Huawei Cloud EulerOS

用户指南

文档版本 08

发布日期 2024-03-30

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 使用概述

您可通过下列方法使用Huawei Cloud EulerOS。

- 首次创建弹性云服务器实例时，推荐使用HCE OS公共镜像。
- 将操作系统切换为HCE OS。

  如果现有的弹性云服务器配置（网卡、磁盘、VPN等配置的类型和数量）都不需要改变，仅需要修改ECS的操作系统镜像，并且您的软件和原操作系统耦合度较低，适配到HCE OS改动较小，建议使用系统切换，可快速切换到HCE OS。
- 将操作系统迁移为HCE OS。

  如果现有的弹性云服务器配置（网卡、磁盘、VPN等配置的类型和数量）都不需要改变，操作系统软件的配置参数希望保留，可以通过操作系统迁移的方式迁移到HCE OS。

  说明

  仅支持迁移至Huawei Cloud EulerOS 2.0标准版和Huawei Cloud EulerOS 1.1CentOS兼容版，不支持迁移至其他HCE OS镜像版本。

表 1-1 系统切换和迁移的区别

区别	系统切换	系统迁移
数据备份	• 切换操作系统会清除系统盘数据，包括系统盘上的系统分区和所有其它分区。 • 切换操作系统不影响数据盘数据。	• 迁移操作系统不会清除系统盘数据，为避免系统软件的数据丢失，建议将其备份。 • 迁移操作系统不影响数据盘数据。
个性化设置	切换操作系统后，当前操作系统内的个性化设置（如DNS、主机名等）将被重置，需重新配置。	迁移操作系统后，当前操作系统内的个性化设置（如DNS、主机名等）不需重新配置。

2 新创建弹性云服务器时选择 HCE OS

操作步骤

1. 登录控制台，进入**购买弹性云服务器**页面。
2. 选择HCE OS公共镜像。

   在基础配置环节选择公共镜像时，选择Huawei Cloud EulerOS操作系统和具体的镜像版本。购买弹性云服务器完整的操作步骤详见**购买ECS**。

说明

HCE OS公共镜像当前仅支持部分实例，详见**支持的实例规格**。

3 将操作系统切换为 HCE OS

约束与限制

- “包年/包月”方式购买的弹性云服务器切换操作系统时，由于所选镜像不同，当前云服务器的系统盘容量可能不足，不支持切换后的镜像使用。此时，需先卸载系统盘并进行扩容，然后再重新切换操作系统。
- 云硬盘的配额需大于0。
- 不支持BIOS启动方式与UEFI启动方式的操作系统互相切换。

切换须知

- 切换操作系统后，将不再保留原操作系统，并删除原有系统盘及清除系统盘数据，包括系统盘上的系统分区和所有其它分区，请做好数据备份。详细内容，请参考**备份弹性云服务器。**
- 切换操作系统不影响数据盘数据。
- 切换操作系统后IP地址和MAC地址不发生改变。
- 切换操作系统成功后会自动开机。
- 切换操作系统后不支持更换系统盘的云硬盘类型。
- 切换操作系统后，您的业务运行环境需要在新的系统中重新部署。
- 切换操作系统后，当前操作系统内的个性化设置（如DNS、主机名等）将被重置，需重新配置。

计费规则

- “按需付费”方式购买的弹性云服务器切换操作系统后，由于所选镜像不同，系统盘的容量可能会增大，由此将带来费用的变更。

前提条件

- 待切换操作系统的挂载有系统盘。
- 如果原服务器使用的是密码登录方式，切换操作系统后使用密钥登录方式，请提前创建密钥文件。
- 如果您使用私有镜像切换操作系统请参考**《镜像服务用户指南》**提前完成私有镜像的制作。
  - 如果需要指定云服务器的镜像，请提前使用指定云服务器创建私有镜像。

- 如果需要使用本地的镜像文件，请提前将镜像文件导入并注册为云平台的私有镜像。
- 如果需要使用其他区域的私有镜像，请提前复制镜像。
- 如果需要使用其他账号的私有镜像，请提前完成镜像共享。

操作步骤

1. 登录管理控制台。
2. 单击“ ☰ ”，选择“ > 弹性云服务器”。
3. 在待切换操作系统的弹性云服务器的“操作”列下，单击“更多 > 镜像/磁盘 > 切换操作系统”。

   切换操作系统前请先将云服务器关机，或根据页面提示勾选“立即关机切换操作系统”。
4. 根据需求选择需要更换的规格，包括“镜像类型”和“镜像”。

   说明

   对于“包年/包月”方式购买的，如果系统盘容量小于您选择的待切换镜像的大小，此时，您需要先卸载系统盘，并进行扩容，然后再挂载至原执行切换操作。

   扩容系统盘的操作指导，请参见**“扩容云硬盘”**章节。

   图 3-1 切换操作系统

5. 设置登录方式。

   如果待切换操作系统的是使用密钥登录方式创建的，此时可以更换使用新密钥。
6. 单击“确定”。
7. 在“切换云服务器操作系统”页面，确认切换的操作系统规格无误后，阅读并勾选相关协议或声明，单击“提交申请”。

提交切换操作系统的申请后，的状态变为“切换中”，当该状态消失后，表示切换结束。

说明

切换操作系统过程中，会创建一台临时，切换操作系统结束后会自动删除。

后续处理

- 如果切换操作系统前后都是Linux系统，且数据盘设置了开机自动挂载分区。切换操作系统后，数据盘分区挂载信息会丢失，请更新/etc/fstab配置。
  a. 在/etc/fstab写入切换后的分区信息。
     建议您先备份/etc/fstab文件。
     详细操作请参考**初始化Linux数据盘（fdisk）**，设置开机自动挂载磁盘分区。
  b. 挂载分区。挂载分区后即可开始使用数据盘。
     mount *diskname mountpoint*
  c. 执行以下命令，查看挂载结果。
     df -TH
- 如果操作系统切换失败，公有云平台支持重试功能，用户可重新执行**2**-**7**，切换操作系统。
- 重试后，如果仍未成功，可联系客服进行人工恢复。

4 系统迁移

4.1 x2hce-ca 应用兼容性评估

4.1.1 工具概述

x2hce-ca是华为云对系统迁移提供的一款免费的应用兼容性评估工具。x2hce-ca通过对待迁移应用进行快速扫描分析，帮助您评估应用在源操作系统和目标操作系统的兼容性。

表 4-1 支持兼容性评估的 x86 公共镜像

OS发行系列	源操作系统	目标操作系统
HCE OS	64bit：Huawei Cloud EulerOS：1.1	Huawei Cloud EulerOSS 2.0 标准版 64位
EulerOS	64bit：EulerOS：2.10/2.9/2.5/2.3/2.2	Huawei Cloud EulerOS 2.0 标准版 64位
CentOS	64bit：CentOS 7：7.9/7.8/7.7/7.6/7.5/7.4/7.3/7.2/7.1/7.0 64bit：CentOS 8：8.3/8.2/8.1/8.0	Huawei Cloud EulerOS 2.0 标准版 64位
	64bit：CentOS 7：7.9/7.6	Huawei Cloud EulerOS 1.1 CentOS兼容版

表 4-2 支持兼容性评估的 ARM 公共镜像

OS发行系列	源操作系统	目标操作系统
EulerOS	64bit：EulerOS：2.10/2.9/2.8/2.3	Huawei Cloud EulerOS 2.0 标准版 64位 ARM版

4.1.2 约束限制

- 由于x2hce-ca工具安装会有额外资源包引入，不建议在业务环境中运行。x2hce-ca工具仅支持在HCE OS 2.0的操作系统进行安装使用。
- x2hce-ca工具支持扫描的文件格式为jar、py、pyc、bin、sh、rpm、ko。其中，只支持扫描源码为C、C++、Java和Python语言的rpm格式文件。
- x2hce-ca工具不支持回滚，任务异常中断后会在/opt/x2hce-ca/目录下产生残留文件，并不影响工具再次使用。异常中断的任务请重新执行。
- 安装和运行x2hce-ca工具的系统参数要求如下所述。

表 4-3 运行 x2hce-ca 工具的操作系统参数要求

硬件类型	说明
架构	x86_64
CPU	双核及以上
内存	系统空闲内存要求8GB及以上
硬盘	20GB及以上

4.1.3 安装 x2hce-ca

1. 确认repo源配置正常。

   请检查默认的/etc/yum.repos.d/hce.repo配置文件中参数是否正确，正确的配置如下。

   ```
   [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/
   ......
   ```

2. 安装x2hce-ca。

   通过**yum install -y x2hce-ca-hce.x86_64**命令安装工具。安装完成后，生成**表 4-4**。

表 4-4 用户相关目录列表

目录	说明
/var/log/x2hce-ca	存放工具日志文件的目录。
/var/log/aparser	存放配置收集器日志文件的目录。
/opt/x2hce-ca/output	报告默认输出目录。
/opt/x2hce-ca/scan	待扫描应用软件包的建议存放目录。
/opt/x2hce-ca/update	配置文件更新目录，用于存放更新包和对应的License文件。
/etc/x2hce-ca/config	存放静态配置文件的目录。
/etc/x2hce-ca/database_2.0.0.630	存放数据库文件的目录。
/usr/local/x2hce-ca	程序文件存放路径。

3. 重启操作系统或者执行命令**alias x2hce-ca="python /usr/local/x2hce-ca/x2hce-ca.pyc"**，使**x2hce-ca**命令生效。

4.1.4 评估软件兼容性

扫描方式

x2hce-ca工具支持两种软件包扫描方式，请明确将要使用的扫描方式和评估的软件包。

- 扫描源操作系统上单个目录下的单个应用软件包。
- 扫描源操作系统上单个目录下的所有应用软件包。

操作步骤

1. 默认登录或切换到root用户下进行工具使用。
2. 使用如下命令对软件包进行兼容性扫描。

   x2hce-ca scan <option> [**-os_name** *源系统名称*] [**-target_os_name** *目标系统名称*]

   <option>有两种设置，对应x2hce-ca所支持的两种扫描方式：

   – Dir_Name/App_Name，扫描单个目录下的单个应用软件包。

     以X86和ARM操作系统架构为例：

     扫描/mnt/路径下的应用软件包NetworkManager-1.18.8-1.el7.x86_64.rpm（**X86**）

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

     扫描/mnt/路径下的应用软件包NetworkManager-1.18.8-1.el7.aarch64.rpm（**ARM**）

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

说明

其中**-arch**的默认值为x86_64。

- -b Dir_Name，扫描单个目录下的所有应用包。

  例如，扫描directory1目录下的所有应用包。

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

  说明

  建议单个目录下放置不超过750个文件，且文件总大小不超过900M，过多的软件包可能会导致工具故障。

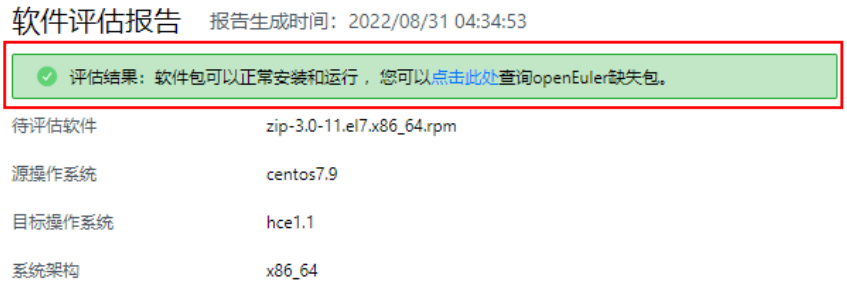
参数	参数类型	说明
-os_name	String	源操作系统。 可选参数，默认参数为centos7.9。 例如设置为-os_name centos8.2，指选择源操作系统为CentOS 8.2。
-target_os_name	String	目标操作系统。 可选参数，默认参数为hce2.0。 例如设置为-target_os_name hce1.1，指选择目标操作系统为Huawei Cloud EulerOS 1.1。

3. 结果分析。

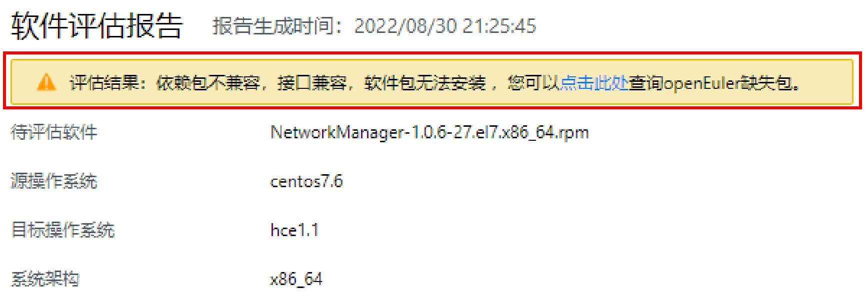
   以扫描/tmp/x2hce-ca_test目录下的三个RPM包为例，命令执行后将有如下输出。

```
[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
083104159.tar.gz
2022-08-31 04:22:05,069 - USER_ID:1001 - INFO - Analyse finished, total 3 items, 3 success, 0 failed
```

   - 软件包的兼容性评估报告保存在/opt/x2hce-ca/output/software/目录下，请自行下载查看具体评估结果。
     - 每个软件包有同名Html格式文件，软件包兼容的评估结果如下。



     不兼容的评估结果如下。



- 详细的依赖包兼容性、接口兼容性等信息可在软件包同名Excel格式文件中查看。

待评估软件	源操作系统	目标操作系统	系统架构	评估项	报告运行时	报告生成时间	直接依赖包数	待确认包	调用外部接口数	待确认接口	接口兼容百分	依赖包兼容百分	源软件包	目标软件包
zip-3.0-11.el7.x86_64.rpm	centos7.9	hce1.1	x86_64	direct dependence, function interface	0.477	20220831043453	2	0	100	0	100%	100%	zip-3.0-11.el7.x86_64.rpm	zip-3.0-11.hce1c.x86_64.rpm
zip-3.0-11.el7.x86_64.rpm	centos7.9	hce1.1	x86_64	direct dependence, function interface	0.477	20220831043453	2	0	100	0	100%	100%	zip-3.0-11.el7.x86_64.rpm	zip-3.0-11.hce1c.x86_64.rpm
zip-3.0-11.el7.x86_64.rpm	centos7.9	hce1.1	x86_64	direct dependence, function interface	0.477	20220831043453	2	0	100	0	100%	100%	zip-3.0-11.el7.x86_64.rpm	zip-3.0-11.hce1c.x86_64.rpm

