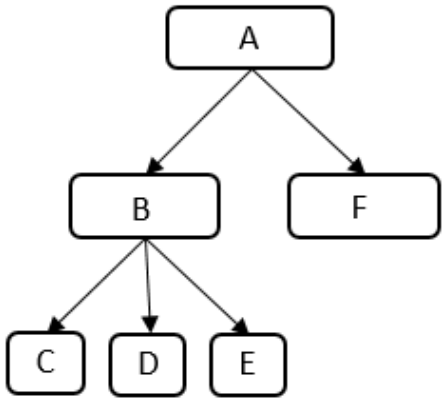


Table 10-4 Data planning

cgroup	memory.limit_in_bytes	memory.min	memory.usage_in_bytes
A	200M	0	-
B	-	0	-
C	-	75M	50M
D	-	25M	50M
E	-	0	50M
F	-	125M	-

1. Create cgroup A and set **memory.limit_in_bytes** to **200M**.
```
mkdir /sys/fs/cgroup/memory/A
echo 200M > /sys/fs/cgroup/memory/A/memory.limit_in_bytes
```
2. Create cgroup B.
```
mkdir /sys/fs/cgroup/memory/A/B
```
3. Create cgroup C, set **memory.min** to **75M**, and create a process that will use 50-MB cache in the cgroup.
```
mkdir /sys/fs/cgroup/memory/A/B/C
echo 75M > /sys/fs/cgroup/memory/A/B/C/memory.min
```
4. Create cgroup D, set **memory.min** to **25M**, and create a process that will use 50-MB cache in the cgroup.
```
mkdir /sys/fs/cgroup/memory/A/B/D
echo 25M > /sys/fs/cgroup/memory/A/B/D/memory.min
```
5. Create cgroup E, set **memory.min** to **0**, and create a process that will use 50-MB cache in the cgroup.
```
mkdir /sys/fs/cgroup/memory/A/B/E
```
6. Create cgroup F, set **memory.min** to **125M**, and request 125-MB cache for memory protection.
```
mkdir /sys/fs/cgroup/memory/A/F
echo 125M > /sys/fs/cgroup/memory/A/F/memory.min
```

Information similar to the following is displayed:

cgroup C: memory.min=75M, memory.usage_in_bytes=50M

cgroup D: memory.min=25M, memory.usage_in_bytes=25M

cgroup E: memory.min=0, memory.usage_in_bytes=0

cgroup B: memory.usage_in_bytes=75M

10.3 Multi-level Hybrid Scheduling of Kernel CPU cgroups

Background

In hybrid deployments, the Linux kernel scheduler assigns more scheduling opportunities to high-priority tasks and minimizes the impact of low-priority tasks on kernel scheduling. However, the two-level scheduling of online and offline services cannot meet this requirement.

To solve the problem, HCE 2.0 allows multi-level scheduling of kernel CPU cgroups and provides **/sys/fs/cgroup/cpu/cpu.qos_level** to extend the scheduling levels from two to five, allowing users to set the priority for each cgroup separately.

Constraints

cpu.qos_level is only available for cgroup-v1 and not for cgroup-v2.

Interface Description

Rules for **cpu.qos_level** to take effect:

- Completely Fair Scheduler (CFS) selects **task_group** level by level from top to bottom. cpu.qos_level takes effect for child cgroups under the same parent cgroup.
- When a child cgroup is created, it inherits the cpu.qos_level value of the parent cgroup by default, but the cpu.qos_level value can be reconfigured.
- For QoS levels with the same priority, their resource competition complies with the policy of CFS.
- On the same CPU, the tasks whose **qos_level** is less than **0** are always unconditionally preempted by tasks whose **qos_level** is greater than or equal to **0**, regardless of their levels.

When a high-priority task is scheduled:

- Online tasks can unconditionally preempt the CPU resources of offline tasks. During multi-core scheduling, online tasks can preferentially preempt the CPU resources of offline tasks on other cores. In the hyper-thread scenario, online tasks with priority 2 can evict offline tasks on the SMT.
- When a task with a higher priority is woken up, the task is accelerated by time slicing and can immediately preempt the CPU resources of tasks with a lower priority to achieve a response at a lower latency (the minimum running time slice of CFS is ignored).

Table 10-5 Interface description

Interface	Description
cpu.qos_level	Specifies the CPU priorities of the cgroups. The value is an integer ranging from **-2** to **2**. The default value is **0**. • cpu.qos_level >= 0 Tasks in the cgroup are online tasks, which can unconditionally preempt offline tasks. A lower value indicates a lower priority (0 < 1 < 2). Online tasks with a higher priority can obtain more CPU resources than those with a lower priority. • cpu.qos_level < 0 Tasks in the cgroup are offline tasks. The priority of -1 is higher than that of -2, meaning that tasks at level -1 have more CPU resources than tasks at level -2. If a parent cgroup is running offline services, child cgroups can only inherit the priority of the parent cgroup, and the priority cannot be changed.

Interface Configuration Example

Create cgroups A, B, and C, and configure the cpu.qos_level interface.

Figure 10-4 cgroup nodes A, B, and C

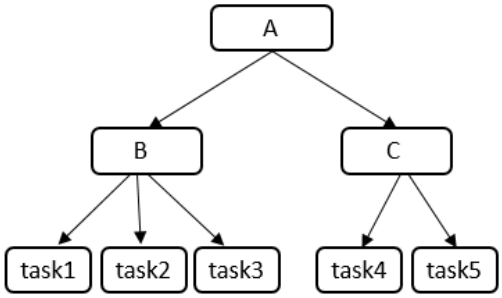


Table 10-6 Data planning

cgroup	cpu.qos_level Value
A	1
B	-2
C	2

1. Create cgroup A and child cgroups B and C, and set their CPU scheduling priorities to **1**, **-2**, and **2**.

   Tasks in cgroups A and C can unconditionally preempt CPU resources of the tasks in cgroup B. cgroup C preferentially preempts CPU resources because the priority of cgroup C is higher than that of cgroup A.

   ```
   mkdir -p /sys/fs/cgroup/cpu/A
   echo 1 > /sys/fs/cgroup/cpu/A/cpu.qos_level
   mkdir -p /sys/fs/cgroup/cpu/A/B
   echo -2 > /sys/fs/cgroup/cpu/A/B/cpu.qos_level
   mkdir -p /sys/fs/cgroup/cpu/A/C
   echo 2 > /sys/fs/cgroup/cpu/A/C/cpu.qos_level
   ```

2. Add the task1, task2, and task3 processes to cgroup B.

   The CPU scheduling priority of the task1, task2, and task3 processes is **-2**.

   ```
   echo $PID1 > /sys/fs/cgroup/cpu/A/B/tasks
   echo $PID2 > /sys/fs/cgroup/cpu/A/B/tasks
   echo $PID3 > /sys/fs/cgroup/cpu/A/B/tasks
   ```

3. Add the task4 and task5 processes to cgroup C.

   The CPU scheduling priority of the task4 and task5 processes is **2**.

   ```
   echo $PID4 > /sys/fs/cgroup/cpu/A/C/tasks
   echo $PID5 > /sys/fs/cgroup/cpu/A/C/tasks
   ```

4. View the CPU scheduling priority and processes of cgroup B.

   ```
   [root@localhost cpu_qos]# cat /sys/fs/cgroup/cpu/A/B/cpu.qos_level
   -2
   [root@localhost boot]# cat /sys/fs/cgroup/cpu/A/B/tasks
   1879
   1880
   1881
   ```

5. View the CPU scheduling priority and processes of cgroup C.

   ```
   [root@localhost cpu_qos]# cat /sys/fs/cgroup/cpu/A/C/cpu.qos_level
   2
   [root@localhost boot]# cat /sys/fs/cgroup/cpu/A/C/tasks
   1882
   1883
   ```


10.4 Kernel Exception Events Analysis

Background

When HCE is running, there are some inevitable kernel events, such as **soft lockup**, **RCU (Read-Copy Update) stall**, **hung task**, **global OOM**, **cgroup OOM**, **page allocation failure**, **list corruption**, **bad mm_struct**, **I/O error**, **EXT4-fs error**, **Machine Check Exception (MCE)**, **fatal signal**, **warning**, **panic**, and **oops**. This section describes those events and how you can trigger them.

Soft Lockup

A soft lockup is the symptom of a task or kernel thread not releasing a CPU for a period longer than allowed (20 seconds by default).

- Details

  A soft lockup is triggered by the watchdog mechanism of the Linux kernel. The kernel starts a FIFO real-time kernel thread (watchdog) with the highest priority for each CPU. The thread name is watchdog/0, watchdog/1, and so on. The thread invokes the watchdog function every 4 seconds by default. Each time the function is invoked, an hrtimer will be reset to expire after a soft lockup threshold, which is 2 times the duration specified by **watchdog_thresh** (a kernel parameter) and defaults to 20 seconds.

  Within this duration, if watchdog is not scheduled and the hrtimer expires, the kernel prints a soft lockup exception similar to the following:

  ```
  BUG: soft lockup - CPU#3 stuck for 23s! [kworker/3:0:32]
  ```

- Triggering method

  Disable interrupts or preemption to result in an infinite loop.

RCU Stall

An RCU stall is an exception that RCU kernel threads are not scheduled within the RCU grace period.

- Details

  RCU readers are allowed to access any data, and RCU records information about these readers. When RCU writers are updating data, they copy a backup and modify the data on the backup. After all readers exit, writers replace the old data at a time.

  Writers can only replace the old data after all readers stop referencing the old data. This period of time is a grace period.

  If the readers do not exit even after the grace period expires and the writers wait for a period longer than the grace period, an RCU stall will be reported.

- Triggering method

  Stimulate a scenario described in **Documentation/RCU/stallwarn.txt** to trigger RCU stalls. An example is that CPU keeps looping in the RCU read-side critical section and keeps looping when the interrupt or preemption function is disabled.

Hung Task

When the kernel detects that a process is in the **D** state for a period longer than the specified time, a hung task exception is reported.

- Details

  One status of a process is **TASK_UNINTERRUPTIBLE**, which is also called the **D** state. A process in the **D** state can be woken up only by **wake_up**. When the kernel introduces the **D** state, the process waits for the I/O to complete. When I/Os are normally, the process should not be in the **D** state for a long time.

  The kernel creates a thread (khungtaskd) to periodically traverse all processes in the system and check whether there is a process that is in the **D** state for a period longer than the preset duration (120 seconds by default). If there is such a process, related warnings and process stacks will be printed and reported. If **hung_task_panic** is configured (through proc or kernel startup parameters), a panic is initiated directly.

- Triggering method

  Create a kernel thread, set it to the **D** state, and use the scheduler to release the time slice.

Global OOM

The Linux OOM killer is a memory management mechanism. When there is less available memory, the kernel kills some processes to release some memory to ensure system continuity.

- Details

  When the kernel allocates memory to a process but the system memory is insufficient, OOM will occur. The OOM killer traverses all processes, scores the processes based on their memory usage, selects a process with the highest score, and terminates this process to release memory.

  The kernel source code is linux/mm/oom_kill.c, and the core function is out_of_memory(). The following describes the processing flow:

  a. The kernel instructs the modules that are registered with oom_notify_list in the system to release some memory. If these modules release some memory, it will take no more actions. If the memory fails to be reclaimed, it will go to the next step.
  b. Generally, the OOM killer is triggered when the kernel is allocating memory to a process. If the process has a pending SIGKILL or is exiting, the kernel will terminate this process to release memory. Otherwise, the kernel will go to the next step.
  c. The kernel checks the settings of the system administrator using panic_on_oom and determines whether to perform OOM killer or panic in case of OOM. If the kernel selects panic, the system will crash and restart. If the kernel selects OOM killer, it will go to the next step.
  d. The kernel enters the OOM killer and checks the system settings. The system administrator can terminate the process that attempts to request memory and causes OOM, or other processes. If the system administrator chooses to terminate the current process, the OOM killer stops. Otherwise, the kernel will go to the next step.

e. The kernel invokes select_bad_process to select appropriate processes, and then invokes oom_kill_process to terminate the selected processes. If select_bad_process does not select any process, the kernel will enter the panic state.

- Triggering method

  Execute the program that occupies large memory until the memory is insufficient.

cgroup OOM

- Difference from global OOM

  The memory of cgroup OOM is different from that of global OOM. When the memory usage of processes in the cgroup exceeds the upper limit, the cgroup kills the processes to release the memory.

- Triggering method

  Specify a process in a cgroup to use large amounts of memory until the memory is insufficient.

Page Allocation Failure

A page allocation failure is an error reported by the system when a program fails to apply for an idle page. When a program applies for memory of an order, but there is no idle page whose order is higher than the required order in the system memory, the kernel reports an error.

- Details

  Linux uses the buddy system to efficiently allocate and manage memory. All idle page tables (with a size of 4 KB per page table) are linked to an array containing 11 elements. Each element in the array forms a linked list with consecutive page tables of the same size. The number of page tables is 1, 2, 4, 8, 16, 32, and 64, or 128, 256, 512, and 1,024. The maximum continuous memory that can be allocated at a time is 4 MB, the memory of 1,024 continuous 4-KB page tables.

  Assume that you apply for memory that contains 256 page tables and whose order is 6. The system searches for the ninth, tenth, and eleventh linked lists in the array in sequence. If the previous linked list is empty, there is no free memory of this order. The system searches for the next linked list until the last linked list.

  If all linked lists are empty, the application fails. The kernel will report a page allocation failure and display an error message indicating that the memory page whose order is 6 fails to be requested.

  ```
  page allocation failure:order:6
  ```

- Triggering method

  Use alloc_pages to continuously apply for high-order memory pages (for example, order=10) and do not release the memory pages until the application fails.