– 对于扫描失败的应用软件包，请在/opt/x2hce-ca/output/software/目录下查看对应Excel报告。

4.2 将操作系统迁移至 HCE OS 2.0

4.2.1 约束限制

- 当弹性云服务器实例规格和替换的OS系统均在支持的实例规格和支持迁移的公共镜像列表中时，才支持系统迁移。
- 操作系统迁移过程中涉及rpm卸载、安装及更新，操作系统存在异常重启的风险。请在迁移前做好操作系统的系统盘备份，可以通过创建云服务器备份。
- 建议操作系统内存剩余大于128MB，系统盘空间剩余大于5GB（指迁移工具运行需要的系统盘空间，不包含数据备份的空间），boot分区可用空间大于200MB。
- 请避免自定义的RPM包和操作系统组件rpm重名。否则迁移时，自定义的rpm会被迁移工具删除。
- 迁移操作系统后不支持更换系统盘的云硬盘类型。
- 系统迁移过程中，待迁移系统中存在部分冲突包。迁移工具会自动删除冲突包以完成系统迁移。冲突包列表详见冲突包列表。

4.2.2 迁移操作

准备迁移工具依赖的软件包

在系统迁移过程中，迁移工具对特定的基础软件和系统参数存在依赖，本节介绍软件包和系统参数的准备工作。

1. 远程连接待迁移的操作系统。

   根据弹性云服务器控制台操作指导，远程登录到待迁移虚拟机内部，远程登录的具体操作，请参见连接方式概述，并确保虚拟机内部与Internet相通。

2. 检查待迁移系统网络是否能够正常访问HCE OS的repo源，确保迁移工具可以获取到依赖的软件（来自HCE OS的repo源）。

执行命令**curl https://repo.huaweicloud.com/hce/2.0/os/x86_64/**命令检测是否能够访问HCE OS的repo源。若有类似如下输出信息，则能正常访问HCE OS的repo源。

```
% 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. 配置repo源（指原操作系统的repo源），确保迁移工具可以获取到依赖的软件。

   各操作系统的repo源地址不同，请配置正确的repo源地址。

4. 安装依赖的软件包。

   a. 安装python基础软件包。

   ```
   [root@localhost ~]# yum install -y python //任意目录执行安装命令
   ```

   b. （可选）创建软连接。

   说明

   CentOS 8系列及**EulerOS 2.10/2.9**版本需执行以下步骤，其他操作系统版本请忽略。

   i. 安装python3基础软件包。

   ```
   [root@localhost ~]# yum install -y python3 //任意目录执行安装命令
   ```

   ii. 检查是否存在python软链接。

   - 若不存在，继续执行**创建python软链接。**
   - 若已存在，但没有链接至python3，须执行如下命令删除原有python软链接，再执行**创建python软链接。**

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

   - 若已存在，并已链接至python3，请继续执行**安装迁移工具并检查迁移条件**。

   iii. 创建python软链接。

   ```
   [root@localhost]# python
   -bash: /usr/bin/python: No such file or directory   //python软链接不存在的提示信息
   [root@localhost]# cd /usr/bin/      //切换目录至/usr/bin下
   [root@localhost bin]# ln -s python3 python   //创建软件python软链接
   [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.
   >>>
   //通过ctrl+D退出上述界面
   ```

安装迁移工具并检查迁移条件

1. 从**华为云开源镜像站**下载最新版本的迁移工具安装包centos2hce2-*.rpm。

*表示迁移工具版本，本节以centos2hce2-1.0.0-0.0.30.hce2.x86_64.rpm安装包示例。由于迁移工具不断更新，版本号也随之不断更新，操作过程中请适配为实际的安装包名称。

```
[root@localhost test]# wget https://repo.huaweicloud.com/hce/2.0/updates/x86_64/Packages/centos2hce2-1.0.0-0.0.30.hce2.x86_64.rpm  //下载centos2hce2-*.rpm
[root@localhost test]# ls  //检查是否下载成功
centos2hce2-1.0.0-0.0.30.hce2.x86_64.rpm
```

2. 安装迁移工具。

```
[root@localhost test]# rpm -ivh centos2hce2-1.0.0-0.0.30.hce2.x86_64.rpm --nodeps
warning: centos2hce2-1.0.0-0.0.30.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. 配置待迁移系统的系统软件数据的备份路径。

   在系统切换前，迁移工具将自动备份系统软件的所有数据至备份路径。

   执行**vim /etc/centos2hce2.conf**命令，在**centos2hce2.conf**配置文件中配置backup_dir字段，配置备份路径。backup_dir默认为/mnt/sdb/.osbak。

```
# backup dir
backup_dir = "/mnt/sdb/.osbak"    #配置原系统软件数据的备份路径
```

   说明

   - 为避免迁移过程中系统数据的丢失，建议配置备份目录。
   - 在系统迁移时，迁移工具会自动检查备份目录的空间。建议配置单独的数据盘（如/dev/sdb/，并将该分区挂载到/mnt/sdb/），避免因为空间不足导致的检查失败。
   - 请勿将tmpfs类型的文件系统（如/dev、/run等）作为备份目录，系统重启后tmpfs类型文件系统内的文件会丢失。

4. 设置系统迁移参数。

   a. 设置web迁移方式。

      web迁移方式通过下载RPM包集合对系统迁移，因此要求在下载RPM包的过程中不能断网。

      在centos2hce2.conf配置文件中，进行如下设置：

```
[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/"
```

表 4-5 参数说明

参数	说明
default_yum_source	迁移方式，设置为'web'。
web_link_dir	HCE OS的base repo源和updates repo源地址，多个repo源之间需用英文分号隔开。设置为https://repo.huaweicloud.com/hce/2.0/os/x86_64/;https://repo.huaweicloud.com/hce/2.0/updates/x86_64/

b. 配置isclose_modules参数，仅CentOS 8系列需要配置。

CentOS 8系列支持将RPM包集成为module的方式批量安装RPM包。HCE OS不支持此种安装方式。因此系统迁移前，须关闭module功能。

- “yes”表示系统迁移前会自动关闭系统上的modules，默认为“yes”。
- “no”表示系统迁移前不会自动关闭系统上的modules，且若检测到有modules开启时，迁移操作中断。

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

说明

- 执行命令**dnf module list**可查看待迁移系统中所有运行的module。
- 执行命令**dnf module list | grep '\[e\]'**可查看待迁移系统开启的module。

5. 执行**centos2hce2.py --check all**命令，检查当前系统配置是否满足迁移条件。
   - 提示“Enviroment check passed!”时，表示满足迁移条件，可直接执行迁移操作。
   - 提示“call migration failed”时，表示不满足迁移条件，请根据步骤6自动处理相关异常信息。
6. 安装迁移工具依赖的软件。

   执行**centos2hce2.py --install all**命令，迁移工具会先进行备份，接着系统自动安装迁移工具依赖的软件包，并进行迁移前相关预处理操作。

   以下提示表明，已安装依赖的软件包及相关预处理操作，需再次执行步骤5进行环境检查。

   ```
   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
   ```

7. (可选) 重复备份。

   执行**centos2hce2.py --backup force** 命令，迁移工具会根据步骤3中配置的备份路径，对当前系统中的文件进行备份。

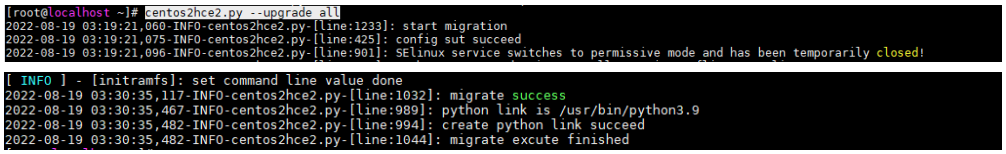
说明

步骤6中安装的工具依赖软件包，在执行此命令之后也会被备份。

迁移系统至 HCE OS

1. 执行迁移命令**centos2hce2.py --upgrade all**进行系统迁移。

   出现migrare sucess提示信息，表明系统迁移成功。迁移后支持回退至原系统，详见操作步骤1。



说明

- 迁移命令不能设置为Linux后台执行方式。
- 可附加--simple_name参数，使得迁移后的grub菜单中显示Huawei Cloud EulerOS的简称。

2. 系统迁移完毕后，执行**reboot**命令（若**reboot**无响应，执行**reboot -f**）使系统完成切换。

   系统重启后，执行**cat /etc/hce-release**命令查看迁移后的操作系统信息，执行**uname -a**命令查看系统内核信息。

   若显示Huawei Cloud EulerOS操作系统，则迁移成功；否则迁移失败，请**联系技术工程师**咨询。

```
[root@localhost ~]# cat /etc/hce-release
Huawei Cloud EulerOS release 2.0 (West Lake)
[root@localhost ~]# uname -a
Linux localhost.localdomain 5.10.0-60.18.0.50.h425_1.hce2.x86_64 #1 SMP Thu Aug 18 16:31:04 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
```

   说明

   操作系统迁移为Huawei Cloud EulerOS后，控制台仍然显示迁移前的操作系统名称。您可手动更新控制台操作系统名称。

3. 清理旧版本组件的文件。

   待迁移系统迁移到HCE OS后，新版本组件替换旧版本组件，但此时旧版本组件的文件仍然保存在系统中。执行命令**centos2hce2.py --precommit upgrade**可清理旧版本组件的文件。

   返回信息中提示“upgrade precommit success”表示环境清理成功。

```
2022-08-19 03:52:32,871-INFO-centos2hce2.py-[line:1112]: remove geolite2-city-20180605-1.el8.noarch succeed
2022-08-19 03:52:32,984-INFO-centos2hce2.py-[line:1112]: remove subscription-manager-rhsm-certificates-1.26.16-1.el8.0.1.x86_64 succeed
2022-08-19 03:52:33,543-INFO-centos2hce2.py-[line:1112]: remove cockpit-ws-211.3-1.el8.x86_64 succeed
2022-08-19 03:52:33,543-INFO-centos2hce2.py-[line:1113]: handle clean rpms finished
2022-08-19 03:52:37,902-INFO-centos2hce2.py-[line:1124]: remove useless link /usr/lib/gcc/x86_64-linux-gnu/10.3.1/32/libasan.a
2022-08-19 03:52:37,906-INFO-centos2hce2.py-[line:1124]: remove useless link /usr/lib/gcc/x86_64-linux-gnu/10.3.1/32/libatomic.a
2022-08-19 03:52:37,910-INFO-centos2hce2.py-[line:1124]: remove useless link /usr/lib/gcc/x86_64-linux-gnu/10.3.1/32/libgomp.so
2022-08-19 03:52:37,915-INFO-centos2hce2.py-[line:1124]: remove useless link /usr/lib/gcc/x86_64-linux-gnu/10.3.1/32/libitm.a
2022-08-19 03:52:37,919-INFO-centos2hce2.py-[line:1124]: remove useless link /usr/lib/gcc/x86_64-linux-gnu/10.3.1/32/libquadmath.a
2022-08-19 03:52:37,923-INFO-centos2hce2.py-[line:1124]: remove useless link /usr/lib/gcc/x86_64-linux-gnu/10.3.1/32/libubsan.a
2022-08-19 03:52:37,927-INFO-centos2hce2.py-[line:1124]: remove useless link /usr/share/doc/e2fsprogs/RELEASE-NOTES
2022-08-19 03:52:37,929-INFO-centos2hce2.py-[line:1126]: remove useless link finished
2022-08-19 03:52:37,930-INFO-centos2hce2.py-[line:1132]: clean system finished and migrate succeed
2022-08-19 03:52:37,930-INFO-centos2hce2.py-[line:1200]: upgrade precommit success
```

   说明

   清理动作可执行多次。

4. （可选）修改Cloud-init相关配置。
   - 若迁移之前的操作系统中存在Cloud-init，服务状态正常，且Cloud-init为rpm包形式，请跳过此步骤。
   - 若迁移之前的操作系统存在Cloud-init服务，服务状态正常，且Cloud-init为某个文件（如CentOS 7系列），非rpm包形式，迁移后请对/etc/cloud/cloud.cfg文件进行如下配置。

   a. 设置开放root密码远程登录并开启root用户的ssh权限。

      设置“disable_root”为“0”不禁用root用户；“ssh_pwauth”为“1”启用密码远程登录；“lock_passwd”为“False”不锁住用户密码。

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

disable_root: 0
ssh_pwauth: 1
```

   b. 执行**/usr/bin/cloud-init init --local**命令，无错误发生，说明Cloud-init配置成功。

      正确安装的Cloud-init会显示Cloud-init的版本详细信息，并且无任何错误信息。

```
[root@localhost ~]# /usr/bin/cloud-init init --local
Cloud-init v. 21.4 running 'init-local' at Fri, 22 Jul 2022 07:43:21 +0000. Up 602150.81 seconds.
[root@localhost ~]#
```

5. （可选）因迁移时会自动关闭selinux服务，如迁移后需启用selinux，执行**centos2hce2.py --precommit upg-selinux**命令。此命令分为两个阶段，每次执行后都需重启系统（若迁移前未开启selinux请忽略此步骤）。

   a. 执行**centos2hce2.py --precommit upg-selinux**命令。

   ```
   [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. 系统重启后，再次执行**centos2hce2.py --precommit upg-selinux**命令。

   ```
   [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.第二次重启后，执行**getenforce**查看selinux状态，Enforcing表明selinux为开启状态。

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

6. （可选）确认迁移完毕后，清理原系统数据。

   迁移操作完成后，原系统的系统数据仍然保留在新系统中，并占用较大内存。建议执行**centos2hce2.py --commit all**命令清理数据。

   执行命令后，系统会自动清理原系统的系统数据，包括步骤3中备份路径下的系统数据。

须知

执行命令后，操作系统无法回退。

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

系统回退

1. 系统回退。

   迁移操作支持系统回退，您可根据需要决定是否回退至原操作系统。

   a. 执行**centos2hce2.py --rollback all**命令进行系统回退。回退后，执行**reboot**命令对系统重启。