List Corruption

A list corruption error is reported when the kernel fails to check the validity of a linked list. There are two error types: list_add corruption and list_del corruption.

- Details

  The kernel provides list_add and list_del to check the validity of the linked list and to add or delete an entry from the linked list if it is valid. If the linked list is invalid, a list corruption error is reported. The kernel source code is lib/list_debug.c.

  Figure 10-5 Error types: list_add corruption and list_del corruption

```
bool __list_add_valid(struct list_head *new, struct list_head *prev,
              struct list_head *next)
{
    if (CHECK_DATA_CORRUPTION(next->prev != prev,
            "list_add corruption. next->prev should be prev (%px), but was %px. (next=%px).\n",
            prev, next->prev, next) ||
        CHECK_DATA_CORRUPTION(prev->next != next,
            "list_add corruption. prev->next should be next (%px), but was %px. (prev=%px).\n",
            next, prev->next, prev) ||
        CHECK_DATA_CORRUPTION(new == prev || new == next,
            "list_add double add: new=%px, prev=%px, next=%px.\n",
            new, prev, next))
        return false;

    return true;
}
EXPORT_SYMBOL(__list_add_valid);

bool __list_del_entry_valid(struct list_head *entry)
{
    struct list_head *prev, *next;

    prev = entry->prev;
    next = entry->next;

    if (CHECK_DATA_CORRUPTION(next == LIST_POISON1,
            "list_del corruption, %px->next is LIST_POISON1 (%px)\n",
            entry, LIST_POISON1) ||
        CHECK_DATA_CORRUPTION(prev == LIST_POISON2,
            "list_del corruption, %px->prev is LIST_POISON2 (%px)\n",
            entry, LIST_POISON2) ||
        CHECK_DATA_CORRUPTION(prev->next != entry,
            "list_del corruption. prev->next should be %px, but was %px\n",
            entry, prev->next) ||
        CHECK_DATA_CORRUPTION(next->prev != entry,
            "list_del corruption. next->prev should be %px, but was %px\n",
            entry, next->prev))
        return false;

    return true;

}
EXPORT_SYMBOL(__list_del_entry_valid);
```

  This error is typically caused by abnormal memory operations, such as memory corruption and memory damage.

- Triggering method

  Use the standard kernel interface of list.h to create a linked list, illegally modify the previous or next pointer of a linked list entry, and then call the kernel list_add or list_del interface.

Bad mm_struct

A bad mm_struct error is reported when one or more mm_struct data structures in the kernel are corrupted or damaged.

- Details

  mm_struct is an important data structure in the Linux kernel. It is used to trace the virtual memory area of a process. If the data structure is damaged, the process or system may break down. This error is usually caused by memory exceptions. For example, the memory in mm_struct is corrupted or memory overwriting occurs.

- Triggering method

  Bad mm_struct is triggered when there is a hardware error or Linux kernel code error.

I/O Error

An I/O error is reported when an input/output operation fails. This error may be printed when the driver of the I/O device such as the NIC or disk is abnormal or the file system is abnormal.

- Details

  The condition under which the code fails to be executed is the cause of this error. Common causes are hardware faults, disk damage, file system errors, driver problems, and permission problems. For example, if an error occurs when the system attempts to read data from or write data to a disk, an I/O error is reported.

- Triggering method

  When the system is reading data from or writing data to the disk, remove the disk to damage the disk data.

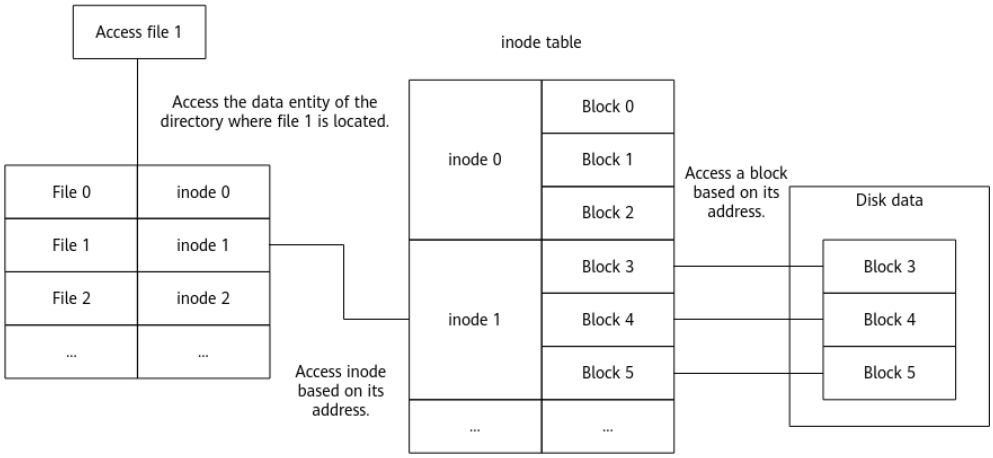
EXT4-fs Error

EXT4-fs errors typically indicate problems with the ext4 file system.

- Details

  A sector is the minimum file storage unit on a storage device. Multiple consecutive sectors form a block. inode stores the metadata of a file, including the creator, creation date, file size, attributes, and the number of blocks. If the inode information in EXT4 format fails to be verified, an EXT4-fs error will be reported.

  Figure 10-6 Structure of the ext4 file system



  The kernel ext4 verification uses checksum to verify inode information. When there is a partition table error or the disk is damaged, the kernel returns the EIO (Input/Output Error) error code and the system reports "EXT4-fs error checksum invalid".

- Triggering method

  Forcibly remove the disk and add it back to read the data.

MCE

An MCE is a type of hardware error that occurs when a CPU detects a hardware problem. The interrupt number is 18, and the exception type is abort.

- Details

  MCEs are caused by bus faults, memory ECC errors, cache errors, TLB errors, or internal clock errors. In addition to hardware faults, inappropriate BIOS configurations, firmware bugs, and software bugs may also cause MCEs.

  When an MCE is reported, the OS checks a group of registers called Machine-Check MSR and executes the corresponding function based on the error codes of the registers. (The function varies depending on the chip architecture.)

- Triggering method

  An MCE is reported when there is a bus fault, memory ECC error, cache error, TLB error, or internal clock error.

Fatal Signal

If a signal cannot be ignored or handled through user-defined processing functions, it is a fatal signal, such as SIGKILL, SIGSTOP, and SIGILL.

- Details

  The signal mechanism is an asynchronous notification mechanism for communication between processes in the system. When a signal is sent to a process but the OS interrupts the process, all non-atomic operations are interrupted.

  If a signal is SIGKILL, SIGSTOP, or SIGILL, it is a fatal signal.

  Figure 10-7 Example of a fatal signal

  ```
  #define sig_fatal(t, signr) \
      (!siginmask(signr, SIG_KERNEL_IGNORE_MASK|SIG_KERNEL_STOP_MASK) && \
  ```

- Triggering method

  Use a user-mode program to execute invalid instructions or run **kill -9** to kill the process.

Warning

Warning is an action taken when a kernel issue is detected and immediate attention is required. Warning prints the call stack information when the issue occurs. The OS continues to run after a warning.

- Details

  Warning is triggered when macros such as WARN, WARN_ON, and WARN_ON_ONCE are invoked.

  There are several causes of invoking a warning macro. You need to trace the call stack to locate the cause. A warning macro does not change the system status and does not provide guidance for handling the warning.

- Triggering method

  Trigger a warning when the system is invoking a macro.

Panic

A kernel panic refers to the action taken by the OS when it detects a fatal internal error and cannot securely handle the error. When an exception occurs during kernel running, the kernel uses the kernel_panic function to print all the information obtained when the exception occurs.

- Details

  There are various causes for the exception. Common causes include kernel stack overflow, division by zero, memory access out of bounds, and kernel deadlock. When this exception occurs, locate the cause of kernel_panic based on the invoking information printed for the exception.

- Triggering method

  Read address 0 in kernel mode.

oops

An oops is generated when the kernel detects a serious problem, such as invalid memory access, division by zero, and invalid instructions in the kernel code.

- Details

  When the kernel detects these problems, it generates a report to record the context, including the register status, call trace, and memory status. This report will be printed to the system log. You can run **dmesg** to check the report. The kernel attempts to recover and continue operations. However, if the problem is serious and cannot be rectified, the system may crash or restart.

- Triggering methods

  a. Invalid memory access: Access unallocated or released memory.

  b. Division by zero: Perform division by zero in the kernel code.

  c. Invalid instructions: Execute an invalid or undefined instruction.

  d. Hardware faults: A hardware fault occurs, for example, a memory or CPU fault.

10.5 Huge Pages

Background

For some frequently accessed applications with large code segments, the TLB miss rate is high. This problem can be resolved by using huge pages. This feature is disabled by default in HCE. You can enable it by setting a parameter. Executable code of an application can be loaded into huge pages to reduce the TLB miss rate and improve performance of databases (MySQL) and other large applications.

Enabling Huge Pages

- Check whether the kernel startup parameter **exec_hugepages** exists. If not, add it and restart the system.
- Enable huge pages for a single application.


```
HUGEPAGE_ELF=1./app    // This environment variable will not be copied to a subprocess.
```

HUGEPAGE_ELF=0 disables huge pages.

- Enable huge pages for applications marked by hugepageedit.

```
export HUGEPAGE_PROBE=1    // Export the environment variable HUGEPAGE_PROBE. This variable can be copied to a subprocess.
hugepageedit./app          // Use hugepageedit to mark binary files of the applications for which huge pages need to be enabled. hugepageedit is provided in the glibc-devel package.
./app                      // Start the applications to enable huge pages.
```

- Enable huge pages for all applications.

```
echo 1 > /sys/kernel/mm/exec_hugepages/enabled    // echo 0 (default) indicates disabled. After this feature is enabled, all applications that meet the conditions will automatically use huge pages.
```

- Check whether huge pages are enabled.

```
cat /proc/ Process ID/smaps | grep eh    // If a process is marked with eh, huge pages are enabled for this process.
```

Constraints

- If application code is not 2 MB aligned, the code that is less than 2 MB in each segment cannot use huge pages. So, alignment is recommended.
- mprotect() requires input addresses be 2 MB aligned. That is, their size must be a multiple of 2 MB. Application code needs to be modified accordingly.

  For address alignment of 2 MB, compile the program to add **-Wl,-zcommon-page-size=2097152 -Wl,-zmax-page-size=2097152** to linker options.
- If reserved huge pages are insufficient for an application, the system will apply for a page of 2 MB. If the application fails due to memory fragments or other reasons, rollback to small pages will be performed. If multiple threads in an application roll back to small pages, SIGBUS will be triggered.

10.6 Custom TCP Retransmission Rules

Background

TCP packet retransmissions comply with the exponential backoff principle. In a low-quality network, the packet arrival rate is low and the latency is high. To address this issue, custom TCP retransmission rules can be created by setting parameters. You can specify the number of linear backoffs, maximum number of retransmissions, and maximum retransmission interval, to increase the packet arrival rate and reduce the latency in a low-quality network.

Parameter Description

- Use **sysctl** to enable or disable customization of TCP retransmission rules.

  net.ipv4.tcp_sock_retrans_policy_custom determines whether to enable customization of TCP retransmission rules.

  0: Disable customization of TCP retransmission rules.

  1: Enable customization of TCP retransmission rules.

```
sysctl -w net.ipv4.tcp_sock_retrans_policy_custom=1
```

- Use **setsockopt** to create TCP retransmission rules for specified sockets.
  - Commands:

```
int setsockopt(int sockfd, int level, int optname, const void *optval, socklen_t optlen);
int getsockopt(int sockfd, int level, int optname, const void *optval, socklen_t optlen);
```


optname passes enumerated options. **optval** passes the start address of the structure.

– Enumerated options:

```
TCP_SOCK_RETRANS_POLICY_CUSTOM 100
```

Structure:

```
struct tcp_sock_retrans_policy {
    uint8_t tcp_linear_timeouts_times; /* number of times linear backoff */
    uint8_t tcp_retries_max; /* maximum retransmission times */
    uint8_t tcp_rto_max; /* maximum RTO time, unit:second */
    uint8_t unused;
};
```

Where:

i. **tcp_linear_timeouts_times** indicates the number of linear backoffs. The value ranges from **0** to **32**.

0: The default rule (exponential backoff) is used.

Non-**0**: A non-0 value indicates a custom number of linear backoffs.

NOTE

- Exponential backoff: When packet retransmission occurs, the RTO value doubles.
- Linear backoff: When packet retransmission occurs, the RTO value remains unchanged.

ii. **tcp_retries_max** indicates the maximum number of retransmissions. The value ranges from **0** to **64**.

0: The maximum number of retransmissions (15 by default) defined in the system will be used.

Non-**0**: A non-0 value indicates a custom maximum number of retransmissions.

The value of **tcp_retries_max** must be larger than the minimum number of retransmissions (**3** by default) defined by the **sysctl** option **net.ipv4.tcp_retries1**.

iii. **tcp_rto_max** indicates the maximum retransmission interval, in seconds. The value must be **0** or an integer ranging from **20** to **120**.

0: The default maximum retransmission interval (120s) is used.

Non-**0**: A non-0 value indicates a custom maximum retransmission interval.

How to Use

- Enable customization of TCP retransmission rules. Set **net.ipv4.tcp_sock_retrans_policy_custom** to **1**.

```
[root@localhost ~]# sysctl -w net.ipv4.tcp_sock_retrans_policy_custom=1
```

Set the number of linear backoffs to **4**, the maximum number of retransmissions to **10**, and the maximum retransmission interval to 20s.

```
tcp_sock_retrans_policy policy = {0};
policy.tcp_linear_timeouts_times = 4;
policy.tcp_retries_max = 10;
policy.tcp_rto_max = 20;
setsockopt(sockfd, SOL_TCP, TCP_SOCK_RETRANS_POLICY_CUSTOM, (const void*)&policy,
sizeof(struct tcp_sock_retrans_policy));
```

- Restore to the default rule.


```
tcp_sock_retrans_policy policy = {0};
setsockopt(sockfd, SOL_TCP, TCP_SOCK_RETRANS_POLICY_CUSTOM, (const void*)&policy,
sizeof(struct tcp_sock_retrans_policy));
```

- Disable customization of TCP retransmission rules.

```
[root@localhost ~]# sysctl -w net.ipv4.tcp_sock_retrans_policy_custom=0
```

CAUTION

Customization of TCP retransmission rules is modified in the HCE 2.0 kernel. To use this function in glibc of HCE 2.0, you need to add the **TCP_SOCK_RETRANS_POLICY_CUSTOM** macro and the **tcp_sock_retrans_policy** structure to the user-mode code.

Constraints

1. This function is only valid for TCP connections in the ESTABLISHED state.
2. IPv4 and IPv6 are supported. Due to historical reasons, TCP-related **sysctl** options are under **net.ipv4**. The **sysctl** options starting with **net.ipv4.tcp** are valid for both IPv4 and IPv6.
3. An application-level TCP retransmission rule has a higher priority than other global TCP retransmission settings.
4. Custom TCP retransmission rules may cause high system loads. Create custom rules based on available system resources. No more than 6 linear backoffs are recommended.
5. If the maximum number of retransmissions is less than the value of **net.ipv4.tcp_retries1** (default: **3**), network detection cannot be triggered. As a result, packets may fail to be sent in some network change scenarios.
6. If the maximum number of retransmissions is not **0**, the value must be no smaller than the number of linear backoffs. Otherwise, an error will be returned.
7. If the maximum retransmission interval is specified but the maximum number of retransmissions is not, the latter may be greater than the value of **net.ipv4.tcp_retries2** (default: **15**). The value of **net.ipv4.tcp_retries2** (R2 for short) indicates the maximum number of retransmissions or the maximum retransmission interval. According to RFC 6069, if R2 is used to describe the number of retransmissions, it must be converted into the retransmission interval. To prevent TCP disconnection from occurring too early after conversion is enabled, the default value of **TCP_RTO_MAX** (120s) instead of the custom maximum retransmission interval is used for the conversion. As a result, the actual number of retransmissions is greater than R2.
8. According to the TCP-TLP algorithm, the tail packet that is not acknowledged may be selected by the Loss Probe timer for retransmission to trigger fast retransmission. This retransmission is not counted in the number of retransmissions defined by the custom rule.

10.7 Soft Binding of CPUs

Background

With NUMA affinity, if the load of two VMs is high but other VMs are idle, idle CPUs cannot be used. Without NUMA affinity, the performance deteriorates significantly especially when the entire system is busy. To address this dilemma, soft binding is provided.

- **preferred CPU**: Some CPUs are selected for being preferentially used if their usage is lower than the threshold.
- **allowed CPU**: When the usage of preferred CPUs exceeds the threshold, cores will be selected from allowed CPUs.
- This soft binding solution also applies to containers.

NOTE

- Preferred CPUs are those preferentially used.
- Allowed CPUs are from a list of CPUs bound through **sched_setaffinity** or **cgroup**.

To intuitively observe the execution of soft binding, soft binding scheduling actions are counted. To enable or disable soft binding without stopping the servers, a switch is added for soft binding scheduling.

Parameter Description

- If soft binding is enabled, **/sys/fs/cgroup/cpuacct/sub-cgroup/cpuacct.nr_select_allowed_cpus** is added to a cgroup to collect statistics on how many times allowed cores are selected by tasks in the cgroup.
- If soft binding is enabled, **/sys/fs/cgroup/cpuacct/sub-cgroup/cpuacct.nr_select_prefer_cpus** is added to a cgroup to collect statistics on how many times preferred cores are selected by tasks in the cgroup.
- If soft binding is enabled, **/sys/fs/cgroup/cpuacct/sub-cgroup/cpuacct.nr_smoothed_prefer_cpus** is added to a cgroup to collect statistics on failures of switching from preferred cores to allowed cores due to a smoothing algorithm when tasks in the cgroup are selecting cores.
- **/proc/sys/kernel/sched_dynamic_affinity_disable** is provided to disable soft binding. **0** indicates soft binding is enabled. **1** indicates soft binding is disabled.

How to Use

You can perform the following operations to configure soft binding.

- Procedure
  a. Use **/proc/***$PID***/task/***$TID***/preferred_cpuset** or **cpuset.preferred_cpus** in **/sys/fs/cgroup/cpuset/** to configure the soft binding CPU list for a process or cgroup.
  b. Use **/proc/sys/kernel/sched_util_low_pct** to set the usage threshold for preferred CPUs. If the usage is lower than the threshold, services select

cores from preferred CPUs. Otherwise, they select cores from allowed CPUs. The preferred CPU usage can be calculated in two ways. For details, see the next step.

c. Enable or disable **DA_UTIL_TASKGROUP** in **/sys/kernel/debug/sched_features** to determine whether to select cores based on the preferred CPU usage of a taskgroup or the total preferred CPU usage.

- **preferred_cpuset** can be used to configure the CPU list for soft binding of a thread. A CPU list contains logical CPU IDs, separated by commas (,). For example, **CPU{1,3,5,6,7}** indicates logical CPU IDs 1,3,5,6,7. Consecutive numbers can be abbreviated in range format. For example, 5,6,7 can be abbreviated as 5-7.

  a. **preferred_cpuset** must be a subset of **allowed cpuset**.

  b. If **preferred_cpuset** is not set, left empty, or the same as **allowed cpuset**, soft binding will be invalid.

  ```
  Checking the parameter:
  cat /proc/$PID/task/$TID/preferred_cpuset
  Example of assigning a value to the parameter:
  echo 5-7 > /proc/$PID/task/$PID/preferred_cpuset
  ```

- **cpuset.preferred_cpus** in each directory of a cpuset cgroup can be used to configure soft binding for the cgroup. The parameter format is the same as that of **preferred_cpuset** for a thread.

  a. Currently, **cgroup cpuset.preferred_cpus** must be a subset of **allowed cpus** (**cpuset.cpus**).

  b. If **cpuset.preferred_cpus** is not set, left empty, or the same as **cpuset.cpus**, soft binding will be invalid.

  c. The value of **cpuset.preferred_cpus** in a directory has no impact on that in any other directory.

  ```
  Checking the parameter:
  cat /sys/fs/cgroup/cpuset/cpuset.preferred_cpus
  Example of assigning a value to the parameter:
  echo 5-7 > /sys/fs/cgroup/cpuset/sub-cgroup/cpuset.preferred_cpus
  ```

- The value of **sched_util_low_pct** ranges from **0** to **100** (unit: %). The default value is **85**. **0** indicates no threshold limit, but preset idle preferred CPUs are still preferentially selected. **100** indicates that cores will be selected from **cpuset.cpus** only when the usage exceeds **preferred_cpus**.

  ```
  Checking the parameter:
  cat /proc/sys/kernel/sched_util_low_pct
  Example of assigning a value to the parameter:
  echo 90 > /proc/sys/kernel/sched_util_low_pct
  ```

- **DA_UTIL_TASKGROUP** is added to **/sys/kernel/debug/sched_features**. By default, **DA_UTIL_TASKGROUP** is enabled. If it is enabled, core selection is determined by checking the usage of **preferred_cpus** for a taskgroup. If it is disabled, core selection is determined by checking the total usage of **preferred_cpus** (preferred CPUs used by non-taskgroup processes are also counted).

  ```
  Enable: echo DA_UTIL_TASKGROUP > /sys/kernel/debug/sched_features
  Disable: echo NO_DA_UTIL_TASKGROUP > /sys/kernel/debug/sched_features
  ```

You can observe soft binding by checking how many times preferred or allowed CPUs are selected. You can also enable or disable soft binding as needed.

- If soft binding is enabled, you can check and reset the parameter.

  ```
  Checking the parameters:
  cat /sys/fs/cgroup/cpuacct/sub-cgroup/cpuacct.nr_select_allowed_cpus // Number of times that
  ```



```
allowed cores are selected
cat /sys/fs/cgroup/cpuacct/sub-cgroup/cpuacct.nr_select_prefer_cpus // Number of times that preferred cores are selected
cat /sys/fs/cgroup/cpuacct/sub-cgroup/cpuacct.nr_smoothed_prefer_cpus // Failures of switching from preferred cores to allowed cores due to a smoothing algorithm
Reset the parameter:
echo 0 > /sys/fs/cgroup/cpuacct/sub-cgroup/cpuacct.nr_smoothed_prefer_cpus
```

- You can run **cat** to check **/proc/sys/kernel/sched_dynamic_affinity_disable** or run **echo** to change its value.

```
Checking the parameter: cat /proc/sys/kernel/sched_dynamic_affinity_disable // The value can be 0 or 1. 0 indicates soft binding is enabled and 1 indicates soft binding is disabled.
Assigning a value to the parameter: echo 0 > /proc/sys/kernel/sched_dynamic_affinity_disable
```

- **/proc/*$pid*/task/*$pid*/selected_cpuset** is added to each process to check the CPUs selected by these processes.

NOTE

If CPU subgroups are not configured for cgroup v1 or the target process is not in a CPU subgroup, cores will be selected based on the total usage of **preferred_cpus** regardless of whether **DA_UTIL_TASKGROUP** is enabled or disabled.

Constraints

1. Only **root** user can enable, disable, and configure soft binding. **root** has the highest privilege in the system. When performing operations as **root**, follow the operation guide to avoid system management or security risks caused by improper operations.
2. After **preferred_cpuset** is set, cores are selected only when a process is woken up or periodic load balancing is performed.
3. Soft binding depends on the Per-Entity Load Tracking (PELT) algorithm when selecting preferred CPUs or allowed CPUs. The PELT calculation result is updated every 1 ms. If the load changes frequently, the selection of preferred CPUs or allowed CPUs will also be frequently performed.

11 xGPU

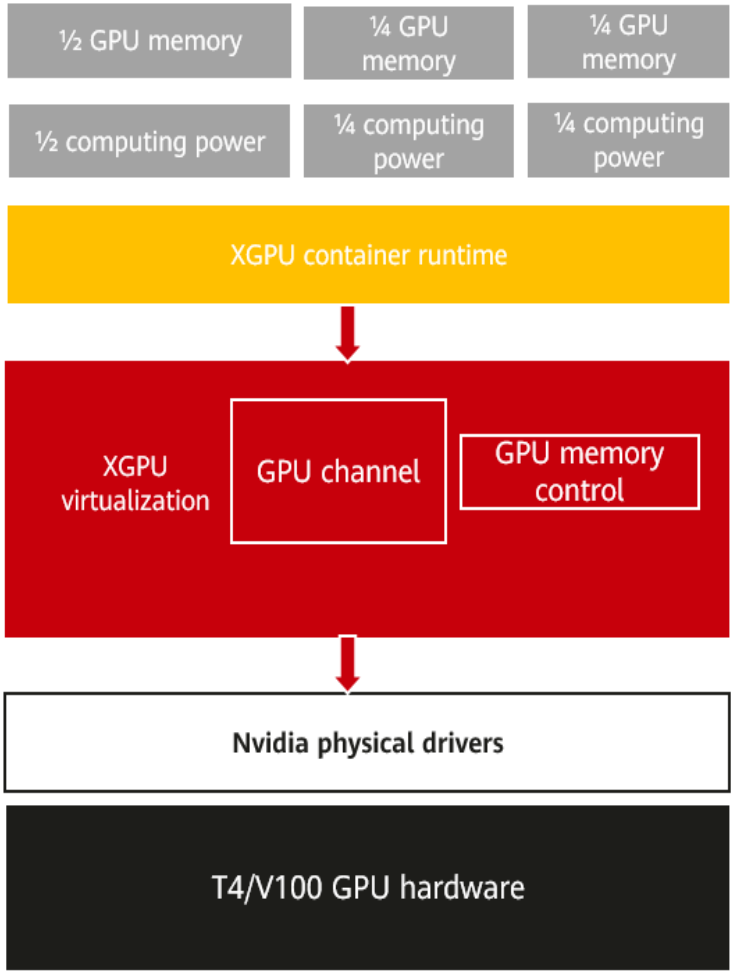
11.1 Overview

xGPU is a GPU virtualization and sharing service developed by Huawei Cloud. This service isolates GPU resources, ensuring service security and helping save costs by improving GPU resource utilization.

Architecture

xGPU uses the in-house kernel drivers to provide vGPUs for containers. This service can isolate the GPU memory and compute while delivering high performance, ensuring that GPU hardware are fully used for training and inference. You can run commands to easily configure vGPUs in a container.

Figure 11-1 xGPU architecture



Why Using xGPU

- Lower cost

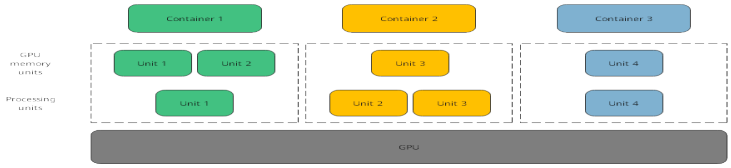
  As GPU technology develops, the price of a single GPU is rising though it can offer larger GPU compute. In some scenarios, an AI application does not require an entire GPU. xGPU enables multiple containers to share one GPU and isolates GPU resources to keep services securely isolated, improving GPU hardware utilization and reducing the resource cost.

- Flexible resource allocation

  xGPU allows you to flexibly allocate physical GPU resources based on your service requirements.