```
[root@localhost ~]# centos2hce2.py --rollback all
2022-08-22 04:03:12,107-INFO-centos2hce2.py-[line:1236]: Start rollback
[ INFO ] - [sut]: ================== Pre Rollback Stage ==================
[ INFO ] - [sut]: skip system version check during rollback
[ INFO ] - [sut]: start to do rollback before reboot
[ INFO ] - [sut]: set dracut module for rollback.
[ INFO ] - [sut]: reboot aborted.You need reboot as soon as possbile
[root@localhost ~]# reboot
```

b. 执行**centos2hce2.py --precommit rollback**命令，恢复环境。

```
[root@localhost ~]# centos2hce2.py --precommit rollback
2022-08-22 04:36:13,902-INFO-centos2hce2.py-[line:1239]: precommit migration
2022-08-22 04:36:13,904-INFO-centos2hce2.py-[line:1071]: /opt/migrate//rsync_backup is not exists, skip it
2022-08-22 04:36:13,905-INFO-centos2hce2.py-[line:483]: sut not backup no need rollback
2022-08-22 04:36:17,996-INFO-centos2hce2.py-[line:1194]: rollback precommit success
```

2. （可选）若迁移前已开启selinux，迁移时会自动关闭selinux服务。如有需要，回退后请手动恢复selinux状态。

   a. 执行**centos2hce2.py --precommit rbk-selinux**命令。

   ```
   [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. 执行**reboot**命令，进行系统重启。

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

   c. 系统重启后，可查看到selinux状态为开启状态。

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

3. 清理系统数据。

   执行**centos2hce2.py --commit all**命令清理数据。

   执行命令后，系统会自动清理目标系统和原系统的系统数据，包括步骤3中备份路径下的系统数据。

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

4.2.3 冲突包列表

表 4-6 CentOS 8 系列冲突包列表

CentOS 版本	冲突包列表
CentOS8.0	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版本	冲突包列表
CentOS8.1	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版本	冲突包列表
CentOS8.2	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版本	冲突包列表
CentOS8.3	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版本	冲突包列表
CentOS8.4	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版本	冲突包列表
CentOS8.5	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-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-

CentOS版本	冲突包列表
	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

表 4-7 CentOS 7 系列冲突包列表

CentOS版本	冲突包列表
CentOS7.0	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
CentOS7.1	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
CentOS7.2	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
CentOS7.3	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
CentOS7.4	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
CentOS7.5	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
CentOS7.6	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版本	冲突包列表
CentOS7.7	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
CentOS7.8	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
CentOS7.9	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

表 4-8 HCE OS 冲突包列表

HCE OS	冲突包列表
HCE OS1.1	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

表 4-9 EulerOS 冲突包列表

EulerOS	冲突包列表
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

4.3 将操作系统迁移至 HCE OS 1.1

4.3.1 约束限制

- 对于HCE OS 1.1镜像，仅支持从CentOS7.9迁移到HCE OS 1.1，并且不支持配置图形化界面的CentOS7.9系统的迁移。
- 操作系统迁移过程中涉及rpm卸载、安装及更新，操作系统存在异常重启的风险。请在迁移前做好操作系统的系统盘备份，可以通过**创建云服务器备份**。
- 建议操作系统内存剩余大于128MB，系统盘空间剩余大于1GB。

4.3.2 迁移操作

本节介绍从CentOS7.9迁移到HCE OS 1.1的操作过程。

准备迁移工具依赖的软件包

1. 远程连接待迁移的操作系统。

   根据弹性云服务器控制台操作指导，远程登录到待迁移虚拟机内部，远程登录的具体操作，请参见**连接方式概述**，并确保虚拟机内部与Internet相通。

2. 先关闭CentOS系统/etc/yum.repos.d下的所有的repo配置，确保CentOS的repo源不与HCE OS的repo源发生冲突。

   ```
   [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]#
   ```

   以Centos_Base.repo为例，将里面的每个子项原始的repo源里面添加enabled=0的配置项，如下图所示。

   ```
   [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. 配置HCE OS的repo源。

   将如下内容添加到hce.repo中，并将hce.repo配置文件存放在/etc/yum.repos.d/目录下。

   ```
   [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. 检查CentOS7.9系统网络是否能够正常访问HCE OS的repo源。

   执行命令**curl https://repo.huaweicloud.com/hce/1.1/os/x86_64/**命令检测是否能够访问HCE OS的repo源。若有类似如下输出信息，则能正常访问HCE OS的repo源。

```
% 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. 执行如下命令安装python3。

```
[root@localhost ~]# yum install -y python3 //任意目录执行安装命令
```

   说明

   如果CentOS 7.9系统已经安装python3，请忽略此步骤。

6. 关闭selinux。

   为了保证迁移前后系统配置文件一致，需要关闭selinux。

   a. 修改/etc/selinux/config文件，将config文件中SELINUX的值设置成disabled。

```
SELINUX=disabled
```

   b. 重启操作系统使selinux配置生效。

安装迁移工具

1. 从**华为云开源镜像站**下载迁移工具安装包centos2hce1-*.rpm。

   *表示迁移工具版本，本节以centos2hce1-1.0.0-0.0.2.x86_64.rpm安装包示例。

```
[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  //下载centos2hce1-*.rpm
[root@localhost test]# ls  //检查是否下载成功
centos2hce1-1.0.0-0.0.2.x86_64.rpm
```

2. 安装迁移工具。

   工具安装完成后，系统自动生成/etc/centos2hce1.conf配置文件。

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

3. 配置centos2hce1.conf文件。

   配置HCE OS的repo源地址，用于检测repo源是否能够正常访问，并更新RPM包。


```
#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=
```

说明

centos2hce1.conf配置文件说明详见**附录：conf配置文件说明**。

系统迁移

1. 备份操作系统。

   系统迁移至HCE OS1.1不支持回滚，请备份CentOS整体操作系统（包括系统盘和数据盘）。

2. 执行**centos2hce1.py**命令，进行系统迁移。

   系统迁移的耗时受更新的RPM包数量、大小和从repo源下载速度等影响，一般会在20分钟到1个小时左右完成，具体时间视实际环境确定，执行操作时注意预留足够的时间。

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

   有如下回显信息，表示迁移完成。若迁移失败请使用备份数据恢复。

   ```
   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]# _
   ```

   说明

   CentOS含有某些HCE 1.1不提供的RPM包，执行**centos2hce1.py**命令迁移系统后，迁移工具会自动清除这些RPM包。如果您想保留这些RPM包，请使用**-s skip**参数进行系统迁移。

3. （可选）删除无用的RPM包。

   如下两个RPM包在迁移过程中并没有使用，也不会对系统运行产生任何影响。在此对您可能产生的疑惑进行解释。

   ```
   [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]#
   ```

   ```
   [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. 执行**reboot**命令重启操作系统。

5. 执行**cat /etc/os-release**命令检查是否迁移成功。
   显示如下Huawei Cloud EulerOS信息表示迁移成功。

```
[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. （可选）开启selinux。
   系统迁移前关闭了selinux，请根据需要选择是否开启selinux。
   a. 修改/etc/selinux/config文件，将config文件中SELINUX的值设置成enforcing
      ```
      SELINUX=enforcing
      ```
   b. 重启操作系统使selinux配置生效。

4.3.3 附录：conf 配置文件说明

本节简介conf配置文件重要的字段。

```
#rpm lists for os migration
[rpm_lists]
#origin system must need rpms
baserpms_list = "basesystem initscripts hce-logos plymouth grub2 grubby"  //系统迁移依赖的RPM包

#old rpm and default conflict rpms  //迁移过程中，原系统可能存在的冲突包
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/"    //日志存放路径

# 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/  //基础yum源路径，用于检查网络状态


#iso as yum source
repostr_hce1_1 =     //提供迁移模式的源路径
[base]
name=hceversion
baseurl=https://repo.huaweicloud.com/hce/1.1/os/x86_64/   //基础yum源路径，用于获取RPM包
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 更新 HCE OS 系统和 RPM 包

5.1 升级概述

HCE OS提供操作系统和RPM包的更新维护，包括部署在HCE OS上的RPM包、安全更新涉及的RPM包和漏洞修复。为了操作系统和RPM包的使用更加安全，请及时升级。

HCE OS支持使用dnf/yum命令和OSMT工具两种升级方式。

- Linux自身支持dnf/yum命令，可对RPM包进行升级和回退，升级操作简单。
- OSMT是华为云提供的对HCE OS系统及RPM包升级和回退的工具，可自定义升级范围和定时检查、延迟重启。

两种升级方式区别如下。

表 5-1 升级方式区别

项目	使用dnf或yum命令升级	使用OSMT工具升级
RPM包升级	• 支持无差别升级所有待更新的RPM包，包括安全更新涉及的RPM包和漏洞修复。 • 支持仅升级安全更新涉及的RPM包。	• 支持无差别升级所有待更新的RPM包，包括安全更新涉及的RPM包和漏洞修复。 • 支持自定义升级范围： – 升级不需要重启的RPM包。 – 升级需要重启的RPM包。 – 升级自定义黑白名或白名单列表中的RPM包。 – 升级安全更新涉及的RPM包。 – 漏洞修复。 – 升级新增功能的RPM包。 – 更新新增的RPM包。 • 支持自定义时间自动更新RPM包、延迟重启。

项目	使用dnf或yum命令升级	使用OSMT工具升级
系统版本升级	不支持系统版本升级	支持HCE OS 2.0及以上版本的升级
支持升级的版本	支持HCE OS 1.1及以上版本的RPM包升级。	支持HCE OS 2.0及以上版本的RPM包升级。
回退	支持回退所有历史操作。	系统或RPM包仅支持最近一次升级的回退。

5.2 使用 dnf 或 yum 命令升级

本节介绍HCE OS1.1及以上版本的RPM包升级和回退操作。dnf和yum命令的使用方法相同，本节以dnf命令为例，HCE OS 1.1用yum的相同命令执行。

说明

- Huawei Cloud EulerOS 2.0及之后版本支持yum和dnf命令。
- Huawei Cloud EulerOS 1.1版本仅支持yum命令。

前提条件

HCE OS中已安装dnf组件，dnf命令可用。

```
[root@localhost bin]# dnf
usage: dnf [options] COMMAND

List of Main Commands:

alias                 List or create command aliases
autoremove            remove all unneeded packages that were originally installed as dependencies
check                 check for problems in the packagedb
check-update          check for available package upgrades
clean                 remove cached data
deplist               [deprecated, use repoquery --deplist] List package's dependencies and what packages
provide them
distro-sync           synchronize installed packages to the latest available versions
downgrade             Downgrade a package
......
```

背景信息

yum作为CentOS的包管理器经历了长时间的发展，有一些问题长期未得到解决，包括性能差、内存占用多、依赖解析速度慢等。dnf作为yum的替代者，提供更好的性能。为了保障兼容性，HCE OS依然提供yum命令。

升级步骤

1. 检查待更新的RPM包。
   - 执行**dnf list updates**命令查看所有待更新的RPM包列表。

     ```
     [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
```

- 执行**dnf list updates --security**命令，仅查看安全更新涉及的RPM包。

```
[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. 升级待更新的RPM包。
   - 执行**dnf update**命令升级所有待更新的RPM包，包括安全更新涉及的RPM包和漏洞修复。执行命令输出信息中会显示组件的目标版本信息(Version列）。

```
[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]:
```

   - 执行**dnf update --security**命令，仅升级安全更新涉及的RPM包。

```
[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. 升级成功后，请及时确认业务运行情况。

   升级过程中遇到的常见问题：

   安全规范要求chronyd服务在安装/升级后默认处于disabled状态，所以从HCE-2.0.2206版本升级至新版本后，chronyd服务会处于disabled状态。如有需要，您可通过**systemctl enable chronyd**使能该服务，并通过**systemctl start chronyd**启动该服务。

回退步骤

1. 执行**dnf history**命令，查询需要回退的历史操作ID号。

```
[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. 执行**dnf history undo** <ID号>命令回退此条历史操作。

5.3 使用 OSMT 工具升级

5.3.1 概述

OSMT是华为云提供的对HCE OS系统及RPM包升级和回退的工具。OSMT可自定义配置RPM包的升级范围，并支持周期性定时升级、在指定的时间段单次升级、延时升级并重启等功能。

- **版本升级和回退**：介绍对整体HCE OS系统的升级及回退操作。
- **更新RPM包**：介绍仅对RPM包的升级和回退操作。

说明

OSMT仅支持针对HCE OS 2.0及之后的版本进行升级和回退。该工具会周期访问repo源以获取软件更新信息，从而产生网络流量。您可通过systemctl stop osmt-agent命令停止该服务，并通过systemctl disable osmt-agent命令禁用该服务自启动。

5.3.2 约束限制

- 升级和回退的耗时受更新的RPM包数量、大小和从repo源下载速度等影响，一般会在30分钟内完成，具体时间视实际环境确定，执行操作时注意预留足够的时间。
- OSMT工具仅支持对base、updates两个官方repo源中的RPM包进行升级，请确保这两个源配置的正确性。修改repo源后需要执行**systemctl restart osmt-agent**重启osmt-agent服务。
- 建议通过**osmt config**命令来修改配置文件，使用其他方式修改配置文件，可能导致OSMT功能异常。
- 升级操作必须使用root用户。
- 系统或RPM包的升级回退对剩余空间的要求：
  - 剩余内存至少512M。
  - 根分区剩余空间至少1.5G。

  - 备份内容的存储目录（store_path）剩余空间至少8G。
  - 系统中/boot分区剩余空间至少100M。

    说明

    - 升级范围、目标版本不同，所需存储空间不同。升级时OSMT工具会自动估算升级所需空间，如果剩余空间不足，会给出相应的错误提示。

- 版本升级、回退对selinux状态的影响：
  - 版本升级对selinux状态不产生影响，即版本升级前后，selinux状态一致。
  - 若回退前系统的selinux状态为enforcing，回退后系统自动将selinux状态变为permissive。
  - 若要开启selinux，请手动修改selinux的状态为enforcing并再次重启系统，可恢复到enforcing状态。
  - 若回退前selinux状态为disabled，回退后仍然为disabled状态。此时系统回退对selinux状态不产生影响。
  - 若要开启selinux，需要将selinux的状态先设置为permissive，并在根目录下创建.autorelabel文件并且重启，再将selinux状态修改enforcing并且重启。
- 升级开始前，OSMT工具会对系统健康状态进行评估，以确保升级过程正常。如升级前检查失败，请根据提示信息进行处理。您也可以手动执行检查命令进行检查，详见**OSMT命令帮助信息**。
- OSMT工具升级过程依赖dnf工具，为了确保升级过程的稳定，OSMT工具会将dnf及其依赖RPM包升级至最新。如需回退这些RPM包，可参见**回退步骤**。
- 如果在RPM包更新后，系统配置被修改（**sysctl -a**可查询系统配置），则存在无法使用OSMT工具升级的情况。可用sysctl命令刷新系统配置，**sysctl -p** *<file>*可指定生效的配置文件。**sysctl --system**可应用所有系统目录下的配置文件，如果使用该命令，需要提前确认所有系统目录下的内核配置文件。
- 内核、内核热补丁分别不能超过5个版本。如果其中一个超过5个版本，则在OSMT检查阶段会检查失败，请您卸载不需要的版本后重新进行升级检查。
- 在chrony和ntp共存，并且chrony处于active状态的情况下，OSMT检查阶段会不通过，请您终止chrony服务或卸载chrony、ntp其中一个服务后重新升级。

5.3.3 版本升级和回退

本节介绍对整体HCE OS系统的升级及回退操作。

版本升级或回退时，会同时将RPM包更新到目标系统对应的RPM包版本，和osmt.conf配置文件中的黑白名单rpm列表无关。

版本升级

1. 确认repo源配置正常。

   请检查默认的/etc/yum.repos.d/hce.repo配置文件中参数是否正确，正确的配置如下。

   ```
   [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/
.....
```

说明

错误的配置内容可能会导致OSMT升级失败，或非预期的升级行为。

2. 更新OSMT软件版本。

   OSMT软件版本和HCE OS版本存在配套关系。HCE OS默认安装当前系统的OSMT工具，系统升级时，需要将OSMT更新至目标系统版本对应的OSMT版本。

   执行**dnf update osmt -y --releasever** [*系统目标版本号*] 更新OSMT。例如，将HCE OS 2.0升级到HCE OS 2.1，则执行**dnf update osmt -y --releasever 2.1**命令更新OSMT到最新版本。

   说明

   若误将OSMT删除，执行**dnf install osmt -y --releasever** [*系统目标版本号*] 进行安装。例如，将系统升级至HCE OS 2.1，则执行**dnf install osmt -y --releasever 2.1**命令安装OSMT最新版本。

3. 升级HCE OS系统版本。

   osmt update --releasever [*系统目标版本号*] **--reboot_config** [*重启配置*]

   请根据是否需要立刻重启，选择合适的升级方式。更多的升级选项，详见**osmt update -h**。

   - 将HCE OS 2.0升级到目标版本，如HCE OS 2.1。

     osmt update --releasever 2.1

     升级后，须执行**reboot**命令重启系统，目标系统版本才能生效。
   - 将HCE OS 2.0升级到目标版本，如HCE OS 2.1，并立刻重启。

     osmt update --releasever 2.1 --reboot_config always
   - 将HCE OS 2.0升级到目标版本，如HCE OS 2.1，并指定重启时间，如“2022-12-30 23:00:00”。

     osmt update --releasever 2.1 --reboot_config "2022-12-30 23:00:00"

4. 重启完成后，检查是否升级成功。

   执行**cat /etc/hce-latest**查看**hceversion**字段，若此字段中版本部分是**--releasever**指定的版本号，表示升级成功。

5. （可选）删除升级备份文件。

   确认升级后功能正常后，执行**osmt remove**删除备份文件。

   说明

   请确认升级无异常后再执行**osmt remove**。执行**osmt remove**将删除所有升级备份数据，执行后无法再执行回退。

版本回退

1. 请根据是否需要立刻重启，选择合适的回退方式。
   - 回退至原系统，不立刻重启。

     osmt rollback
   - 回退至原系统并立刻重启。使用此方式，请忽略步骤2。

     osmt rollback --reboot_config always
2. 执行**reboot**命令重启系统。

必须重启系统才能回退到HCE OS的原系统版本。

3. 检查是否回退成功。

   可执行**cat /etc/hce-latest**查看**hceversion**字段，若此字段中版本部分是升级前的版本号，表示已回退成功。

5.3.4 更新 RPM 包

5.3.4.1 准备工作

RPM包的更新方法有两种：使用**osmt update**命令更新和使用后台osmt-agent服务自动更新。此两种方法，都须先执行本节操作。

1. 确认repo源配置正常。

   请检查默认的/etc/yum.repos.d/hce.repo配置文件的参数是否正确，正确的配置如下。

   ```
   [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/
   ......
   ```

   说明

   错误的配置内容可能会导致OSMT升级失败，或非预期的升级行为。

2. 执行**dnf update osmt -y**命令更新OSMT升级工具。
3. 配置/etc/osmt/osmt.conf文件。

   OSMT根据配置文件osmt.conf的设置，对RPM包进行更新。请根据需要配置osmt.conf文件。

   ```
   [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
# 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 = ""
# 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

[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 =

[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"
```

表 5-2 osmt.conf 主要配置项

配置项	说明
[auto]	• auto_upgrade：指定更新RPM包更新方式。默认为False。 – True：使用**osmt update**命令更新和使用后台osmt-agent服务自动更新两种方式都支持。 – False：仅支持使用**osmt update**命令更新RPM包。 • auto_upgrade为True时，配套如下参数。 – cycle_time：检查是否有待更新软件包的周期，单位是秒。默认值为3600秒。 – minimal_interval：指定**osmt update -b**命令参数中开始时间和截止时间的最小时间间隔，单位是秒。默认值为3600秒。 – auto_upgrade_window：配置后台osmt-agent服务自动升级的时间窗，格式为"HH:MM-HH:MM"，表示升级的开始时间和截止时间。 如果截止时间小于开始时间，则表示本次升级时间段跨越自然日。如“22:00-05:00”表示升级时间段为当日22:00到次日凌晨5:00。 – auto_upgrade_interval：指定两次自动升级之间的最小间隔（单位：天）。 • auto_upgrade为False时，仅配套如下参数，执行配置文件时[auto]配置项中其他参数不生效。 – cycle_time：检查是否有待更新软件包的周期，单位是秒。默认值为3600秒。 – minimal_interval：指定**osmt update -b**命令参数中开始时间和截止时间的最小时间间隔，单位是秒。默认值为3600秒。 • motd_setup: 设置登录提示是否开启。默认为True。 – True：开启登录提示。 – False：关闭登录提示，设置后会立刻删除登录提示，并且不会再次生成。如果重新开启，需要使用**osmt update -s**或任意升级命令重新触发生成。

配置项	说明
[Package]	• filters：指定更新的范围，包括types、blacklist、whitelist。例如filters = "blacklist"，指升级时不更新黑名单中的RPM包。 – types：待更新的RPM包类型。 – blacklist：RPM包黑名单列表。 设置黑名单后，不对黑名单中的RPM包进行更新。如果某RPM包依赖黑名单中的包，则不会升级此RPM包。 – whitelist：RPM包白名单列表。 如果不设置白名单或者黑名单列表，默认更新所有可更新的RPM包。 白名单优先级大于黑名单，如果某RPM包同时配置在白名单和黑名单，则会升级此RPM包。 **说明** 如果使用命令指定了黑名单和白名单，以命令指定的黑名单和白名单为准，不再根据配置文件中的黑名单和白名单列表更新RPM包。 • need_reboot_rpms：升级后要重启的RPM包。 后台osmt-agent服务自动升级时，不会更新need_reboot_rpms中的RPM包。若要更新need_reboot_rpms列表中的RPM包，必须使用**osmt update --auto --reboot_config always**或**osmt update --auto --reboot_config “重启时间”**命令更新。 • preinstalled_only：当设置成True时，只升级/etc/osmt/preinstalled.list中的RPM包。
[backup]	• store_path：在该目录下创建备份目录。 升级过程中，OSMT会在store_path下创建.osbak目录，如果原先存在该目录，则需要先使用**osmt remove**将其删除。 • backup_dir：需要备份的目录。其中/etc、/usr、/boot、/var、/run为默认的需要备份的目录，且不可以删减。这些目录会在执行版本升级功能的时候自动进行备份。 • recover_service：OSMT会检查升级前后此列表中服务的启用状态是否一致，如果服务的启用状态被修改，OSMT会尝试恢复此服务的状态。 **说明** [backup]中的路径需要配置为绝对路径的形式。
[cmdline]	cmdline_value：指定升级之后系统的启动项。请配置正确的启动项，确保系统能够正常启动。默认值为HCE的默认启动项。
[conflict]	conflict_rpm：指定升级过程中冲突的软件包，升级时系统将自动删除冲突的软件包。

说明

其他配置项不建议修改，详情请参见**/etc/osmt/osmt.conf配置文件说明**。

5.3.4.2 osmt update 命令更新

手动更新RPM包有两种方式。

- 根据配置文件中的**filters**字段更新RPM包。

  osmt update --auto --reboot_config [*重启配置*]

  表 5-3 重启配置参数说明

参数	说明
never	若未指定重启配置参数，或指定为never时，更新结束后，不重启。 此种方式下，**need_reboot_rpms**列表中的RPM包不会被升级。这种情况下，若要升级**need_reboot_rpms**列表中的RPM包，请将**filters**字段配置为"whitelist"且将对应的RPM包配置到"whitelist"选项中，操作命令如下。在这种情况下，升级完成后，请手动重启系统，使RPM包更新生效。 **osmt config -k filters -v "whitelist"** **osmt config -k whitelist -v** "*rpm1, rpm2, rpm3*"
always	更新RPM包后（包括**need_reboot_rpms**列表中的RPM包），立即自动重启。
<specific time>	更新RPM包后（包括**need_reboot_rpms**列表中的RPM包），在指定的重启时间自动重启。时间格式如 "2020-02-02 2:02:02"。

- 使用黑名单和白名单的方式更新RPM包。

  osmt update --pkgs [*rpm1 rpm2 rpm3 ...*] **--exclude_pkgs** [*rpm4 rpm5 rpm6 ...*] --**reboot_config** [*重启配置*]

  – --pkgs：可选，指待更新的白名单列表，RPM包以空格分隔。

    例如，执行如下命令更新白名单中的hce-logos、hce-lsb、tomcat包。

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

  – --exclude_pkgs：可选，指待更新的黑名单列表，RPM包以空格分隔。

    例如，执行如下命令不更新黑名单中的ongres-scram、llvm-static包。

    osmt update --exclude_pkgs ongres-scram llvm-static

  – --reboot_config [*重启配置*]：可选，指定重启方式，包含always、never、重启时间。

    - always：更新结束后，若有重启才能生效的RPM包，则立即自动重启；若没有，则不重启。
    - never ： 更新结束后，不重启。
    - <specific time>：指定重启时间。更新结束后，若有重启才能生效的RPM包，则在指定时间自动重启；若没有，则不重启。时间格式如 "2020-02-02 2:02:02"。

说明

- 如果使用黑名单和白名单的更新方式，--pkgs和--exclude_pkgs至少使用一个。
- 如果使用命令指定了黑名单和白名单，以命令指定的黑名单和白名单为准，不再根据配置文件中的黑名单和白名单列表更新RPM包。

5.3.4.3 osmt-agent 服务自动更新

osmt-agent服务支持周期性检查是否有待更新的RPM包，并自动更新RPM包。检查的周期和执行更新的时间段可以自定义设置。

1. 执行以下命令，确保osmt.conf文件auto_upgrade字段为True。

   osmt config -k auto_upgrade -v True

2. 执行**systemctl status osmt-agent.service**命令确认osmt-agent服务是否正常开启。
   - Active为active (running)状态，表示osmt-agent正常开启。
   - 如果osmt-agent没有处于active (running)状态，请执行**systemctl start osmt-agent.service**命令启动osmt-agent。

   ```
   ● osmt-agent.service - osmt-agent - The agent that manages HCE OS.
        Loaded: loaded (/usr/lib/systemd/system/osmt-agent.service; enabled; vendor preset: disabled)
        Active: active (running) since Sat 2022-12-24 18:32:42 CST; 2 days ago
      Main PID: 1421 (python)
         Tasks: 3 (limit: 21113)
        Memory: 72.7M
        CGroup: /system.slice/osmt-agent.service
                ├─  1421 python /usr/bin/osmt_server
                ├─  1474 /bin/bash /usr/bin/osmt-agent
                └─ 32445 sleep 3600
   ```

3. 执行如下命令设置自动更新RPM包的时间段与升级周期。
   - 指定可自动升级的时间段

     命令格式：osmt config -k auto_upgrade_window -v "*auto_upgrade_window*"

     auto_upgrade_window：配置后台osmt-agent服务自动升级的时间窗，格式为"HH:MM-HH:MM"，表示升级的开始时间和截止时间。

     如果截止时间小于开始时间，则表示本次升级时间段跨越自然日。如“22:00-05:00”表示升级时间段为当日22:00到次日凌晨5:00。

     例如，配置当日23:00到次日01:00时间段为可升级时间窗为：

     osmt config -k auto_upgrade_window -v "23:00-01:00"
   - 指定检查升级的时间间隔

     命令格式：osmt config -k auto_upgrade_interval -v *auto_upgrade_interval*

     auto_upgrade_interval：指定两次自动升级之间的最小间隔（单位：天）。

     例如，配置每隔1天进行自动升级的命令为：

     osmt config -k auto_upgrade_interval -v 1

5.3.5 升级后续操作

1. 升级成功后，请及时确认业务运行情况。如业务运行正常，请在合适的时候执行**osmt remove**命令删除备份内容。删除后将无法回退本次升级内容。
2. 安全规范要求chronyd服务在安装/升级后默认处于disabled状态，所以从HCE-2.0.2206版本升级至新版本后，chronyd服务会处于disabled状态。如有需