  – You can allocate resources by GPU memory or compute as required.

Figure 11-2 GPU resource allocation



  - You can isolate either of GPU memory or compute, and specify weights to allocate the GPU compute. The GPU compute is allocated at a granularity of 1%. It is recommended that the minimum GPU compute be greater than or equal to 4%.
- Robust compatibility

  xGPU is compatible with open source container technologies such as Docker, Containerd, and Kubernetes.
- Easy of use

  You do not need to recompile your AI applications and replace the Compute Unified Device Architecture (CUDA) library when your services are running.

11.2 Installing and Using xGPU

This section describes how to install and use xGPU.

Constraints

- xGPU is supported only on NVIDIA Tesla T4, V100, and L2.
- The HCE kernel version is 5.10 or later.
- xGPU is supported by CUDA 12.2.0 to 12.8.0.
- NVIDIA driver 535.54.03, 535.216.03, or 570.86.15 has been installed on GPU-accelerated ECSs.
- Docker 18.09.0-300 or later has been installed on GPU-accelerated ECSs.
- Limited by GPU virtualization, compute isolation of xGPUs cannot be used for rendering. You need to use a native scheduling policy for rendering.
- Limited by GPU virtualization, when applications in a container are initialized, the GPU compute monitored by NVIDIA System Management Interface (nvidia-smi) may exceed the upper limit of the GPU compute available for the container.
- When the first CUDA application is created, a percentage of the GPU memory (about 3 MiB on NVIDIA Tesla T4) is requested from each GPU. This GPU memory is considered as the management overhead and is not controlled by the xGPU service.
- xGPU cannot be used on bare metal servers and VMs configured with a passthrough NIC at the same time.

- GPU memory isolation provided by xGPU cannot be used for GPU memory applications through unified virtual memory (UVM) by calling CUDA API cudaMallocManaged(). For more information, see **NVIDIA official document**. Use other methods to request GPU memory. For example, call cudaMalloc().

  NOTE

  - xGPU allows users to disable GPU memory application through UVM. For details, see the description of the **uvm_disable** parameter.

Installing xGPU

To install xGPU, contact technical support.

You are advised to use xGPU through CCE. For details, see **GPU Virtualization**.

Using xGPU

The following table describes the environment variables of xGPU. When creating a container, you can specify the environment variables to configure the GPU compute and memory that a container engine can obtain using xGPU.

Table 11-1 Environment variables that affect xGPU

Environment Variable	Value Type	Description	Example
GPU_IDX	Integer	Specifies the GPU that can be used by a container.	Assigning the first GPU to a container: GPU_IDX=0
GPU_CONTAINER_MEM	Integer	Specifies the size of the GPU memory that will be used by a container, in MiB.	Allocating 5,120 MiB of the GPU memory to a container: GPU_CONTAINER_MEM=5120
GPU_CONTAINER_QUOTA_PERCENT	Integer	Specifies the percentage of the GPU compute that will be allocated to a container from a GPU. The GPU compute is allocated at a granularity of 1%. It is recommended that the minimum GPU compute be greater than or equal to 4%.	Allocating 50% of the GPU compute to a container: GPU_CONTAINER_QUOTA_PERCEN=50

Environment Variable	Value Type	Description	Example
GPU_POLICY	Integer	Specifies the GPU compute isolation policy used by the GPU. • **0**: native scheduling. The GPU compute is not isolated. • **1**: fixed scheduling • **2**: average scheduling • **3**: preemptive scheduling • **4**: weight-based preemptive scheduling • **5**: hybrid scheduling • **6**: weighted scheduling For details, see **GPU Compute Scheduling Examples**.	Setting the GPU compute isolation policy to fixed scheduling: GPU_POLICY=1
GPU_CONTAINER_PRIORITY	Integer	Specifies the priority of a container. • **0**: low priority • **1**: high priority	Creating a high-priority container: GPU_CONTAINER_PRIORITY=1

NVIDIA Docker can use GPUs. It allows NVIDIA GPUs to be used by containers. The following uses NVIDIA Docker to create two containers as an example to describe how to use xGPUs.

Table 11-2 Data planning

Parameter	Container 1	Container 2	Description
GPU_IDX	0	0	Two containers use the first GPU.
GPU_CONTAINER_QUOTA_PERCENT	50	30	Allocate 50% of GPU compute to container 1 and 30% of GPU compute to container 2.
GPU_CONTAINER_MEM	5120	1024	Allocate 5,120 MiB of the GPU memory to container 1 and 1,024 MiB of the GPU memory to container 2.
GPU_POLICY	1	1	Set the first GPU to use fixed scheduling.

Parameter	Container 1	Container 2	Description
GPU_CONTAINER_PRIORITY	1	0	Give container 1 a high priority and container 2 a low priority.

Example:

```
docker run --rm -it --runtime=nvidia -e GPU_CONTAINER_QUOTA_PERCENT=50 -e GPU_CONTAINER_MEM=5120 -e GPU_IDX=0 -e GPU_POLICY=1 -e GPU_CONTAINER_PRIORITY=1 --shm-size 16g -v /mnt/:/mnt nvcr.io/nvidia/tensorrt:19.07-py3 bash
docker run --rm -it --runtime=nvidia -e GPU_CONTAINER_QUOTA_PERCENT=30 -e GPU_CONTAINER_MEM=1024 -e GPU_IDX=0 -e GPU_POLICY=1 -e GPU_CONTAINER_PRIORITY=0 --shm-size 16g -v /mnt/:/mnt nvcr.io/nvidia/tensorrt:19.07-py3 bash
```

Viewing procfs Directory

At runtime, xGPU generates and manages multiple proc file systems (procfs) in the **/proc/xgpu** directory. The following are operations for you to view xGPU information and configure xGPU settings.

1. Run the following command to view the node information:
   ```
   ls /proc/xgpu/
   0 container version uvm_disable
   ```
   The following table describes the information about the procfs nodes.

   Table 11-3 Directory description

Directory	Read/Write Type	Description
0	Read and write	xGPU generates a directory for each GPU on a GPU-accelerated ECS. Each directory uses a number as its name, for example, 0, 1, and 2. In this example, there is only one GPU, and the directory ID is 0.
container	Read and write	xGPU generates a directory for each container running in the GPU-accelerated ECSs.
version	Read-only	xGPU version.
uvm_disable	Read and write	Controls whether to disable the option that allows all containers to request GPU memory through UVM. The default value is **0**. • **0**: This option is enabled. • **1**: This option is disabled.

2. Run the following command to view the items in the directory of the first GPU:
   ```
   ls /proc/xgpu/0/
   max_inst meminfo policy quota utilization_line utilization_rate xgpu1 xgpu2
   ```
   The following table describes the information about the GPU.

Table 11-4 Directory description

Directory	Read/Write Type	Description
max_inst	Read and write	Specifies the maximum number of containers. The value ranges from **1** to **25**. This parameter can be modified only when no container is running.
meminfo	Read-only	Specifies the total available GPU memory.
policy	Read and write	Specifies the GPU compute isolation policy used by the GPU. The default value is **1**. • **0**: native scheduling. The GPU compute is not isolated. • **1**: fixed scheduling • **2**: average scheduling • **3**: preemptive scheduling • **4**: weight-based preemptive scheduling • **5**: hybrid scheduling • **6**: weighted scheduling For details, see **GPU Compute Scheduling Examples**.
quota	Read-only	Specifies the total weight of the GPU compute.
utilization_line	Read and write	Specifies the GPU usage threshold for online containers to suppress offline containers. If the GPU usage exceeds the value of this parameter, online containers completely suppress offline containers. If the GPU usage does not exceed the value of this parameter, online containers partially suppress offline containers.
utilization_rate	Read only	Specifies the GPU usage.
xgpuIndex	Read and write	Specifies the **xgpu** subdirectory of the GPU. In this example, **xgpu1** and **xgpu2** belong to the first GPU.

3. Run the following command to view the directory of a container:

```
ls /proc/xgpu/container/
9418 9582
```

The following table describes the content in the directory.

Table 11-5 Directory description

Directory	Read/Write Type	Description
containerID	Read and write	Specifies the container ID. xGPU generates an ID for each container during container creation.

4. Run the following commands to view the directory of each container:

```
ls /proc/xgpu/container/9418/
xgpu1 uvm_disable
ls /proc/xgpu/container/9582/
xgpu2 uvm_disable
```

The following table describes the content in the directory.

Table 11-6 Directory description

Directory	Read/Write Type	Description
xgpuIndex	Read and write	Specifies the **xgpu** subdirectory of the container. In this example, **xgpu1** is the subdirectory of container 9418, and **xgpu2** is the subdirectory of container 9582.
uvm_disable	Read and write	Controls whether to disable the option that only allows a container to request GPU memory through UVM. The default value is **0**. • **0**: This option is enabled. • **1**: This option is disabled.

5. Run the following command to view the xgpuIndex directory:

```
ls /proc/xgpu/container/9418/xgpu1/
meminfo priority quota
```

The following table describes the content in the directory.

Table 11-7 Directory description

Directory	Read/Write Type	Description
meminfo	Read-only	Specifies the GPU memory allocated using xGPU and the remaining GPU memory. For example, **3308MiB/5120MiB, 64% free** indicates that 5,120 MiB of memory is allocated and 64% is available.

Directory	Read/Write Type	Description
priority	Read and write	Specifies the priority of a container. The default value is **0**. • **0**: low priority • **1**: high priority This parameter is used in hybrid scenarios where there are both online and offline containers. High-priority containers can preempt the GPU compute of low-priority containers.
quota	Read-only	Specifies the percentage of the GPU compute allocated using xGPU. For example, **50** indicates that 50% of the GPU compute is allocated using xGPU.

You can run the following commands to perform operations on GPU-accelerated ECSs. For example, you can change the scheduling policy and modify the weight.

Table 11-8 Example commands

Command	Description
echo 1 > /proc/xgpu/0/policy	Changes the scheduling policy to weight-based preemptive scheduling for the first GPU.
cat /proc/xgpu/container/$containerID/$xgpuIndex/meminfo	Queries the memory allocated to a container.
cat /proc/xgpu/container/$containerID/$xgpuIndex/quota	Queries the GPU compute weight specified for a container.
cat /proc/xgpu/0/quota	Queries the weight of the remaining GPU compute on the first GPU.
echo 20 > /proc/xgpu/0/max_inst	Sets the maximum number of containers allowed on the first GPU to **20**.
echo 1 > /proc/xgpu/container/$containerID/$xgpuIndex/priority	Sets the xGPU in a container to high priority.
echo 40 > /proc/xgpu/0/utilization_line	Sets the suppression threshold of the first GPU to 40%.
echo 1 > /proc/xgpu/container/$containerID/uvm_disable	Controls whether to disable the option that allows containers to use xGPU to request GPU memory through UVM.

Upgrading xGPU

You can perform cold upgrades on xGPU.

1. Run the following command to stop all running containers:
   docker ps -q | xargs -I {} docker stop {}
2. Run the following command to upgrade the RPM package of xGPU:
   rpm -U hce_xgpu

Uninstalling xGPU

1. Run the following command to stop all running containers:
   docker ps -q | xargs -I {} docker stop {}
2. Run the following command to uninstall the RPM package of xGPU:
   rpm -e hce_xgpu

Monitoring Containers Using xgpu-smi

You can use xgpu-smi to view xGPU information, including the container ID, GPU compute usage and allocation, and GPU memory usage and allocation.

The following shows the xgpu-smi monitoring information.

Figure 11-3 xgpu-smi monitoring information

```
+=============================================================================+
|  HUAWEI CLOUD XGPU-SMI                          XGPU Version: 1.0           |
|=============================================================================|
|   Container-Id   |  GPU   |   GPU-Util/Limit   |   GPU-Memory-Usage/Limit   |
+------------------+--------+--------------------+----------------------------+
|     800a09ae74bc |      1 |            0% / 30% |                 0M / 2000M |
|     b8402c9316e4 |      0 |            0% / 10% |                 0M / 6000M |
+------------------+--------+--------------------+----------------------------+
```

You can run the **xgpu-smi -h** command to view the help information about the xgpu-smi tool.

Figure 11-4 xgpu-smi help information

```
[root@localhost ~]#
[root@localhost ~]# xgpu-smi -h
xgpu-smi provides monitoring information about xgpu containers.
The data is presented in either a plain text or a json format, via stdout or a file.

Usage:
  xgpu-smi [flags]
  xgpu-smi [command]

Available Commands:
  clean       Release XGPU
  list        List XGPU containers and used GPUs
  query       Show XGPU containers' information and GPUs' available resource information in JSON format

Flags:
  -c, --container string                filter the specific containers' information
  -f, --filename string                 log to a specified file, rather than to stdout
  -g, --gpu string                      filter the specific GPUs' information
  -h, --help                            help for xgpu-smi
  -r, --remote-runtime-endpoint string   endpoint of remote runtime endpoint (default "unix:///var/run/containerd
/containerd.sock")

Use "xgpu-smi [command] --help" for more information about a command.
[root@localhost ~]#
[root@localhost ~]#
```

11.3 GPU Compute Scheduling Examples

When you want to create an xGPU device, the xGPU service sets time slices (*X* ms) for each GPU based on the maximum number of containers (**max_inst**) to

allocate the GPU compute to containers. The time slices are represented by processing unit 1, processing unit 2, ..., processing unit *N*. The following describes different scheduling policies, and the maximum number of containers is 20 (**max_inst=20**).