要，您可通过**systemctl enable chronyd**使能该服务，并通过**systemctl start chronyd**启动该服务。

5.3.6 回退 RPM 包

执行**osmt rollback --reboot_config always**命令回退RPM包。仅支持回退最近一次更新的RPM包。

--reboot_config always：可选，若上次升级有need_reboot_rpms列表中的RPM包，请使用此参数。

如果不带--reboot_config always，上次升级有涉及need_reboot_rpms列表中的RPM包，需要手动重启后才能回退生效。

说明

请使用最新版本的OSMT工具进行操作，不建议通过OSMT工具回退OSMT自身版本。

5.4 附录

5.4.1 OSMT 命令帮助信息

- 执行**osmt -h**命令，显示OSMT的帮助信息。

```
[root@localhost SOURCES]# osmt -h
usage: osmt [-h] {update,rollback} ...
positional arguments:
  {update,rollback}
    update           update os version or packages
    rollback         rollback last update
    remove           remove backup files in store path
    config           modify config file by command line
    job              handle upgrade task.
optional arguments:
  -h, --help         show this help message and exit
```

表 5-4 参数说明

参数	说明
update	升级操作系统或RPM包。
rollback	回退操作系统或RPM包。
remove	删除存储路径中的备份文件。
config	查询或修改配置文件。
job	查询或管理升级任务。
-h, --help	可选参数，提供**osmt**命令的帮助信息。

- 执行**osmt update -h**命令，显示升级操作系统或RPM包的帮助信息。

```
[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
```

表 5-5 参数说明

参数	说明
-h, --help	提供**osmt update**命令的帮助信息。
--nosignature	升级时不根据包的签名筛选待更新的RPM包。
-s,--show	显示当前系统可用的升级或更新信息。 ● --all：显示所有待更新的RPM包。 ● --security：显示待更新的安全包。 ● --version：显示所更新到的版本号。
-a,--auto	**指定更新RPM包更新方式**，与-v、-p、-e互斥。
-p,--pkgs	指定需要更新的RPM包白名单列表，与-v、-a互斥。
-e,--exclude_pkgs	指定不需要更新的RPM包黑名单列表，与-v、-a互斥。
-v , --releasever	指定需要升级到的HCE OS版本号。
-r, --reboot_config	指定重启方式。 ● always：更新结束后，若有重启才能生效的RPM包，则立即自动重启；若没有，则不重启。 ● never ： 更新结束后，不重启。 ● <specific time>：指定重启时间。更新结束后，若有重启才能生效的RPM包，则在指定时间自动重启；若没有，则不重启。时间格式如 "2020-02-02 2:02:02"。

参数	说明
-b, --between	指定osmt执行更新的开始时间和结束时间，格式为"HH:MM-HH:MM"。 如果截止时间小于开始时间，则表示本次升级时间段跨越自然日。如“22:00-05:00”表示升级时间段为当日22:00到次日凌晨5:00。
-j, --job	以后台进程方式进行本次升级。
-c, --check	升级前检查系统状态，确认系统是否能升级。该检查操作是可选的，在实际升级时会再次执行升级前检查。 建议在升级命令增加-c参数，对升级命令进行检查。例如，执行**osmt update -a**进行升级时，在升级前执行**osmt update -a -c**提前检查。
-V, --verbose	可选参数，显示详细的升级信息。
-o, --preinstalled-only	可选参数，仅升级/etc/osmt/preinstalled.list中的RPM包列表，该选项仅对版本升级有效。
-t, --retry	可选参数，进行升级重试。
--nocheck	可选参数，升级前不做任何检查，直接进入升级流程。

- 执行**osmt rollback -h**命令，显示回退操作系统或RPM包的帮助信息。

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

表 5-6 参数说明

参数	说明
-h, --help	提供**osmt rollback**命令的帮助信息。
-r, --reboot_config	指定是否允许重启。
-V, --verbose	可选参数，显示详细的过程日志。
-t, --retry	可选参数，进行回退重试。
--nocheck	可选参数，回退前不进行任何检查，直接进入回退阶段。

- 执行**osmt config -h**命令，显示修改配置项或显示配置项的帮助信息。

```
usage: osmt config [-h]  [-k] { minimal_interval,cycle_time…}  [-v VALUE]
optional arguments:
-h, --help            show this help message and exit
-k {minimal_interval,cycle_time…} --key {minimal_interval,cycle_time…}
-v VALUE, --value VALUE
-V, --verbose         show more log to screen
```

表 5-7 参数说明

参数	说明
-h, --help	提供**osmt config**命令的帮助信息。
-k, --key	指定要查询或修改的key值。
-v, --value	指定对应修改key值的value值。
-V, --verbose	可选参数，显示详细的的过程日志。

说明

建议只通过**osmt config**命令来修改配置文件，使用其他方式修改配置文件，可能导致OSMT功能异常。

- 指定**osmt job -h**命令，显示任务管理的帮助信息。

```
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
```

表 5-8 参数说明

参数	说明
-h, --help	提供**osmt job**命令的帮助信息。
-s, --show	显示后台任务信息。
-d DELAY, --delay DELAY	允许将当前等待重启的任务延迟一段时间，具体格式为“1:50:00”，表示将重启时间推迟1小时50分。
-c, --cancel	取消当前job。
-y, --yes	默认用户同意本次操作。
-V, --verbose	可选参数，显示详细的过程日志。

5.4.2 /etc/osmt/osmt.conf 配置文件说明

本节对OSMT工具的配置文件osmt.conf不建议修改的配置项进行说明。

```
[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
# 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"
# 升级后需要重启系统才能生效的rpm包
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
```

表 5-9 osmt.conf 不建议修改的配置项

配置项	说明
types	按照security、 bugfix、enhancement、newpackage、unknown五个配置项指定RPM包的更新范围，不建议修改。如有特殊需要可以根据系统实际情况进行修改。
severity	升级安全更新，不建议修改。默认升级安全更新。如有特殊需要可以根据系统实际情况进行修改。
[resource_needed]	指升级或更新前进行检查的系统资源限制值，不建议修改。如有特殊需要可以根据系统实际情况进行修改。

6 对 HCE OS 进行安全更新

6.1 安全更新概述

本节主要介绍如何使用yum或dnf命令查询并安装Huawei Cloud EulerOS中的安全更新。

各版本对yum和dnf命令的支持情况不同，本节以yum命令为例介绍。

说明

dnf作为yum的替代者，提供更好的性能，dnf和yum命令的使用方法相同。

- Huawei Cloud EulerOS 2.0及之后版本支持yum和dnf命令。
- Huawei Cloud EulerOS 2.0之前版本仅支持yum命令。

6.2 关于通用漏洞披露（CVE）

CVE（Common Vulnerabilities and Exposures）是已公开披露的各种计算机安全漏洞，所发现的每个漏洞都有一个专属的CVE编号。Huawei Cloud EulerOS为保障系统安全性，紧密关注业界发布的CVE信息，并会及时修复系统内各类软件漏洞，增强系统的安全性。您可在Huawei Cloud EulerOS的安全公告中查看安全更新记录。

- **Huawei Cloud EulerOS 1.1安全公告**
- **Huawei Cloud EulerOS 2.0安全公告**

根据CVSS（Common Vulnerability Scoring System）评分，Huawei Cloud EulerOS将安全更新分为四个等级：

- Critical（高风险，必须安装）
- Important（较高风险，强烈建议安装）
- Moderate（中等风险，推荐安装）
- Low（低风险，可选安装）

若您目前已安装Huawei Cloud EulerOS 1.1及以上版本，您可以根据以下操作查询并安装安全更新，修复系统漏洞。本章节以Huawei Cloud EulerOS 2.0为例举例说明。

说明

Huawei Cloud EulerOS系统版本不同，部分显示可能与本文档存在差异，以实际显示为准。

6.3 yum 命令参数

命令格式：yum <command> [option]

表 6-1 command 的主要参数

参数	说明
help	显示帮助信息。
updateinfo	显示软件包更新信息摘要。
upgrade	升级软件包。
check-update	检查可用的软件包更新。

表 6-2 option 的主要参数

参数	说明
-h, --help, --help-cmd	显示命令帮助信息。
--security	显示安全相关的软件包信息。
--advisory ADVISORY	指定ADVISORY相关的软件包，可用于安装、查询或更新。 多个软件包之间使用英文逗号分隔。
--cve CVES	指定CVE相关的软件包，可用于安装、查询或更新。 多个软件包之间使用英文逗号分隔。
--sec-severity {Critical,Important,Moderate,Low}	指定安全更新等级相关的软件包，可用于安装、查询或更新。 {}中的安全更新等级参数可任意组合。

说明

更多详细信息，请使用**yum --help**获取帮助信息。

6.4 查询安全更新

命令格式：yum updateinfo <command> [option]

- 执行**yum updateinfo**命令，查询全部可用的安全更新信息。

```
[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)
```

- <command>的主要参数。
  - list：查询当前可用的安全更新列表。

    ```
    [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>：查询指定SA ID的安全更新详情。

    ```
    [root@localhost ~]# yum updateinfo info HCE2-SA-2022-0029
    Last metadata expiration check: 5:09:15 ago on Tue 13 Sep 2022 09:43:13 AM CST.
    ==================================================================
    An update for python3 is now available for HCE 2.0
    ==================================================================
    Update ID: HCE2-SA-2022-0029
    Type: security
    Updated: 2022-09-08 22:08:34
    CVEs: CVE-2021-28861
    Description: Security Fix(es):
    : Python 3.x through 3.10 has an open redirection vulnerability in lib/http/server.py due to no protection against multiple (/) at the beginning of URI path which may leads to information disclosure. (CVE-2021-28861)
    Severity: Important
    ```

- [option]的主要参数。
  - --sec-severity={Critical,Important,Moderate,Low}：指定安全更新级别，{}中的安全更新等级参数可任意组合。

    例如，使用--sec-severity=Critical查询某个安全更新级别。

    ```
    [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
    ......
    ```

    例如，使用--sec-severity={Critical,Moderate}查询多个安全更新级别。

    ```
    [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>：查询指定的CVE ID。

    ```
    [root@localhost ~]#  yum updateinfo info --cve=CVE-2021-28861
    Last metadata expiration check: 5:10:38 ago on Tue 13 Sep 2022 09:43:13 AM CST.
    ==================================================================
    An update for python3 is now available for HCE 2.0
    ==================================================================
    Update ID: HCE2-SA-2022-0029
    Type: security
    Updated: 2022-09-08 22:08:34
    CVEs: CVE-2021-28861
    Description: Security Fix(es):
    : Python 3.x through 3.10 has an open redirection vulnerability in lib/http/server.py due to no protection against multiple (/) at the beginning of URI path which may leads to information disclosure. (CVE-2021-28861)
    Severity: Important
    ```

说明

更多详细信息，请使用**yum updateinfo --help**获取帮助信息。

6.5 检查安全更新

- 执行**yum check-update --security**命令，检查系统当前可用的安全更新。

```
[root@localhost ~]# yum check-update --security
Last metadata expiration check: 0:11:39 ago on Thu 08 Sep 2022 05:30:23 PM CST.
curl.x86_64             7.79.1-2.h6.hce2               hce2
gnupg2.x86_64           2.2.32-1.h6.hce2               hce2
kernel.x86_64           5.10.0-60.18.0.50.h425_2.hce2              hce2
unbound-libs.x86_64     1.13.2-3.h2.hce2               hce2
```

- 执行**yum check-update --sec-severity={Critical,Important,Moderate,Low}**命令，检查指定级别的安全更新。

  {}中的安全更新等级参数可任意组合。

```
[root@localhost ~]# yum check-update --sec-severity=Moderate
Last metadata expiration check: 0:23:57 ago on Thu 08 Sep 2022 05:30:23 PM CST.
gnupg2.x86_64               2.2.32-1.h6.hce2               hce2
python3-unbound.x86_64          1.13.2-3.h2.hce2               hce2
unbound-libs.x86_64         1.13.2-3.h2.hce2               hce2
```

6.6 安装安全更新

- 执行**yum upgrade --security**命令，安装全部安全更新。

```
[root@localhost ~]# yum upgrade --security
Last metadata expiration check: 5:21:24 ago on Tue 13 Sep 2022 09:43:13 AM CST.
Dependencies resolved.
================================================================
Package       Arch       Version                Repository    Size
================================================================
Installing:
Kernel        x86_64       5.10.0-60.18.0.50.h498_2.hce2    hce2       49 M
Upgrading:
Curl          x86_64       7.79.1-2.h6.hce2           hce2       147 k
......
Transaction Summary
================================================================
Install   1 Package
Upgrade  22 Packages
Total download size: 69 M
Is this ok [y/N]:
```

- 执行**yum upgrade --sec-severity={Critical,Important,Moderate,Low}**命令，安装指定级别的安全更新。

  {}中的安全更新等级参数可任意组合。

```
[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      hce2          2.2 M
python3-unbound         x86_64              1.13.2-3.h2.hce2         hce2            96 k
unbound-libs            x86_64              1.13.2-3.h2.hce2      hce2          505 k
Transaction Summary
================================================================Upgrade       3
Packages
Total download size: 2.8 M
Is this ok [y/N]:
```

- 执行**yum upgrade --advisory =<SA ID>**命令，安装指定SA ID的安全更新。

  多个软件包之间使用英文逗号分隔。

```
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        hce2        87 k
rpm           x86_64        4.17.0-8.h13.hce2        hce2        498 k
rpm-libs        x86_64        4.17.0-8.h13.hce2        hce2        376 k
Transaction Summary
================================================================
Upgrade  3 Packages
Total download size: 962 k
Is this ok [y/N]:
```

- 执行**yum upgrade --cve=<CVE ID>**命令，安装指定CVE ID的安全更新。

  多个软件包之间使用英文逗号分隔。

```
[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           hce2           8.0 M
python3-unversioned-command    x86_64          3.9.9-7.h10.hce2           hce2           3.9 k
Transaction Summary
================================================================
Upgrade  2 Packages
Total download size: 8.0 M
Is this ok [y/N]:
```

7 HCE OS 获取 openEuler 扩展软件包

HCE OS默认不加载开源社区openEuler的repo源，避免openEuler的软件包和HCE OS的软件包冲突。

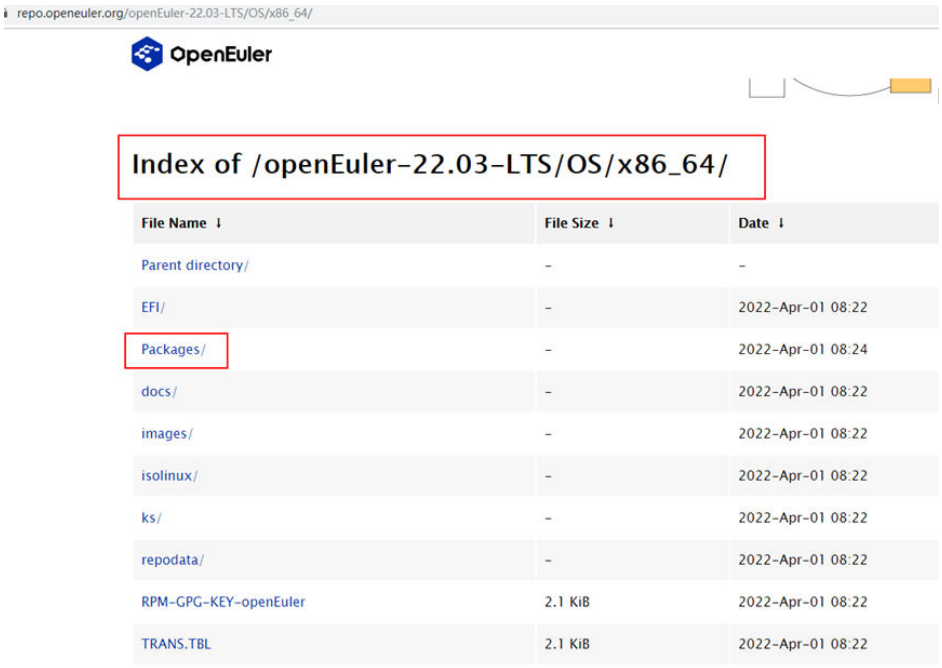
当前HCE OS 2.0版本仅兼容openEuler 22.03 LTS版本。本节介绍HCE OS 2.0如何获取openEuler 22.03 LTS的扩展软件包。获取方法有两种，请根据需要选择合适的方法。

获取方法	适用场景	安装依赖RPM包	repo文件的备份及恢复
通过wget命令下载RPM包	适用于少量RPM包的安装。	需要手动下载并安装依赖的RPM包。	不涉及
通过repo文件批量下载RPM包	适用于较多RPM包的安装。	自动安装依赖的RPM包，无需再次下载。	须先将/etc/yum.repos.d目录下原有的repo文件进行备份，并删除此目录下原有的repo文件。安装RPM包后，须再次恢复这些repo文件。

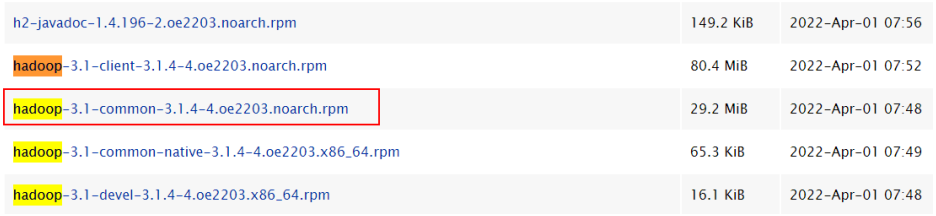
通过 wget 命令下载 RPM 包

本节以下载hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm为例，介绍使用**wget**命令下载并安装RPM包。

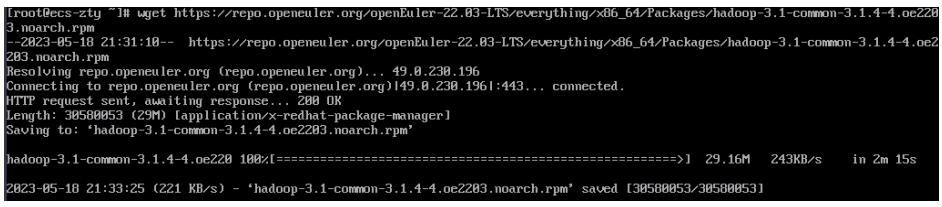
1. 单击**这里**登录openEuler社区。
2. 在OS/everything目录下，选择aarch64/或者x86_64/系统架构目录，并打开“Packages/”目录。



3. 查找所需要的RPM包，例如hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm。



4. 选择此包后右击复制下载链接，执行**wget**命令下载RPM包。



5. 检查是否下载成功。如下所示表示下载成功。

6. 使用**rpm -ivh hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm**命令安装RPM包，如下所示表示安装成功。

   如果安装过程中提示需要依赖其他的安装包，请根据同样的操作步骤先安装所依赖的安装包。


```
[root@ecs-zty ~]# rpm -ivh hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm
warning: hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm: Header V3 RSA/SHA1 Signature, key ID b25e7f66: NOKEY
error: Failed dependencies:
        apache-zookeeper is needed by hadoop-3.1-common-3.1.4-4.oe2203.noarch
        leveldb is needed by hadoop-3.1-common-3.1.4-4.oe2203.noarch
        protobuf2-java is needed by hadoop-3.1-common-3.1.4-4.oe2203.noarch
[root@ecs-zty ~]# rpm -ivh hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm --force
warning: hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm: Header V3 RSA/SHA1 Signature, key ID b25e7f66: NOKEY
error: Failed dependencies:
        apache-zookeeper is needed by hadoop-3.1-common-3.1.4-4.oe2203.noarch
        leveldb is needed by hadoop-3.1-common-3.1.4-4.oe2203.noarch
        protobuf2-java is needed by hadoop-3.1-common-3.1.4-4.oe2203.noarch
[root@ecs-zty ~]# rpm -ivh hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm --nodeps
warning: hadoop-3.1-common-3.1.4-4.oe2203.noarch.rpm: Header V3 RSA/SHA1 Signature, key ID b25e7f66: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:hadoop-3.1-common-3.1.4-4.oe2203 ################################# [100%]
[root@ecs-zty ~]#
```

通过 repo 文件批量下载 RPM 包

本节以openEuler-22.03-LTS/everything/x86_64为例，介绍下载openEuler-22.03-LTS/everything/x86_64目录下的RPM包并使用yum命令安装。

1. 首先确保虚拟机能访问**https://repo.openeuler.org/openEuler-22.03-LTS/**网址。
2. 配置yum源。

   进入/etc/yum.repos.d目录，新建一个openEuler.repo文件，并将以下内容复制到该文件里面。

   说明

   由于openEuler.repo文件和HCE OS系统repo文件有冲突，请先将/etc/yum.repos.d目录下HCE OS原有的repo文件进行备份，并删除HCE OS原有的repo文件，再创建openEuler.repo文件。

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

3. 执行**yum clean all**清除原来yum源的缓存信息。
4. 执行**yum makecache**连接新配置的源 ，如下图所示表示repo源连接成功。

```
[root@ecs-zty ~]# yum clean all
12 files removed
[root@ecs-zty ~]# yum makecache
openeuler                                                  235 kB/s | 3.4 MB     00:14
everything                                                 229 kB/s |  16 MB     01:12
HCE 2.0 base                                                59 MB/s | 6.1 MB     00:00
HCE 2.0 updates                                            1.8 MB/s | 7.8 MB     00:04
Last metadata expiration check: 0:00:02 ago on Thu 18 May 2023 09:55:03 PM CST.
Metadata cache created.
```

5. 安装RPM包，以hadoop-3.1-common包为例。
   a. 执行**yum list**命令查看是否存在该包。

```
[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. 执行**yum -y install hadoop-3.1-common**命令来安装此包，如下所示表示该包已经安装成功。

```
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. 恢复repo文件。

   安装所需的openEuler包后，删除openEuler.repo文件，并将步骤**2**中删除的repo文件通过备份恢复。

8 制作 Docker 镜像并启动容器

本节介绍在HCE OS上制作HCE OS系统的Docker镜像并启动容器。

约束限制

- 运行容器镜像的HCE OS系统版本和制作的HCE OS容器镜像版本须保持一致。

制作镜像归档文件

1. 确认HCE OS的repo源配置正常。

   请检查默认的/etc/yum.repos.d/hce.repo配置文件中的repo源地址是否正确，正确的配置如下。

   ```
   [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/
   ......
   ```

2. 新建临时目录作为Docker镜像的根系统文件，并将软件包安装到临时目录。

   ```
   rm -rf /tmp/docker_rootfs
   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
   ```

   ⚠ 注意

   - 上述操作中的**releasever**需替换为HCE OS对应的版本号。
   - 也可在此处安装其他所需的软件包，但是要确保/tmp文件下的空间足够。

3. chroot进入临时目录，进行如下配置。

   ```
   chroot /tmp/docker_rootfs
   ```

   ```
   [root@localhost tmp]# chroot /tmp/docker_rootfs
   [root@localhost /]#
   [root@localhost /]#
   ```

a. 使用HCE security-tool.sh关闭不必要服务。

```
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
```

执行过程中，有以下错误打印均为正常现象，报错的原因为：

- 缺少服务文件。在chroot文件系统下没有启动服务导致。
- 缺少引导系统的文件/etc/sysconfig/init。工具在系统启动阶段关闭服务，镜像rootfs不涉及系统启动。
- 缺少/proc/sys/kernel/sysrq，这是系统启动后生成的调用节点，在chroot文件系统下不存在。

```
[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. 卸载security-tool、cronie、systemd软件包及其依赖软件包。

```
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'
[ -d /var/lib/dnf ] && rm -rf /var/lib/dnf/*
[ -d /var/lib/rpm ] && rm -rf /var/lib/rpm/__db.*
```

c. 移除/boot目录。

```
rm -rf /boot
```

d. 设置容器镜像语言为en_US。

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

e. 移除共享文件 man、doc、info和mime。

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

f. 移除缓存日志文件。

```
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. 移除java安全证书。
```
rm -rf /etc/pki/ca-trust/extracted/java/cacerts /etc/pki/java/cacerts
```

h. 移除/etc/machine-id。
```
rm -rf /etc/machine-id
```

i. 移除/etc/mtab。
```
rm -rf /etc/mtab
```

4. 退出chroot。
```
exit
```

5. 打包压缩临时目录，生成Docker镜像归档文件hce-docker.x86_64.tar.xz。

   归档路径为/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
```

6. 将镜像归档文件转换为带有layer信息的文件。

   上述生成的镜像归档文件没有layer信息，不能使用**docker load**命令加载镜像。需要用**docker import**命令先将归档文件生成为镜像，再用**docker save**命令保存为一个带有layer信息的镜像文件，这样就可以使用**docker load**命令来加载镜像了。下面以镜像名为my_image，归档文件为docker_save.tar.xz举例。
```
docker import hce-docker.x86_64.tar.xz my_image:v1
docker save -o docker_save.tar.xz my_image:v1
```

   此时生成的docker_save.tar.xz归档文件就可以用**docker load**命令来加载镜像。
```
docker load -i docker_save.tar.xz
```

使用镜像归档文件启动容器

1. 确认HCE OS的repo源配置正常。

   请检查默认的/etc/yum.repos.d/hce.repo配置文件中的repo源地址是否正确，正确的配置如下。
```
[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/
.....
```

2. 安装docker软件包。
```
yum install docker -y
```

3. 使用镜像归档文件创建容器镜像。
```
mv /tmp/docker_rootfs/hce-docker.x86_64.tar.xz .
docker import hce-docker.x86_64.tar.xz
```

   执行**docker images**命令可查看到容器镜像ID为6cfefae3a541。
```
[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
```

说明

创建镜像可使用如下命令指定镜像的REPOSITORY和TAG参数。

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

4. 在容器中运行镜像bash文件。

   运行如下命令后，如果shell视图改变，表示成功进入容器bash。

   docker run -it 6cfefae3a541 bash