Native Scheduling (policy=0)

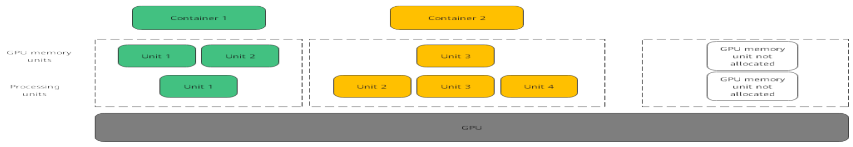
Native scheduling indicates that the GPU compute scheduling method of NVIDIA GPUs. In the native scheduling policy, xGPU is used only for GPU memory isolation.

Figure 11-5 Native scheduling

Fixed Scheduling (policy=1)

Fixed scheduling indicates that the GPU compute is allocated to containers based on a fixed percentage. For example, 5% of the GPU compute is allocated to container 1, and 15% of the GPU compute is allocated to container 2, as shown in the figure.

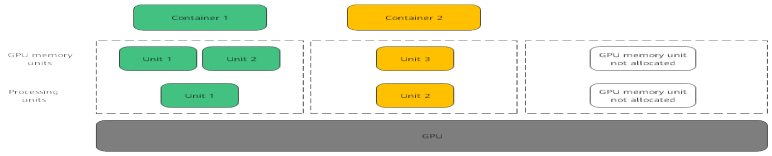
Figure 11-6 Fixed scheduling



Average Scheduling (policy=2)

Average scheduling indicates that each container can have the same percentage of GPU compute (**1/max_inst**). For example, if **max_inst** is set to **20**, each container obtains 5% of the GPU compute, as shown in the following figure.

Figure 11-7 Average scheduling

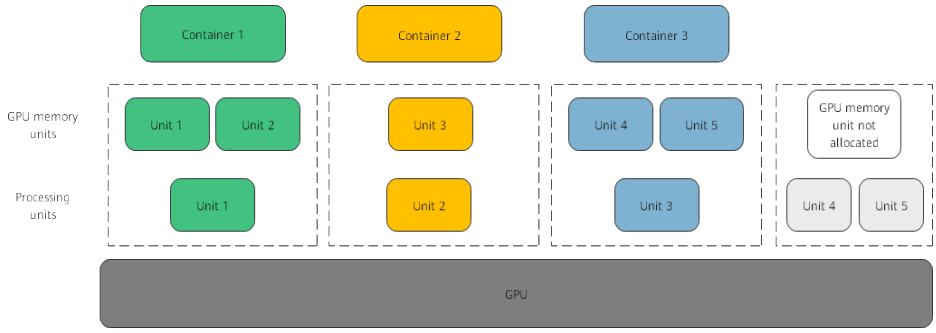


Preemptive Scheduling (policy=3)

Preemptive scheduling indicates that each container obtains one time slice, and xGPU starts scheduling from processing unit 1. However, if a processing unit is not allocated to a container or no process in the container is using the GPU, the processing unit will be skipped, and scheduling will start from the next slice. Processing units in gray are skipped and do not participate in scheduling.

In this example, containers 1, 2, and 3 each occupy 33.33% of the GPU compute.

Figure 11-8 Preemptive scheduling

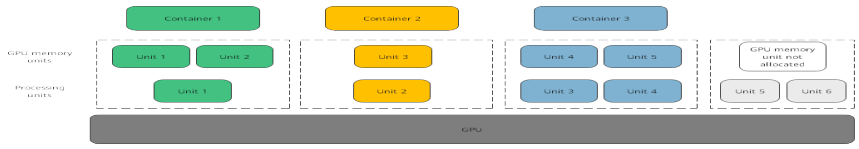


Weight-based Preemptive Scheduling (policy=4)

Weight-based preemptive scheduling indicates that time slices are allocated to containers based on the GPU compute percentage of each container. xGPU starts scheduling from processing unit 1. However, if a processing unit is not allocated to a container, the processing unit will be skipped, and scheduling will start from the next slice. For example, if 5%, 5%, and 10% of the GPU compute is allocated to containers 1, 2, and 3, respectively, container 1 and container 2 each occupy 1 processing unit, and container 3 occupies 2 processing units. Processing units in gray are skipped and do not participate in scheduling.

In this example, containers 1, 2, and 3 occupy 25%, 25%, and 50% of the GPU compute, respectively.

Figure 11-9 Weight-based preemptive scheduling

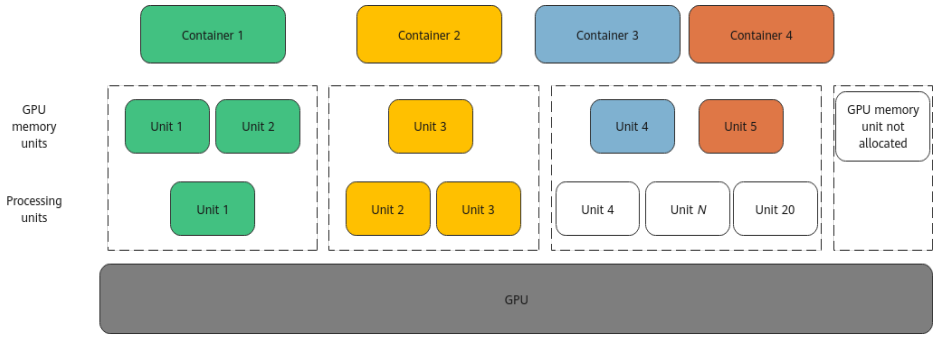


Hybrid Scheduling (policy=5)

Hybrid scheduling indicates that a single GPU supports the isolation of the GPU memory and the isolation of both the GPU compute and GPU memory. The isolation of both the GPU compute and GPU memory is the same as that of the fixed scheduling (policy=1). The containers with only GPU memory isolated share the remaining GPU compute after the GPU compute is allocated to the containers with both GPU compute and memory isolated. If **max_inst** is set to **20**, containers 1 and 2 have both GPU compute and memory isolated, and 5% of the GPU compute is allocated to container 1 and 10% of the GPU compute is allocated to container 2. Containers 3 and 4 have only the GPU memory isolated. Because container 1 occupies one processing unit, container 2 occupies two processing units, containers 3 and 4 share the remaining 17 processing units. In addition, if no processes in container 2 are using the GPU, container 1 occupies one processing unit, container 2 occupies 0 processing units, and containers 3 and 4 share the remaining 19 processing units.

In hybrid scheduling, whether GPU compute isolation is enabled for containers is determined by whether **GPU_CONTAINER_QUOTA_PERCENT** is set to **0**. All containers whose **GPU_CONTAINER_QUOTA_PERCENT** is **0** share the idle GPU compute.

Figure 11-10 Hybrid scheduling



NOTE

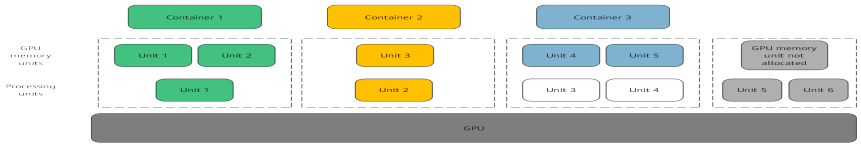
The hybrid scheduling policy is not available for high-priority containers.

Weighted Scheduling (policy=6)

Weighted scheduling indicates that time slices are allocated to containers based on the percentage of the CPU compute of each container. Its compute isolation is not as good as that in weight-based preemptive scheduling. xGPU starts scheduling from processing unit 1. However, if a processing unit is not allocated to a container or no process in the container is using the GPU, the processing unit will be skipped, and scheduling will start from the next slice. For example, if 5%, 5%, and 10% of the GPU compute is allocated to containers 1, 2, and 3, respectively, container 1 and container 2 each occupy 1 processing unit, and container 3 occupies 2 processing units. In the following figure, processing units in white are idle and for container 3, and processing units in both white and gray are skipped and do not participate in scheduling.

In this example, containers 1, 2, and 3 occupy 50%, 50%, and 0% of the GPU compute, respectively.

Figure 11-11 Weighted scheduling



NOTE

Weighted scheduling involves preemption of idle compute. When a container switches between idle and busy statuses, the compute of other containers is affected, and there will be compute fluctuation. When a container switches from the idle state to the busy state, the latency for the container to preempt the compute does not exceed 100 ms.

12 Configuring an HCE Repository

HCE software is managed through RPM packages. By default, an official HCE repository is provided to release and update packages. You can use DNF or YUM commands for software management, such as installation, upgrade, and uninstallation.

Official Repository

By default, an official repository is configured in the **/etc/yum.repos.d/hce.repo** file of an HCE image. Take **HCE 2.0** as an example. The file content is as follows:

```
[base]
name=HCE $releasever base
baseurl=https://repo.huaweicloud.com/hce/$releasever/os/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/os/RPM-GPG-KEY-HCE-2

[updates]
name=HCE $releasever updates
baseurl=https://repo.huaweicloud.com/hce/$releasever/updates/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/updates/RPM-GPG-KEY-HCE-2

[debuginfo]
name=HCE $releasever debuginfo
baseurl=https://repo.huaweicloud.com/hce/$releasever/debuginfo/$basearch/
enabled=0
gpgcheck=1
gpgkey=https://repo.huaweicloud.com/hce/$releasever/debuginfo/RPM-GPG-KEY-HCE-2
```

The fields are described as follows:

- **name**: name of the repository.
- **baseurl**: address of the repository server. The value can be in the format of **http://**, **ftp://**, or **file://**.
- **enabled**: whether to enable the repository. **1** indicates the repository is enabled. **0** indicates the repository is disabled.
- **gpgcheck**: whether to enable GNU Privacy Guard (GPG) verification. **1** indicates GPG verification is enabled. **0** indicates GPG verification is disabled.
- **gpgkey**: address for storing the public key that is used for GPG verification.

CAUTION

Modifying this file may affect software installation and updates. You are not advised to modify this file.

Configuring a Third-Party Repository

To add a third-party repository for HCE 2.0, perform the following steps (OpenEuler is used as an example):

1. Add the **openEuler.repo** file to the **/etc/yum.repos.d/** directory. The file name can be customized, but the file name extension must be .repo. Run **vim /etc/yum.repos.d/openEuler.repo** to edit the file.
2. Set the repository name, for example, **openEuler-everything**. The repository name must be unique and can be changed based on site requirements.
3. Configure the **name** field. For example, set it to **openEuler everything repository**, which is the detailed description of the repository. You can change the description based on site requirements.
4. Set **baseurl** to **https://repo.openeuler.org/openEuler-22.03-LTS/everything/x86_64**, which is the URL for obtaining packages. For details, see the official documentation of OpenEuler. If you are using other third-party repository, see the official documentation of that repository provider.
5. Configure the **gpgcheck** field. **1** indicates GPG verification will be performed on packages to be installed.
6. Configure the **enabled** field. **1** indicates the repository is enabled.
7. Set **gpgkey** to **https://repo.openeuler.org/openEuler-22.03-LTS/everything/x86_64/RPM-GPG-KEY-openEuler**, which is the link where the public key used for GPG verification is obtained.

Add an openEuler update repository in the same way. The final **openEuler.repo** file is as follows:

```
[openEuler-everything]
name=openEuler everything repository
baseurl=https://repo.openeuler.org/openEuler-22.03-LTS/everything/x86_64
gpgcheck=1
enabled=1
priority=3
gpgkey=https://repo.openeuler.org/openEuler-22.03-LTS/everything/x86_64/RPM-GPG-KEY-openEuler
[openEuler-update]
name=openEuler update repository
baseurl=https://repo.openeuler.org/openEuler-22.03-LTS/update/x86_64/
gpgcheck=1
enabled=1
priority=3
gpgkey=https://repo.openeuler.org/openEuler-22.03-LTS/everything/x86_64/RPM-GPG-KEY-openEuler
```

NOTICE

You can use **priority** to set the priority of each repository. To use the default HCE repository preferentially, add **priority=1** to the **hce.repo** configuration file (a smaller value indicates a higher priority). Then, add **priority=2** to the configuration file of a third-party repository. **priority=2** is an example only. You can adjust the value based on site requirements.

NOTE

To upgrade packages, see **Upgrading HCE and RPM Packages**.

Common YUM and DNF Commands

In HCE 1.1, you can only use YUM commands. In HCE 2.0, both YUM and DNF commands can be used. The following are common commands for software management.

Function	YUM	DNF	Example
Installing a package	yum install <*Software package*>	dnf install <*Software package*>	yum install gcc
Uninstalling a package	yum remove <*Software package*>	dnf remove <*Software package*>	yum remove gcc
Listing installed software packages	yum list installed	dnf list installed	yum list installed
Searching for a package	yum search <*Software package*>	dnf search <*Software package*>	yum search gcc
Querying information about a package	yum info <*Software package*>	dnf info <*Software package*>	yum info gcc

13 HCE-specific Kernel Parameters

Compared with CentOS 8, some custom kernel parameters are available for HCE 2.0.

Kernel Parameters

The following parameters are from the files in the **/proc/sys/kernel** and **/sys/kernel** directories.

Task scan

Automatic NUMA balancing scans the address space of a task and cancels page mapping to check whether the page is correctly placed or whether data should be migrated to the memory node local to where the task is running. Each time a scan is delayed, the task scans the next number of pages in its address space. When the end of the address space is reached, the scanner starts from the beginning.

The scan delay and scan size determine the scan rate. When the scan delay decreases, the scan rate increases. The scan delay and the scan rate of every task are adaptive and depend on historical behavior. If pages are properly placed, the scan delay increases. Otherwise, the scan delay decreases. The scan size is not adaptive. However, a larger scan size indicates a higher scan rate.

A higher scan rate results in higher system overhead because page errors must be trapped and data must be migrated. However, the higher the scan rate, the faster the memory of the task is migrated to the local node. If the workload pattern changes, this minimizes performance impact due to remote memory accesses. Parameters in **Table 13-1** control the thresholds for the scan delay and the number of pages scanned.