```
[root@localhost /]# docker run -it 6cfefae3a541 bash
[root@5c639b63fc0e /]#
[root@5c639b63fc0e /]#
```

9 工具类

9.1 毕昇编译器

毕昇编译器（bisheng compiler）是华为提供的一款提供高性能、高可信及易扩展的编译器工具链。毕昇编译器引入了多种编译技术，支持C/C++/Fortran编译语言。

约束限制

- 仅HCE OS 2.0 x86架构支持使用毕昇编译器。
- HCE OS原生的clang编译语言和毕昇编译器提供的clang编译语言不能同时使用。如果您已经安装原生的clang编译语言并需要使用它，就不能安装毕昇编译器。

  在安装了毕昇编译器之后，如果需要使用原生的clang编译语言，可执行**rpm -e bisheng-compiler**命令删除毕昇编译器，然后打开新终端。在新终端中，就可以使用原生的clang编译语言。

安装毕昇编译器

1. 确认repo源配置正常。

   请检查默认的/etc/yum.repos.d/hce.repo配置文件中参数是否正确，正确的配置如下。

   ```
   [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/
   ......
   ```

2. 执行**yum install bisheng-compiler**命令安装工具。
3. 执行**source /usr/local/bisheng-compiler/env.sh**命令，导入环境变量。

   如果打开了新的终端，需要在新的终端重新导入环境变量才能正常使用毕昇编译器。
4. 检查工具是否安装成功。

执行**clang -v**查看工具的版本号。若返回结果包含毕昇编译器版本信息，表示工具安装成功。

使用毕昇编译器

1. 编译运行C/C++程序。
```
clang [command line flags] hello.c -o hello.o
./hello.o
clang++ [command line flags] hello.cpp -o hello.o
./hello.o
```
2. 编译运行Fortran程序。
```
flang [command line flags] hello.f90 -o hello.o
./hello.o
```
3. 指定链接器。

   毕昇编译器指定的链接器是LLVM的lld，若不指定它则使用默认的ld。
```
clang [command line flags] -fuse-ld=lld hello.c -o hello.o
./hello.o
```

9.2 应用加速工具

9.2.1 概述

应用加速工具是华为云提供的一款对应用进行性能优化的工具。

应用加速工具优化应用程序有两种方式。

- 静态加速：

  静态加速只需要在应用程序运行时采集所在CPU上的pmu监控信息，基于采集到的监控信息将应用程序做静态重新制作，生成新的高性能应用程序二进制。该过程不需要应用程序代码做修改或者仅需要对编译器参数做调整。静态加速有两种优化方式。
  - **使用原生的BOLT工具优化应用程序**：只能使用固定参数组合优化应用。
  - **使用hce-wae-auto命令优化应用程序**：可以根据自定义参数范围，生成不同的参数组合分别来优化应用。
- 动态加速：

  动态加速工具直接对目标应用进程进行加速，无需中断业务，在业务无感知的情况下完成优化工作。

表 9-1 静态加速和动态加速优缺点

应用加速方式	优点	缺点
静态加速	以二进制可执行文件为粒度进行优化，无需修改程序代码。	优化后需要重启应用程序。
动态加速	以应用进程为粒度进行优化，无需重启应用程序，并能够生成应用快照保存优化结果。同时保证二进制文件溯源能力，能够不断迭代优化应用进程，直至达到性能优化瓶颈。	当前仅支持插桩方式采集数据且仅能够进行一次优化。

约束限制

- 仅HCE OS 2.0 x86架构支持使用应用加速工具。
- 仅root用户支持使用应用加速工具。

操作流程

1. 步骤一：**安装应用加速工具**。
2. 步骤二：采用**静态加速**或者**动态加速**方式优化应用。

9.2.2 安装工具

1. 确认repo源配置正常。

   请检查默认的/etc/yum.repos.d/hce.repo配置文件中参数是否正确，正确的配置如下。

   ```
   [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/
   ......
   ```

2. 执行**yum install hce-wae**命令安装加速工具。
3. 检查工具是否安装成功。

   执行llvm-bolt帮助命令有如下输出信息，表示工具安装成功。

   ```
   [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
   ```

9.2.3 静态加速

准备工作

1. 执行如下命令检查待优化的二进制文件中是否可以重新定位。可以重新定位表示可以进行应用优化。

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

   - 如果二进制文件中.rela.text段存在，表示可以重新定位。

     ```
     [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]#
     ```

- 如果不存在，为了允许BOLT在程序中重新排列函数（除了重新排列函数中的代码），需要将--emit-relocs或-q添加到应用程序的最后链接步骤中。

2. 采集应用运行时的日志数据。

   部署并预热应用后，即可使用**llvm-bolt -instrument -o -instrumentation-file**命令配置应用的日志采集方式。

   例如，配置test.so文件运行后每隔30秒收集一次日志，日志保存到运行时test.log文件中请使用如下命令。

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

   - instrument -o：配置完日志采集方式后生成的新的动态库文件。本例中新生成的动态库为testd.so。
   - instrumentation-file：日志保存的文件名称。本例为test.log。
   - instrumentation-sleep-time：采集日志的时间间隔，单位为秒。本例中每隔30秒采集一次日志。
   - instrumentation-no-counters-clear：表示每次日志采集后不要清除日志计数器信息，保持日志信息上下文连续

3. 运行testd.so对应的应用程序，应用程序运行日志会自动保存在test.log文件中。

   应用加速工具会根据test.log文件中的动态数据来优化应用。

优化应用程序

- 使用原生的BOLT工具优化应用程序。

  准备好test.log文件后，就可以使用它与BOLT对应用程序进行优化。举例如下。

  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

  说明

  以上为优化参数示例，并非最优参数组合。

- 使用**hce-wae-auto**命令优化应用程序。

  hce-wae-auto命令根据自定义的**配置文件**来优化应用程序。

  命令格式为：hce-wae-auto [-h] [-c <Path>] [-s <Keyword>] [-e <Pattern>] [-l] [--free] application

  说明

  - 执行**hce-wae-auto**命令后，产生的参数集信息路径默认与配置文件中二进制输出路径一致，路径为：/data/hce-wae-auto/hce-wae-auto.data。
  - 执行**hce-wae-auto**命令后，系统自动产生日志文件，默认路径为：/var/log/hce-wae-auto/hce-wae-auto.log。

  表 9-2 参数说明

参数名	是否需要赋值	取值类型	可选/必选	参数含义
-h	否	/	可选	显示命令参数帮助信息。

参数名	是否需要赋值	取值类型	可选/必选	参数含义
-c	是	string	可选	定义配置文件路径，默认路径为/etc/hce-wae-auto.conf。
-s	是	string	可选	定义筛选的关键词，用于模糊搜索上次参数集内容。 如果没有保存的参数集，则不生效。
-e	是	string	可选	根据上次参数集内容，执行选定的参数组合。需指定序号。 ● 单个数字指定对应序号的参数组合。 ● “:”表示序号范围。例如，4:6表示序号4到序号6；:7表示序号1到序号7；5:表示序号5到最后一个；:表示全部。 ● 单个数字可与序号范围“:”一起使用，用“,”分割。例如，1,4:6表示序号1、序号4、序号5和序号6。
-l	否	/	可选	根据配置文件生成参数集并打印，不自动执行命令，不保存参数集信息。
--free	否	/	可选	不校验配置文件中参数合法性。 应用加速工具会对配置文件中的参数进行简单校验，判断参数是否属于bolt的优化参数。若指定--free指令，则取消该校验，允许输入非优化参数。
application	否	/	必选	要执行的二进制文件，可以添加相对路径或绝对路径。例如，mysqld，/usr/bin/mysqld，../bin/mysqld。

说明

-c、-s、-e、-l参数存在冲突时，优先级为-s > -l > -e > -c。

hce-wae-auto命令使用示例如下。

示例	说明
hce-wae-auto mysqld	从默认路径读取配置，根据配置参数生成参数集，并自动执行命令。命令执行完成后自动保存参数集信息。
hce-wae-auto mysqld -c /etc/my.conf -l	从指定路径/etc/my.conf读取配置参数，生成参数集并打印生成命令。不执行命令并且不保存参数集信息。

示例	说明
hce-wae-auto mysqld -c /etc/my.conf	从指定路径/etc/my.conf读取配置参数，生成参数集并自动执行命令。执行完成后自动保存参数集信息。
hce-wae-auto mysqld -s align	从上次生成的参数集信息中模糊搜索关键词align，打印匹配的参数组合。
hce-wae-auto mysqld -e 4:6	从上次生成的参数集信息中，选择序号4、5、6的参数组合，生成二进制文件。
hce-wae-auto mysqld -e 1,4:6	从上次生成的参数集信息中，选择序号1、4、5、6的参数组合，生成二进制文件。
hce-wae-auto mysqld -e :	从上次生成的参数集信息中，选择所有参数组合，生成二进制文件。
hce-wae-auto mysqld -e 4:6 -l	从上次生成的参数集信息中，选择序号4、5、6的参数组合，打印生成命令，并且不执行命令。
hce-wae-auto mysqld -e : -l	从上次生成的参数集信息中，选择所有参数组合，打印生成的命令，并且不执行命令。
hce-wae-auto mysqld --free	从默认路径读取配置，不对参数进行校验，根据配置参数生成参数集，并自动执行命令。命令执行完成后自动保存参数集信息。
hce-wae-auto -h	显示指令帮助信息。

9.2.4 动态加速

准备工作

在做动态加速之前，请先做如下两个检查。两个条件都满足，才能对应用进行动态加速。

1. 执行如下命令检查待优化的二进制文件中是否可以重新定位。可以重新定位表示可以进行应用优化。

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

   - 如果二进制文件中.rela.text段存在，表示可以重新定位。

     ```
     [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]#
     ```

   - 如果不存在，为了允许BOLT在程序中重新排列函数（除了重新排列函数中的代码），需要将--emit-relocs或-q添加到应用程序的最后链接步骤中。

2. 执行**hce-wae --check**命令查看应用是否支持动态加速。

   如果检查结果为3，表示可以使用动态加速工具。否则不支持动态加速。

   ```
   [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
   ```

操作步骤

本例以优化/data/apps/mysql-8.0.28/bin目录下的mysqld应用 ，为您介绍动态加速方式优化应用的操作。

1. 生成插桩版应用并运行。

   a. 执行命令**/data/hce-wae/dbo/gen_instrumentation /data/apps/mysql-8.0.28/bin/mysqld**生成插桩版应用。

   命令格式：**/data/hce-wae/dbo/gen_instrumentation** *应用路径*

```
[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 ~]#
```

   命令运行完成后，会在当前目录生成对应的以.inst为后缀的插桩文件mysqld.inst。

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

   b. 运行插桩文件获取进程PID，本例为87042。

```
[root@localhost ~]# /data/apps/mysql-8.0.28/bin/mysqld.inst -uroot &
[1] 87042
[root@localhost ~]# 2023-09-15T02:30:08.152217Z 0 [System] [MY-010116] [Server] /data/apps/mysql-8.0.28/bin/mysqld.inst (mysqld 8.0.28) start
ing as process 87042
2023-09-15T02:30:08.565656Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2023-09-15T02:30:11.547443Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2023-09-15T02:30:13.032029Z 0 [System] [MY-010229] [Server] Starting XA crash recovery...
2023-09-15T02:30:13.040024Z 0 [System] [MY-010232] [Server] XA crash recovery finished.
2023-09-15T02:30:13.377357Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2023-09-15T02:30:13.377427Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now suppo
rted for this channel.
2023-09-15T02:30:13.488591Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /tmp/mysq
lx.sock
2023-09-15T02:30:13.488631Z 0 [System] [MY-010931] [Server] /data/apps/mysql-8.0.28/bin/mysqld.inst: ready for connections. Version: '8.0.28'
  socket: '/tmp/mysql.sock'  port: 3306  Source distribution.
```

2. 创建mysqld的应用加速动态配置文件。

   每一个待优化的应用都要有一个对应的配置文件，应用加速工具根据此配置文件对应用进行动态加速。

   a. 执行如下命令复制一份默认的配置文件为**/data/hce-wae/config/mysqld.conf**。

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

   b. 设置**/data/hce-wae/config/mysqld.conf**配置文件中的origin-exe字段。

   origin-exe为待优化应用的位置，本例为**/data/apps/mysql-8.0.28/bin/mysqld**

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

```
# 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=/data/apps/mysql-8.0.28/bin/mysqld

# 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=1
```

3. 使用配置文件和对应的进程PID配置动态加速工具。

   命令格式：hce-wae --conf [PID] [/path/to/config]

```
[root@localhost ~]# hce-wae --conf 87042 /data/hce-wae/config/mysqld.conf
2023-09-15 10:33:29,478-INFO: Log level: INFO
2023-09-15 10:33:29,479-INFO: mission will stop after run 1 times
2023-09-15 10:33:29,479-INFO: record mission from config succeed
[root@localhost ~]#
```

4. 启动动态加速，对插桩版mysql进行优化。

   命令格式：hce-wae --start [PID]

```
[root@localhost ~]# hce-wae --start 87042
2023-09-15 10:43:18,421-INFO: Log level: INFO
2023-09-15 10:43:18,422-INFO: start dbo mission for /data/apps/mysql-8.0.28/runtime_output_directory/mysqld.inst ...
2023-09-15 10:43:18,422-INFO: extracting call sites, it may take serval minutes ...
2023-09-15 10:44:00,414-INFO: preparing ...
start running dbo server to monitor process 87042
2023-09-15 10:44:00,419-INFO: run dbo for /data/apps/mysql-8.0.28/runtime_output_directory/mysqld.inst(pid: 87042) succeed
2023-09-15 10:44:00,419-INFO: starting ...
start optimizing
2023-09-15 10:44:00,422-INFO: run dbo for /data/apps/mysql-8.0.28/runtime_output_directory/mysqld.inst(pid: 87042) succeed
2023-09-15 10:44:00,422-INFO: mission started, target_pid: 87042
2023-09-15 10:44:00,422-INFO: recording started mission info...
2023-09-15 10:44:00,423-INFO: start mission worker...
2023-09-15 10:44:00,424-INFO: mission worker for 87042 started
```

   启动后，可以通过--status参数查看当前优化状态。当状态为Running时，表示进程正在优化中；Finished时，表示进程已经优化完成。

   命令格式：hce-wae --status [PID]

```
[root@localhost ~]# hce-wae --status 87042
2023-09-15 10:44:48,379-INFO: Log level: INFO
2023-09-15 10:44:48,384-INFO: status: {
    "status": "Running",
    "sub_status": "DataCollecting",
    "run_times": 0,
    "failed_code": "NoFailed"
}
[root@localhost ~]# hce-wae --status 87042
2023-09-15 10:45:22,066-INFO: Log level: INFO
2023-09-15 10:45:22,072-INFO: status: {
    "status": "Finished",
    "sub_status": "Done",
    "run_times": 1,
    "failed_code": "NoFailed"
}
```

5. 优化后，通过--snapshot参数生成优化后的.dbo二进制快照文件，本例为mysqld.dbo。

```
[root@localhost ~]# hce-wae --snapshot 87042
2023-09-15 10:46:00,433-INFO: Log level: INFO
2023-09-15 10:46:00,438-INFO: dbo snapshot for 87042 succeed
2023-09-15 10:46:00,438-INFO: snapshot generated, target_pid: 87042, path: /data/hce-wae/snapshot
[root@localhost ~]# ll /data/hce-wae/snapshot/
total 71384
-rwxrwxrwx. 1 root root 148936512 Sep 15 10:46 mysqld.dbo
```

快照生成的默认路径为/data/hce-wae/snapshot/，可在配置文件中对快照位置进行修改。后续您可以直接使用此优化后的快照文件mysqld.dbo运行应用，无需重复优化。

6. 终止动态加速工具，应用优化结束。

命令格式：hce-wae --stop [PID]