Table 13-1 Scan parameters

Parameter	Description	Value
kernel.numa_balancing_scan_delay_ms	Specifies the starting scan delay used for a task when it initially forks.	The default value is **1000**, in milliseconds.

Parameter	Description	Value
kernel.numa_balancing_scan_period_max_ms	Specifies the maximum time to scan a task's virtual memory. It effectively controls the minimum scan rate for each task.	The default value is **60000**, in milliseconds.
kernel.numa_balancing_scan_period_min_ms	Specifies the minimum time to scan a task's virtual memory. It effectively controls the maximum scanning rate for each task.	The default value is **1000**, in milliseconds.
kernel.numa_balancing_scan_size_mb	Defines the size of the page to be scanned each time.	The default value is **256**, in MB.

CFS

The Completely Fair Scheduler (CFS) uses nanosecond granularity accounting and does not rely on any jiffies or other HZ detail. Thus, it has no notion of "time slices" in the way the previous scheduler had, and has no heuristic algorithms. There is only one central tunable (you have to switch on **CONFIG_SCHED_DEBUG**):

```
/proc/sys/kernel/sched_min_granularity_ns
```

CFS is not prone to any of the attacks (such as fiftyp.c, thud.c, chew.c, ring-test.c, and massive_intr.c) on heuristic algorithms of conventional schedulers. So, the interactivity will not be affected.

CFS has much stronger handling of nice levels and SCHED_BATCH than the previous vanilla scheduler: both types of workloads are isolated much more aggressively. SMP load balancing has been reworked/sanitized: the runqueue-walking assumptions are gone from the load-balancing code now, and iterators of the scheduling modules are used. As a result, the load balancing code becomes simpler.

Table 13-2 Parameter description

Parameter	Description	Value
kernel.sched_min_granularity_ns	Tunes the scheduler from "desktop" (low latency) to "server" (good batch processing) workloads. The default value is suitable for desktop workloads. SCHED_BATCH is handled by the CFS scheduler module too.	The default value is **3000000**, in nanoseconds.

Fault locating

Table 13-3 Parameter description

Parameter	Description	Value
kernel.net_res_debug_enhance	When a large number of packets are sent and received, the resources of the kernel stack may be insufficient or exceed the thresholds. As a result, the user-mode socket and send interfaces fail to return responses, or packet loss occurs. If this option is enabled, the fault locating information is recorded in the system log. The value **1** indicates this option is enabled, and **0** indicates this option is disabled.	The default value is **0**.

OOM event fault locating

The product/platform service software or OS may have insufficient memory due to a special reason, which triggers an OOM event. Kbox can record the time when an OOM event occurs, details about the OOM process, and system process information in the storage device, facilitating fault locating.

Table 13-4 Information control parameters

Parameter	Description	Value
kernel.oom_enhance_enable	Indicates whether to enable OOM information printing. The value **1** indicates this option is enabled, and **0** indicates this option is disabled.	The default value is **1**.
kernel.oom_print_file_info	Indicates whether to print file system information. The value **1** indicates that the file system information is printed, and **0** indicates that the file system information is not printed.	The default value is **1**.
kernel.oom_show_file_num_in_dir	Specifies the number of files in the printed file system information.	The default value is **10**.

SMT expeller

Table 13-5 Parameter description

Parameter	Description	Value
kernel.qos_offline_wait_interval_ms	Specifies the sleep time (in milliseconds) of the offline task before entering the user mode in the event of overload.	The value ranges from **100** to **1000**, and the default value is **100**.
kernel.qos_overload_detect_period_ms	Specifies a period of time, in milliseconds. If the online task has been occupying a CPU's resources for a period longer than the value specified by this parameter, the process of resolving priority inversion is triggered.	The value ranges from **100** to **100000**, and the default value is **5000**.

The two parameters are new openEuler kernel parameters. For details, see **SMT Expeller Free of Priority Inversion** in the *openEuler Technical White Paper*.

Compute statistics

Due to factors such as turbo frequency, tunning, SMT, and big and small cores on the same node, the CPU usage collected by the cpuacct subsystem cannot reflect the actual compute. The difference between the compute represented by the CPU usage on different nodes can reach over 30%. Compute statistics is used to resolve the problem that the duty cycle cannot reflect the actual CPU compute usage. The cgroup's CPU usage derived based on the actual compute can better represent service performance metrics compared with duty cycle. The impact of SMT is mainly considered in compute statistics.

Table 13-6 Parameter description

Parameter	Description	Value
kernel.normalize_capacity.sched_normalize_util	Controls whether to dynamically enable CPU normalization. The value **0** indicates that this option is disabled, and **1** indicates that this option is enabled.	The default value is **0**.
kernel.normalize_capacity.sched_single_task_factor	In hyper-threading scenarios, set the compute coefficient for independently running logical cores. The value ranges from **1** to **100**. A larger value indicates higher compute of the logical cores.	The default value is **100**.
kernel.normalize_capacity.sched_multi_task_factor	In hyper-threading scenarios, set the compute coefficient for running logical cores in parallel. The value ranges from **1** to **100**. A larger value indicates higher compute of the logical cores.	The default value is **60**.
kernel.normalize_capacity.sched_normalize_adjust	Read and write interface. It controls whether to dynamically enable compute compensation. The value **0** indicates that this option is disabled, and **1** indicates that this option is enabled.	The default value is **0**.

Usage of preferred CPU

Table 13-7 Parameter description

Parameter	Description	Value
kernel.sched_util_low_pct	Specifies the usage threshold of the preferred CPU. When the preferred CPU usage is lower than the threshold, services cannot select cores from the preferred CPU. They can only select cores from the allowed CPU.	The default value is **85**.

Watchdog detection period

In addition to kernel watchdog functions, watchdog invalidity caused by incorrect reset for lockups can be detected. In this case, failure information will be recorded or displayed, helping R&D and O&M personnel quickly locate faults. The **sysctl** parameter is added for the watchdog detection period. The detection and alarm log printing periods can be dynamically adjusted based on product requirements.

Table 13-8 Parameters for configuring the watchdog detection period

Parameter	Description	Value
kernel.watchdog_enhance_enable	Controls all watchdog enhancement functions. **0** indicates that the functions are disabled, and other values indicate that the functions are enabled. The recommended values are **0** and **1**.	The default value is **1**.
kernel.watchdog_softlockup_divide	Adjusts the watchdog detection interval, in seconds. The value can be calculated using the formula: The value of **kernel.watchdog_thresh** × 2/The value of **kernel.watchdog_softlockup_divide**	The value ranges from **1** to **60**, and the default value is **5**.

Parameter	Description	Value
kernel.watchdog_print_period	Specifies the interval (in seconds) for printing process information after the watchdog detects that a process is not scheduled.	The value ranges from **1** to **60**, and the default value is **10**.

CPU QoS interface compatibility

Table 13-9 Parameter description

Parameter	Description	Value
kernel.sched_qos_level_0_has_smt_expell	Whether to enable CPU QoS interface compatibility after the level of priority is increased from 2 to 5 for CCE clusters that use HCE. **0** indicates this option is disabled, which means that there are five levels of priority. When compatibility is required, you can enable this option by setting the value to **1** so that the semantics of priority 0 remains unchanged when there are both online and offline workloads.	The default value is **0**.

Dynamic adjustment of Frequency

Table 13-10 Parameter description

Parameter	Description	Value
kernel.actual_hz	Whether to dynamically adjust the frequency. **0** indicates dynamic adjustment is disabled. The default value 1,000 Hz will be used.	The value ranges from **0** to **1000**, and the default value is **0**.

kernel scheduler

Table 13-11 Parameter description

Parameter	Description	Value
kernel.sched_latency_ns	A variable used to define the initial value for the scheduler period. The scheduler period is a period of time during which all runnable tasks should be allowed to run at least once.	The default value is **24000000**, in nanoseconds.
kernel.sched_migration_cost_ns	Specifies the amount of time after the last execution that a task is considered to be "cache hot" in migration decisions. A "hot" task is less likely to be migrated, so increasing this variable reduces task migrations.	The default value is **500000**, in nanoseconds.
kernel.sched_nr_migrate	If a SCHED_OTHER task spawns a large number of other tasks, they will all run on the same CPU. The migration task or softirq will try to balance these tasks so that they can run on idle CPUs. This option can be set to specify the number of tasks to be moved at a time.	The default value is **32**.
kernel.sched_tunable_scaling	Automatic adjustment of the value of **sched_min_granularity_ns/sched_latency_ns/ sched_wakeup_granularity_ns** based on the number of online CPUs.	**0**: no adjustment **1**: logarithmic adjustment (with 2 as the base) **2**: linear adjustment The default value is **1**.
kernel.sched_wakeup_granularity_ns	Gives preemption granularity when tasks wake up.	The default value is **4000000**, in nanoseconds.

Parameter	Description	Value
kernel.sched_autogroup_enabled	Controls whether to enable **autogroup** scheduling. By default, it is disabled. If enabled, all members in an **autogroup** belong to the same task group of the kernel scheduler. The CFS scheduler uses an algorithm to evenly allocate CPU clock cycles among task groups.	**0**: disabled **1**: enabled Default value: **0**

Dynamic affinity in scheduler

Table 13-12 Parameter description

Parameter	Description	Value
kernel.sched_dynamic_affinity_disable	Controls whether to disable dynamic affinity in scheduler. The value **0** indicates that this option is enabled, and the value **1** indicates that this option is disabled.	The default value is **0**.

CPU QoS priority-based load balancing

Table 13-13 Parameter description

Parameter	Description	Value
kernel.sched_prio_load_balance_enabled	Specifies whether to enable CPU QoS priority-based load balancing. The value **0** indicates that this option is disabled, and **1** indicates that this option is enabled.	The value is **0** or **1**, and the default value is **0**.

For details about new kernel options of openEuler, see **CPU QoS Priority-based Load Balancing** in the *openEuler Technical White Paper*.

Core suspension detection

CPU core suspension is a special issue. When this issue occurs, the CPU core cannot execute any instructions or respond to interrupt requests. So, kernel tests cannot cover this issue. Chips need to use a simulator to locate the root cause. To improve the efficiency of locating faults, core suspension detection is provided for kernels to check whether core suspension occurs.

Table 13-14 Parameter description

Parameter	Description	Value
kernel.corelockup_thresh	If the threshold is set to *x* and a CPU does not receive hrtimer and NMI interrupts for *x* consecutive times, the CPU core will be suspended.	Default value: **5**

Idle polling control

Table 13-15 Parameter description

Parameter	Description	Value
kernel.halt_poll_threshold	If idle pooling is enabled, the guest OS kernel performs idle polling before entering the idle state, without VM-exit. **kernel.halt_poll_threshold** determines the idle polling duration. During idle polling, task scheduling will not increase Inter-Processor Interrupt (IPI) overheads.	Default value: **0** Value range: **0** to max(uint64)

Reset failures in User to Core Environment (UCE)

Table 13-16 Parameter description

Parameter	Description	Value
kernel.machine_check_safe	Ensure that **CONFIG_ARCH_HAS_COPY_MC** is enabled in the kernel. If **/proc/sys/kernel/machine_check_safe** is set to **1**, machine check is enabled. If it is set to **0**, machine check is disabled. Other values are invalid.	Default value: **1** Values: **0** or **1**

Printing source IP addresses upon network packet verification errors

When hardware errors occur or networks are under attack, the kernel receives network packets with verification errors and discards them. As a result, the sources of the data packets cannot be located. This feature provides fault locating in this case. After the verification fails, the source IP addresses of the data packets are printed in system logs.

Table 13-17 Parameter description

Parameter	Description	Value
kernel.net_csum_debug	**1** indicates source IP address printing is enabled and **0** indicates it is disabled. This parameter is only available in Arm.	The default value is **0**.

Cluster scheduling

Table 13-18 Parameter description

Parameter	Description	Value
kernel.sched_cluster	**1** indicates cluster scheduling is enabled and **0** indicates it is disabled.	The default value is **1**.

Parameters in the /proc/sys/net Directory

The following parameters are from the files in the **/proc/sys/net** directory.

TCP socket buffer control

Table 13-19 Parameter description

Parameter	Description	Value
net.ipv4.tcp_rx_skb_cache	Controls the cache of each TCP socket of an SKB, which may help improve the performance of some workloads. This option can be dangerous on systems with a large number of TCP sockets because it increases memory usage.	The default value is **0**, indicating that this option is disabled.

Network namespace control

Table 13-20 Parameter description

Parameter	Description	Value
net.netfilter.nf_namespace_change_enable	If this option is set to **0**, the namespace is read-only in a non-initialized network namespace.	The default value is **0**.

Querying VF information and displaying the broadcast address

Table 13-21 Parameter description

Parameter	Description	Value
net.core.vf_attr_mask	Determines whether to display the broadcast address when netlink is used to query VF link information (for example, the **ip linkshow** command). The default value is **1**, indicating that the broadcast address is displayed, which is the same as that in the community.	The default value is **1**.

Custom TCP retransmission rules

The TCP packet retransmission of HCE complies with the exponential backoff principle. In a low-quality network, the packet arrival rate is low and the latency is high. To address this issue, HCE allows custom TCP retransmission rules by using APIs. Users can specify the number of linear backoff times, maximum number of retransmission times, and maximum retransmission interval, to increase the packet arrival rate and reduce the latency in a low-quality network.

Table 13-22 Parameter description

Parameter	Description	Value
net.ipv4.tcp_sock_retrans_policy_custom	Determines whether to enable custom TCP retransmission rules. **0**: Custom TCP retransmission rules cannot be configured. **1**: Custom TCP retransmission rules can be configured.	The default value is **0**.

IPv6 re-path to avoid congestion

Cloud DCN networks have various paths and use Equal Cost Multi Path (ECMP) to balance loads and reduce conflicts. In the case of dynamic bursts or fluctuant flow sizes, load imbalance and congestion hotspots can still occur.

IPv6 re-path balances network loads through device-side congestion detection and path switching to improve the service flow throughput and reduce transmission latency. The sender side dynamically detects the network congestion status and evaluates whether path switching is necessary. If it is necessary, the sender side performs re-path to switch to a light-load path, thereby avoiding network congestion hotspots.

Table 13-23 Parameter description

Parameter	Description	Value
net.ipv6.tcp_repath_cong_thresh	Number of consecutive congestion rounds allowed during idle hours. If the number is exceeded, path switching will be performed.	Value range: **1** to **8192** Default value: **10**
net.ipv6.tcp_repath_enabled	**0** indicates IPv6 re-path is disabled and **1** indicates it is enabled.	Values: **0** or **1** Default value: **0**

Parameter	Description	Value
net.ipv6.tcp_repath_idle_rehash_rounds	Number of consecutive congestion rounds allowed during non-idle hours. If the number is exceeded, path switching will be performed.	Value range: **3** to **31** Default value: **3**
net.ipv6.tcp_repath_rehash_rounds	Percentage of lost packets allowed in a round. If the threshold is exceeded, congestion occurs in the round.	Value range: **3** to **31** Default value: **3**
net.ipv6.tcp_repath_times_limit	Maximum number of path switching operations per second.	Value range: **1** to **10** Default value: **2**

Network PPS performance tunning

For a large-specification VM (for example, 192 vCPUs) using the HCE 5.10 kernel, the **net.core.high_order_alloc_disable** parameter can be used to improve the network PPS performance when the number of concurrent requests is greater than 8.

Table 13-24 Network PPS performance tunning parameter

Parameter	Description	Temporary Settings	Permanent Settings
net.core.high_order_alloc_disable	This parameter is used to disable high-order page allocation so that network PPS performance can be improved. **0** indicates high-order page allocation is not disabled. **1** indicates high-order page allocation is disabled and order-0 pages will be allocated. Default value: **0**	Run **sysctl -w net.core.high_order_alloc_disable=1**. After a restart, the parameter value will be restored to the default one.	1. Add **net.core.high_order_alloc_disable=1** into the **/etc/sysctl.conf** file. 2. Run **sysctl -p command**. The modification is applied immediately. After a restart, the parameter value remains unchanged.

NOTE

If **net.core.high_order_alloc_disable** is set to **1**, order-0 pages will be allocated. **1** is recommended for high-concurrency TCP and UDP services (the number of concurrent requests is greater than 8) to improve network performance. You can set it to **0** for low-concurrency services. **net.core.high_order_alloc_disable=0** may deteriorate network performance.

Parameters in the /proc/sys/vm Directory

The following parameters are from the files in the **/proc/sys/vm** directory.

Periodic reclamation

Table 13-25 Parameter description

Parameter	Description	Value
vm.cache_reclaim_s	Specifies the interval for periodically reclaiming memory. When periodic memory reclamation is enabled, the memory is reclaimed at an interval defined by **cache_reclaim_s** (unit: **s**).	The default value is **0**.
vm.cache_reclaim_weight	This is used to speed up page cache reclaim. When periodic memory reclamation is enabled, the amount of memory reclaimed each time can be calculated using the following formula: `reclaim_amount = cache_reclaim_weight × SWAP_CLUSTER_MAX × nr_cpus_node(nid)` • **SWAP_CLUSTER_MAX** is defined in **include/linux/swap.h**. • **nr_cpus_node** is used to obtain the number of CPUs on a node. Workqueue is used to reclaim memory. If the memory reclamation task is time-consuming, subsequent work will be blocked, which may affect time-sensitive work.	The default value is **1**.
vm.cache_reclaim_enable	Specifies whether to enable periodic memory reclamation.	The default value is **1**.

Page cache upper limit

Table 13-26 Parameter description

Parameter	Description	Value
vm.cache_limit_mbytes	Limits the page cache amount, in MB. If the page cache exceeds the limit, the page cache is periodically reclaimed.	The default value is **0**.

Maximum batch size and high watermark

Table 13-27 Parameter description

Parameter	Description	Value
vm.percpu_max_batchsize	Specifies the maximum batch size and high watermark per CPU in each region.	• The default value is **(64 × 1024)/ PAGE_SIZE**. • The maximum value is **(512 × 1024)/ PAGE_SIZE**. • The minimum value is **(64 × 1024)/ PAGE_SIZE**.

Maximum fraction of pages

Table 13-28 Parameter description

Parameter	Description	Value
vm.percpu_pagelist_fraction	Defines the maximum fraction (with high watermark **pcp->high**) of pages in each zone that can be allocated for each per-cpu page list. The minimum value is **8**, which means that up to 1/8th of pages in each zone can be allocated for each per-CPU page list. This entry only changes the value of each hot per-CPU page list. A user can specify a number like 100 to allocate 1/100th of pages in each zone for each per-CPU list. The batch value of each per-CPU page list will be updated accordingly and set to **pcp->high/4**. The upper limit of batch is **(PAGE_SHIFT x 8)**. The initial value is zero. The kernel does not use this value to set the high watermark for each per-CPU page list at startup. If the user writes **0** to this sysctl, it will revert to the default behavior.	The default value is **0**.

Memory priority classification

Table 13-29 Parameter description

Parameter	Description	Value
vm.memcg_qos_enable	Dynamically enables memory priority classification. The value **0** indicates that this option is disabled, and **1** indicates that this option is enabled.	The default value is **0**.

Virtual address cycle for mmap loading

Table 13-30 Parameter description

Parameter	Description	Value
vm.mmap_rnd_mask	This parameter can be used to set any number of bits of the virtual address loaded by the mmap to **0** to control the virtual address period loaded by the mmap.	The default value is **null**.

Hugepage management

Table 13-31 Parameter description

Parameter	Description	Value
vm.hugepage_mig_noalloc	When hugepages are migrated from one NUMA node to another, if the number of available hugepages on the destination NUMA node is insufficient, the system determines whether to allocate new hugepages based on the value of this parameter. Set **1** to forbid new hugepage allocation in hugepage migration when hugepages on the destination node run out. Set **0** to allow hugepage allocation in hugepage migration as usual.	The default value is **0**.
vm.hugepage_nocache_copy	In the x86 architecture, when hugepages are migrated to the NUMA node where Intel AEP is used, this parameter determines the way to copy hugepages. If the value is **1**, the NT instruction is used. If the value is **0**, the native MOV instruction is used.	The default value is **0**.
vm.hugepage_pmem_allocall	During hugepage allocation on the NUMA node where Intel AEP is used, this parameter determines whether to limit the number of hugepages. If the value is **0**, the number of hugepages that can be converted to is limited by the kernel threshold. If the value is **1**, all available memory can be applied as hugepages.	The default value is **0**.

vmemmap memory source

Table 13-32 Parameter description

Parameter	Description	Value
vm.vmemmap_block_from_dram	Controls whether to apply the vmemmap memory from the AEP when the AEP memory is hot added to the system NUMA node. If the value is **1**, the memory is from the DRAM. If the value is **0**, the memory is from the corresponding AEP.	The default value is **0**.

Memory overcommitment

Table 13-33 Parameter description

Parameter	Description	Value
vm.swap_madvised_only	Specifies whether to enable memory overcommitment. **1** indicates this option is enabled. **0** indicates this option is disabled.	The default value is **0**.

QEMU hot replacement

Table 13-34 Parameter description

Parameter	Description	Value
vm.enable_hotreplace	Specifies whether to enable QEMU hot replacement. This option supports quick QEMU version upgrade without interrupting services. It can be used in the host OS hot patch and cannot be enabled for the guest OS. The value can be **0** or **1**.	The default value is **0**, indicating that this option is disabled.

Slab allocation

Slab allocation is used to cache kernel data to reduce memory fragmentation and improve system performance. However, as the process progresses, the slabs may occupy a large amount of memory. Enabling **drop_slabs** can release the cache to increase the available memory of the host.

Table 13-35 Parameter description

Parameter	Description	Value
vm.drop_slabs	**0** indicates that the cache will not be released. **1** indicates that the cache will be released.	The value can be **0** or **1**. The default value is **1**.
vm.drop_slabs_limit	Priority. A larger value indicates that fewer slabs will be released. This is to prevent the CPU from being occupied for a long time when too many slabs are being released.	The value is an integer from **0** to **12**. The default value is **7**.

cgroup isolation for ZRAM memory compression

This feature binds memcgs and ZRAM devices. A specified memcg can use a specified ZRAM device, and the memory used by the ZRAM device is obtained from the memory of the container in the group.

Table 13-36 Parameter description

Parameter	Description	Value
vm.memcg_swap_qos_enable	Read and write interface. The default value is **0**, indicating that the feature is disabled. If the value is **1**, **memory.swapfile** of all memcgs is set to **all**. If the value is **2**, **memory.swapfile** of all memcgs is set to **none**. If the value is **1** or **2** and you want to change it to another value, you need to set the value to **0**.	Values: **0**, **1**, or **2** Default value: **0**

Disabling swappiness globally

In the community kernel, if swappiness is set to **0**, the kernel avoids swappiness as much as possible The **swap_extension** option is provided to disable swappiness globally to forcibly prevent anonymous pages from being swapped out.

Table 13-37 Parameter description

Parameter	Description	Value
vm.swap_extension	Disables swappiness globally to forcibly prevent anonymous pages from being swapped out. Set **swap_extension** to **1** to disable anonymous page reclamation. By default, anonymous pages are not swapped out unless the process proactively calls madvise to use the swap space. The anonymous page exchange of processes in the cgroup is not affected. If **swap_extension** is set to **2**, the swapcache is cleared when the swap space is used up, which is irrelevant to this feature. If **swap_extension** is set to **3**, disabling anonymous page reclamation and clearing swapcache are enabled at the same time.	Values: **0**, **1**, **2**, or **3** Default value: **0**

kernel memory watermarks

Table 13-38 Parameter description

Parameter	Description	Value
vm.lowmem_reserve_dma_ratio	If the GFP flag (such as **GFP_DMA**) is not used to express how that memory should be allocated, when the memory in **ZONE_HIGHMEM** is insufficient, the memory can be allocated from the low end **ZONE_NORMAL**. When the memory in **ZONE_NORMAL** is insufficient, the memory can be allocated from the low end **ZONE_DMA32**. "low end" means that the physical memory address of the zone is smaller. "insufficient" means that the free memory of the current zone is less than the requested memory. Except **ZONE_HIGHMEM**, other zones reserve memory for the zone that has higher physical memory address, and the reserved memory is called lowmem reserve. The default value is **DMA/normal/ HighMem: 256 320**, in pages.	The default value is **0**.

OOM incident management

In some cases, you may hope that OOM does not kill the processes. Instead, the black box can report the incidents and trigger crashes to locate faults.

Table 13-39 Parameter description

Parameter	Description	Value
vm.enable_oom_killer	**1** indicates OOM incident management is enabled and **0** indicates it is disabled.	Default value: **0**

Memory UCE error collection and reporting

After this feature is enabled, the system collects error information and sends an alarm to the alarm forwarding service when a memory UCE error occurs. Then, the programs that subscribe to the memory UCE fault event can receive the alarm and handle it. This way, memory UCE errors can be detected in real time and handled in a timely manner.

Table 13-40 Parameter description

Parameter	Description	Value
vm.uce_handle_event_enable	Whether to enable the kernel memory to report UCE errors. **0** indicates disabled and other values indicate enabled. This parameter is only available in Arm.	The default value is **0**.

Parameters in the /proc/sys/mce Directory

The following parameters are from the files in the **/proc/sys/mce** directory.

UCE mechanism enhancement

Table 13-41 Parameter description

Parameter	Description	Value
mce.mce_kernel_recover	Controls whether to enable the kernel UCE mechanism enhancement. The value **1** indicates this option is enabled. You can run the following command to disable this option: `echo 0 > /proc/sys/mce/mce_kernel_recover`	The default value is **1**.

Parameters in the /proc/sys/debug Directory

The following parameters are from the files in the **/proc/sys/debug** directory.

System exception notification

When an oops occurs, the kernel enters the die process and panic process. In this case, the callback functions registered with the die and panic notification chains are called. If the callback functions cause the kernel oops, the kernel enters the oops process. When the callback functions are called again in the oops process, an oops occurs again, and there is a nested oops. As a result, the system is suspended.

To enhance system fault locating and reliability, the system exception notification chain is introduced. When a nested oops occurs in the panic or die notification chain registered by a user, error logs are printed, and the crash process is executed to reset the system.

Table 13-42 System exception parameters

Parameter	Description	Value
debug.nest_oops_enhance	Controls whether to enable the nested oops enhancement interface of the panic notification chain.	The default value is **1**, indicating that this option is enabled.
debug.nest_panic_enhance	Controls whether to enable the nested oops enhancement interface of the die notification chain.	The default value is **1**, indicating that this option is enabled.

14 HCE-specific System Startup Parameters

Compared with CentOS 8, HCE has some custom system startup parameters.

nohz

In versions earlier than Linux kernel 2.6.17, the Linux kernel sets a periodic clock interrupt for each CPU. The kernel processes some cron jobs during the interruption, such as thread scheduling. As a result, a lot of clock interrupts are generated even if a CPU does not need a timer, causing resource wastes. **nohz** is introduced into Linux kernel 2.6.17. It makes clock interrupts configurable to reduce clock interrupts on idle CPUs.

nohz can help improve CPU energy efficiency but it may have a negative impact on load balancing. Enabling **nohz** in some cases will deteriorate performance. So, you are advised to disable **nohz** by default. You can add **nohz=off** to system startup parameters and restart the system to disable it.

CAUTION

nohz has a positive impact on performance in some special cases, for example, when multiple threads concurrently read **/proc/cpuinfo**.

mitigations

In January 2018, Google Project Zero disclosed that modern processors have security vulnerabilities Spectre and Meltdown, which exist in most mainstream processors (including Intel, AMD, and Arm architectures). These vulnerabilities were fixed in all mainstream operating systems, for example, Linux. **mitigations** is used to control whether to enable fixing these CPU vulnerabilities.

Vulnerability fixing depends on speculative execution and out-of-order execution features of processor hardware. These features are critical for improving the performance of modern processors. So, CPU vulnerability fixing will deteriorate performance. In some extreme cases, the performance deterioration even exceeds 50%. Additionally, software fixing can only alleviate but cannot solve the

vulnerability problem. You are advised to disable **mitigations** by default. You can add **mitigations=off** to system startup parameters and restart the system to disable it.

cstate

Since the version released in December 2024, HCE 2.0 supported C-states of Intel GNR servers. C-states can save power but deteriorate the kernel scheduling performance. C-states are enabled in BIOS by default. You can run **cpupower idle-info** to check whether C-states are enabled. The following figure shows the command output.

```
[                                                    ]# cpupower idle-info
CPUidle driver: intel_idle
CPUidle governor: menu
analyzing CPU 0:

Number of idle states: 3
Available idle states: POLL C1 C1E
POLL:
Flags/Description: CPUIDLE CORE POLL IDLE
Latency: 0
Usage: 27
Duration: 163
C1:
Flags/Description: MWAIT 0x00
Latency: 1
Usage: 66
Duration: 11541
C1E:
Flags/Description: MWAIT 0x01
Latency: 4
Usage: 132338
Duration: 63065904
```

Number of idle states indicate the number of available idle C-states. **Available idle states** indicates the available C-states. If **No idle states** is returned, C-states are disabled. If high performance is required, you can run **cpupower idle-set --disable level** to disable C-states. **level** indicates the C-state level, ranging from **0** to **Number of idle states** – 1. This method does not require a restart. Alternatively, you can add **intel_idle.max_cstate=0** to startup parameters to disable C-states. This method requires a restart.

15 Renaming Network Interfaces

Introduction

In an earlier OS (such as Fedora 13, Ubuntu 15, and CentOS 6), network interfaces are named eth0, eth1, and eth2. During system initialization, the Linux kernel allocates names to network interfaces by combining fixed prefixes and indexes. For example, eth0 indicates the first Ethernet interface detected during system startup. If a new network interface is added, the existing network interface names may change because they may be initialized in a different sequence after the system is restarted.

To solve this problem and make it easier to identify network devices, modern Linux distributions use a consistent network device naming rule and use udev to manage devices in a unified manner based on these rules. There are multiple naming rules available for udev. By default, udev allocates names to network interfaces based on firmware information, topology, and physical device locations. The renaming service for network interfaces is provided by systemd-udevd which can ensure the consistency and stability of network device names.

A Consistent Network Naming Rule

The naming rule is: device type + device location.

Device type

- **en**: Ethernet
- **wl**: WLAN
- **ww**: WWAN

Device location

Table 15-1 Device locations

Format	Description
o<on-board_index_number>	Network interface built in BIOS of the mainboard

Format	Description
s<hot_plug_slot_index_number>[f<function>][d<device_id>]	PCI-E network interface built in BIOS of the mainboard
x<MAC>	MAC address
p<bus>s<slot>[f<function>][d<device_id>]	Independent PCI-E network interface
P<domain_number>]p<bus>s<slot>[f<function>][u<usb_port>][.][c<config>][i<interface>]	USB network interface

Procedure

Rename a network interface. For example, rename ens5 to eth0.

Step 1 Modify the configuration file of the network interface.

- If the configuration file **/etc/sysconfig/network-scripts/ifcfg-ens5** exists, change its name from **ifcfg-ens5** to **ifcfg-eth0**.
  ```
  mv /etc/sysconfig/network-scripts/ifcfg-ens5 /etc/sysconfig/network-scripts/ifcfg-eth0
  ```
  Edit the **ifcfg-eth0** file to change the values of **NAME** and **DEVICE** to the new network interface name **eth0**.
- If the configuration file **/etc/sysconfig/network-scripts/ifcfg-ens5** does not exist, create a configuration file **/etc/sysconfig/network-scripts/ifcfg-eth0** and add the following content to the new file:
  ```
  DEFROUTE=yes
  BOOTPROTO=dhcp
  NAME=eth0
  DEVICE=eth0
  ONBOOT=yes
  ```

Step 2 Modify the GRUB configuration.

Modify **/etc/default/grub** to add **net.ifnames=0** and **biosdevname=0** to **GRUB_CMDLINE_LINUX**. Then, run the following command to reload the GRUB configuration:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Step 3 Create a persistent rule file.

Create a file **70-persistent-net.rules** in **/etc/udev/rules.d/** and add the following content to the file:

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="xx:xx:xx:xx:xx:xx", NAME="eth0"
```

Step 4 Restart the node.

----End

Alternatively, you can use a script to change a network interface name. Enter the original and new network interface names as prompted when the script is executed. After the execution is complete, restart the system. The new name will be applied.


```
#!/bin/bash

function check_networkcard() {
    while true; do
        echo "Enter the original network interface name (for example, ens33):"
        read interface_name
        if [[ -z "$interface_name" ]]; then
            echo "The input cannot be empty."
            continue
        fi
        ifconfig -a | grep -q "$interface_name"
        if [[ $? -eq 0 ]]; then
            break
        else
            echo "Network interface $interface_name cannot be found. Enter another one."
        fi
    done
}
function modify_grub() {
    sed -i 's/resume/net.ifnames=0 biosdevname=0 &/' /etc/sysconfig/grub
    grub2-mkconfig -o /boot/grub2/grub.cf
}
function modify_adaptername() {
    while true; do
        echo "Enter a new network interface name (for example, eth0):"
        read new_interface_name
        if [[ -z "$new_interface_name" ]]; then
            echo "The input cannot be empty."
            continue
        else
            break
        fi
    done
    cd /etc/sysconfig/network-scripts/
    if [[ -e ifcfg-$interface_name ]]; then
        mv ifcfg-$interface_name ifcfg-$new_interface_name
        sed -i "s/$interface_name/$new_interface_name/g" ifcfg-$new_interface_name
    else
       sudo cat <<EOF >/etc/sysconfig/network-scripts/ifcfg-$new_interface_name
DEFROUTE=yes
BOOTPROTO=dhcp
NAME="$new_interface_name"
DEVICE="$new_interface_name"
ONBOOT=yes
EOF
    fi
}
function reload_network() {
    mac=$(ip link show $interface_name | awk '/ether/{print $2}')
    echo "SUBSYSTEM==\"net\", ACTION==\"add\", DRIVERS==\"?*\", ATTR{address}==\"$mac\", NAME=
\"$new_interface_name\"" >> /etc/udev/rules.d/70-persistent-net.rules
}
function clear_variable() {
    unset interface_name
    unset interface_cardname
    unset mac
}
function main() {
    check_networkcard
    modify_adaptername
    modify_grub
    reload_network
    clear_variable
    echo "The network interface has been renamed. Reboot the node to make the new name take effect."
}
main
```


16 Tuning of Transparent Huge Pages

16.1 Overview

Transparent Huge Page (THP) is a mechanism provided by the Linux kernel to optimize memory management.

It automatically merges the default 4 KB small pages into a larger page (usually 2 MB) when conditions are met, to reduce the number of page table entries and improve the Translation Lookaside Buffer (TLB) hit rate. This will reduce memory access overheads and improve the system performance.

16.2 Related Settings

Configuring Whether to Enable THP

You can configure whether to enable Transparent Huge Page (THP).

The configuration file is:

```
/sys/kernel/mm/transparent_hugepage/enabled
```

Options:

- **always**: THP is enabled for all memory allocations whenever possible.
- **madvise**: THP is enabled only when a process explicitly requests huge pages through the madvise() system call.
- **never**: THP is disabled.

Defragmentation

THP requires contiguous physical memory areas. However, memory fragments are generated due to frequent memory allocation and release after a system runs for a long time. Defragmentation is a process where the kernel merges scattered physical pages by means such as page migration to form a large, contiguous memory area for hugepage allocation. Defragmentation may cause system latency to increase, especially when the defragmentation is performed in the foreground.

The configuration file is:

```
/sys/kernel/mm/transparent_hugepage/defrag
```

Options:

- **always**: A thread always waits for the kernel to defragment and allocate hugepages. This may slow the response.
- **defer**: A thread does not wait for defragmentation. Defragmentation is performed asynchronously in the background.
- **madvise**: Defragmentation is only performed for the memory area that explicitly calls madvise().
- **defer+madvise**: A thread does not wait for defragmentation. Defragmentation is performed in the background. Defragmentation is only performed for the memory area that explicitly requests to use hugepages.
- **never**: Defragmentation is not used at all. If the kernel fails to allocate a hugepage, it will fall back to using small pages.

A Background Thread for Merging Pages

khugepaged is a kernel thread. It periodically scans anonymous memory and automatically merges appropriate 4 KB pages into 2 MB hugepages.

The configuration file is stored in:

```
/sys/kernel/mm/transparent_hugepage/khugepaged/
```

Key configuration files:

- **defrag**: whether to enable or disable **khugepaged**.
  - **1**: Enable **khugepaged** to proactively defragment memory.
  - **0**: Disable **khugepaged** from proactively defragmenting memory.
- **alloc_sleep_millisecs**: retry interval (unit: ms). This parameter specifies the interval between two **khugepaged** attempts to allocate huge pages. The default value is **60000**.
- **pages_to_scan**: number of pages to be scanned. This parameter specifies the number of pages scanned by **khugepaged** each time it is woken up. The default value is **4096**.
- **scan_sleep_millisecs**: scanning interval (unit: ms). This parameter is used to adjust the scanning frequency. The default value is **10000**.

NOTE

If services are sensitive to latency, you can increase the scanning interval or disable **khugepaged**.

16.3 Tuning Suggestions for Common Scenarios

Recommended Settings for Common Scenarios

Enable the **madvise** mode so that huge pages can be used only when necessary.

```
echo madvise > /sys/kernel/mm/transparent_hugepage/enabled
```


Use **defer+madvise** for memory defragmentation in the background only for explicit requests.

```
echo defer+madvise > /sys/kernel/mm/transparent_hugepage/defrag
```

Database or Latency-Sensitive Services

Databases such as MySQL, PostgreSQL, and Redis are sensitive to performance jitter. You are advised to disable THP.

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

To permanently disable THP, add the following information to startup parameters:

```
sudo grubby --args="transparent_hugepage=never" --update-kernel="/boot/vmlinuz-$(uname -r)" sudo reboot
```

Controlling the Activity of khugepaged

If the CPU usage of the khugepaged daemon is high, you can use the following settings to reduce the activity of khugepaged.

Reduce the scanning frequency of khugepaged.

```
echo 30000 > /sys/kernel/mm/transparent_hugepage/khugepaged/scan_sleep_millisecs
```

Increase the retry interval between two allocation attempts.

```
echo 120000 > /sys/kernel/mm/transparent_hugepage/khugepaged/alloc_sleep_millisecs
```

Reduce the number of pages scanned each time.

```
echo 2048 > /sys/kernel/mm/transparent_hugepage/khugepaged/pages_to_scan
```

Disable automatic defragmentation of khugepaged (if necessary).

```
echo 0 > /sys/kernel/mm/transparent_hugepage/khugepaged/defrag
```

16.4 Viewing the THP Usage

View the global THP usage.

```
cat /proc/meminfo | grep AnonHugePages
```

A non-0 value indicates transparent huge pages are used in the system.

Check whether a specific process is using THP.

```
cat /proc/<PID>/smaps | grep AnonHugePages
```

Replace *<PID>* with a process ID to check which memory segments of the process are using huge pages.

17 NetworkManager Selection and Usage Guide

NetworkManager Selection

NetworkManager is a more powerful alternative to the network service. NetworkManager is used as the mainstream network management tool in the new OS. The network service is more suitable for users who are familiar with command lines and parameter settings, while NetworkManager is more suitable for common users or situations that require automatic network connection management. In most Linux distributions, NetworkManager has become the default network management tool.

CAUTION

NetworkManager and network cannot be used at the same time.

How to Use NetworkManager

- Check the NetworkManager status.
  ```
  systemctl status NetworkManager
  ```
- Stop NetworkManager.
  ```
  systemctl stop NetworkManager
  ```
- Restart NetworkManager.
  ```
  systemctl restart NetworkManager
  ```


18 XFS File System

XFS is a high-performance journaling file system. HCE only inherits some of open-source XFS features from the community.

Constraints

1. The formatting options **inobtcount=1**, **rmapbt=1**, and **reflink=1** are not available.

2. The mount options **discard**, **logdev=device**, **noattr2**, and **ikeep** are not available.

3. Only the disk sector size of 512 bytes is available.

4. If **remount** is used to mount an XFS file system, only pairwise combination between the two option sets (**inode32**/**inode64** and **ro**/**rw**) is available.

NOTE

If unavailable parameters are used, unexpected results may occur. For example, if **logdev=device** is used, I/O timeout may occur in some cases.

How to Use

XFS features are from the open-source community. You can use them in the same way as you are using open-source features. For details, see the community documentation.

CAUTION

To use the XFS file system, the kernel version of HCE must be kernel-5.10.0-182.0.0.95.r1941_123.hce2 or later. You can run **uname -r** to check the kernel version.