```
[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
```

动态应用加速工具字符交互界面

动态应用加速工具支持字符交互界面，交互界面支持指令如图9-1、图9-2和表9-3所示。

图 9-1 动态应用加速工具启动界面

```
[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>
```

图 9-2 动态应用加速工具帮助界面

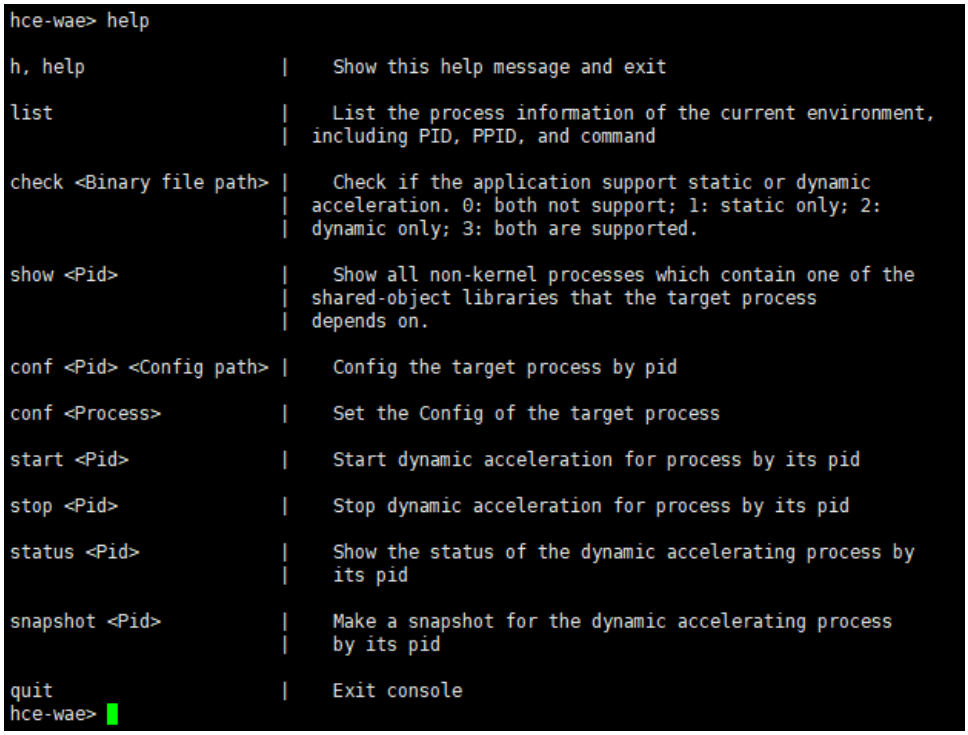


表 9-3 动态应用加速工具字符交互界面指令

字符交互界面指令	使用方式	指令描述
list	list	获取当前环境的进程信息，包含PID、PPID以及command信息。
status	status <PID>	查询当前正在进行的动态加速任务及状态。输入后在字符界面展示信息包括： • 应用进程PID • 应用进程名 • 当前已优化次数 • 当前加速状态(Initialized/DataCollecting/Optimizing/HotPatching/Done/Failed)
check	check <PID>	检测当前进程是否支持静态/动态加速，展示文字结果。
show	show <PID>	查询进程的依赖关系
conf	conf <PID>	为目标进程配置动态加速能力，输入指令后依次弹出配置选项，根据提示进行配置即可。
start	start <PID>	启动动态加速
stop	stop <PID>	终止动态加速

字符交互界面指令	使用方式	指令描述
snapshot	snapshot <PID>	生成动态加速快照
quit	quit	退出字符交互界面
h/help	h/help	显示帮助信息

9.2.5 配置文件

本节提供了**静态加速配置文件**和**动态加速配置文件**中每个配置项的说明。

静态加速配置文件

应用加速工具默认静态加速配置信息如下，您可以自定义配置文件来优化应用。

```
# 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
```

表 9-4 配置信息说明

模块名	描述
binary	定义二进制文件的相关属性。binary参数详情参见**表9-5**。
parameter	用户定义的参数集合，工具根据此集合生成参数集。至少定义一个参数。 通过**llvm-bolt -h**命令可查看所有参数。

模块名	描述
include	用户定义参数集中需包含的参数，允许定义多个参数，多个参数之间采用“与”逻辑。 选项以参数名为key，指定值为value。例如，frame-opt=none。 **说明** • 若key无法指定值或不需要指定值时，仍需要以“=”结束，例如frame-opt=。 • include和exclude配置包含同一个参数时，exclude的优先级大于include。
exclude	用户定义参数集中不需要包含的参数。允许定义多个参数，多个参数之间采用“与”逻辑。 **说明** • 若key无法指定值或不需要指定值时，仍需要以“=”结束，例如frame-opt=。 • include和exclude配置包含同一个参数时，exclude的优先级大于include。

表 9-5 binary 参数说明

选项名	值类型	默认值	选项描述
binary_out_path	string	"/data/hce-wae-auto"	定义自动生成的二进制文件的保存路径。
binary_file_suffix	string	"blot.auto"	定义自动生成的二进制文件名后缀。

静态加速配置文件示例

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

[parameter]
parameters =
    --align-blocks              # 允许添加参数前缀--
    frame-opt                   # 用户定义的参数集合中若不需要指定参数的值，则无需以=结束
    align-functions=1           # 用户定义参数集合中指定了参数对应的值，则生成的参数集中，所有参数组合中该参数的值都为1

[include]
align-blocks=                   # 参数无法指定值仍需以=结束
[exclude]
frame-opt=none                  # 指定参数及对应的值，生成的参数集中会过滤参数为frame-opt，且值为none的参数组合
indirect-call-promotion=        # 指定参数,该参数为枚举类型，则生成的参数集中会过滤所有参数为frame-opt的参数组合
```

动态加速配置文件

应用加速工具默认动态加速配置信息如下，您可以自定义配置文件来优化应用。


```
# 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
```

表 9-6 配置信息说明

模块名	描述
[mission]优化运行中的应用所要配置的参数。	
log-type	运行时日志采集方式，当前仅支持instrument方式。
hotpatch-type	热补丁模式，当前仅支持mode1即ptrace方式。
snapshot-path	优化后的二进制快照文件存放的目录路径。
origin-exe	原始应用的位置，使用instrument日志采集模式时，须指定此参数。
[stop-strategy]应用优化停止策略，请选择其中一种配置。	
run-times	指定优化次数，达到该次数时动态加速工具会停止优化，当前仅支持1次。
period	指定优化周期，达到该时间周期时停止优化，单位为秒，取值范围为1~600。
condition	指定优化条件，达到该条件时停止优化，当前不支持。

9.3 Pod 带宽管理工具

在业务混合部署的场景下，Pod带宽管理功能根据QoS分级对资源进行合理调度，提升网络带宽利用率。HCE OS提供oncn-tbwm带宽管理工具，使用tbwmcli命令对收发包方向的网络限速功能，实现网络QoS。

前提条件

本功能固定使用ifb0，使用前请确定虚拟网卡ifb0未被使用，并加载ifb驱动。

约束与限制

- 仅HCE OS 2.0 x86架构支持使用tbwmcli命令。

- 仅允许root用户执行tbwmcli命令。
- tbwmcli命令同一时间只能在一个网卡使能Qos功能，多个网卡不支持并行使能网络QoS。
- 网卡被插拔重新恢复后，原来设置的QoS规则会丢失，需要手动重新配置网络QoS功能。
- 不支持cgroup v2。
- 升级oncn-tbwm软件包不会影响升级前的使能状态。卸载oncn-tbwm软件包会关闭对所有设备的使能。
- 仅支持识别数字、英文字母、中划线“-” 和下划线“_”四类字符类型的网卡名，其他字符类型的网卡不被识别。
- 实际使用过程中，带宽限速有可能造成协议栈内存积压，此时依赖传输层协议自行反压，对于udp等无反压机制的协议场景，可能出现丢包、ENOBUFS、限流不准等问题。
- 收包方向的网络限速依赖于TCP的反压能力，在非TCP协议的场景中，网络包已经收至目标网卡，不支持对于收包方向的网络限速。
- 不支持tbwmcli、tc命令和网卡命令混用，只能单独使用tbwmcli工具进行限速。例如，某个网卡上已经设置过tc qdisc规则的情况下，对此网卡使能网络QoS功能可能会失败。

使用方法

1. 安装oncn-tbwm软件包。
   a. 确认repo源配置正常。

      请检查默认的/etc/yum.repos.d/hce.repo配置文件中参数是否正确，正确的配置如下。

      ```
      [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/
      ......
      ```

   b. 执行**yum install oncn-tbwm**命令安装oncn-tbwm软件包。
   c. 验证oncn-tbwm软件包正确性。
      - 执行**tbwmcli -v**命令，正确安装则结果显示如下。
        ```
        version: 1.0
        ```
      - 确认以下oncn-tbwm服务组件，正常情况下以下服务组件均存在。
        ```
        /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. 根据需要执行tbwmcli命令。

表 9-7 tbwmcli 命令说明

命令	说明
tbwmcli -e ethx tbwmcli -d ethx egress	使能/关闭某个网卡的发包方向Qos功能。 示例：使能eth0网卡的发包方向Qos功能。 **tbwmcli -e eth0** enable eth0 egress success 示例：关闭eth0网卡的发包方向Qos功能。 **tbwmcli -d eth0 egress** disable eth0 egress success
tbwmcli -i ethx online/offline tbwmcli -d ethx ingress	使能/关闭某个网卡的收包方向Qos功能。 示例：使能eth0网卡收包方向Qos功能，并设置为在线网卡。 **tbwmcli -i eth0 online** enable eth0 ingress success, dev is online 示例：使能eth0网卡收包方向Qos功能，并设置为离线网卡。 **tbwmcli -i eth0 offline** enable eth0 ingress success, dev is offline **说明** 收包方向不支持同时设置为多个离线网卡的情况，支持一个离线网卡和多个在线网卡。 示例：关闭eth0网卡收包方向Qos功能。 **tbwmcli -d eth0 ingress** disable eth0 ingress success
tbwmcli -d ethx	强制关闭某个网卡的所有Qos功能，并关闭ifb功能。 示例：强制关闭eth0网卡的所有Qos功能，并关闭ifb功能。 **tbwmcli -d eth0** disable eth0 success
tbwmcli -p istats/estats	打印收/发包方向内部统计信息。 示例：打印收包方向内部统计信息。 **tbwmcli -p istats** offline_target_bandwidth: 94371840online_pkts: 3626190offline_pkts: 265807online_rate: 0offline_rate: 13580offline_prio: 0 示例：打印发包方向内部统计信息。 **tbwmcli -p estats** offline_target_bandwidth: 94371840online_pkts: 4805452offline_pkts: 373961online_rate: 0offline_rate: 19307offline_prio: 1

命令	说明
tbwmcli -s path <prio> tbwmcli -p path	设置/查询cgroup的QoS优先级。 当前仅支持设置离线和在线两个QoS优先级。 ● 0：设置cgroup为在线QoS优先级。 ● -1：设置cgroup为离线QoS优先级。 示例：设置test_online cgroup的优先级为0。 **tbwmcli -s /sys/fs/cgroup/test_online 0** set prio success 示例：查询test_online cgroup的优先级。 **tbwmcli -p /sys/fs/cgroup/test_online** prio is 0
tbwmcli -s bandwidth <low，high> tbwmcli -p bandwidth	设置/查询离线带宽范围。 示例：设置离线宽带范围为30mb~100mb。 **tbwmcli -s bandwidth 30mb,100mb** set bandwidth success 示例：查询离线带宽范围。 **tbwmcli -p bandwidth** bandwidth is 31457280(B),104857600(B)
tbwmcli -s waterline <val> tbwmcli -p waterlin	设置/查询在线网络带宽水线。 示例：设置在线网络带宽水线为20mb。 **tbwmcli -s waterline 20mb** set waterline success 示例：查询在线网络带宽水线。 **tbwmcli -p waterline** waterline is 20971520 (B)
tbwmcli -p devs	查看系统上所有网卡的使能状态。 **tbwmcli -p devs** lo Egress : disabled lo Ingress : disabled eth0 Egress : disabled eth0 Ingress : enabled, it's offline ifb0 Egress : enabled
tbwmcli -c	强制删除所有网卡的qos，谨慎使用。
modprobe ifb numifbs=1	加载ifb。
rmmod ifb	卸载ifb。

10 内核功能与接口

10.1 内核 memory 的 OOM 进程控制策略

背景信息

现有操作系统中，支持配置离线业务和在线业务。当内存发生OOM时，会优先选择离线业务控制组中的消耗内存最多的进程，结束进程回收内存，但是对于某些离线业务也有核心业务，因此会造成很大的影响。

针对这个问题，HCE OS调整了OOM时回收内存的策略，增加了配置cgroup优先级的功能。 内存紧张情况下内核会遍历cgroup，对低优先级的cgroup结束进程，并回收内存，使离线业务中重要的业务可以存活下来。

前提条件

vm.panic_on_oom接口默认开启，系统OOM时panic。故使用memcg OOM优先级配置时（即memcg_qos_enable配置为1或2），须先执行**sysctl -w vm.panic_on_oom=0**命令，关闭系统参数vm.panic_on_oom。

memcg OOM 优先级接口功能说明

接口	说明	取值
memcg_qos_enable	memcg OOM优先级策略开关。 ● 0：不开启优先级配置。当OOM时，按照系统原有的OOM操作结束进程，结束内存消耗最大的进程，回收内存。 ● 1：开启优先级配置并以cgroup为粒度。当OOM时，结束优先级低的cgroup所有进程，并回收内存。 ● 2：开启优先级配置并以单个进程个为粒度。当OOM时，结束优先级低的cgroup中的最大的一个进程，并回收内存。	整数形式，取值范围为0~2，默认值为0。

接口	说明	取值
memory.qos_level	配置cgroup组优先级。值越小cgroup组优先级越低。 ● 当memcg OOM时，会以当前cgroup组为父节点，查找子节点优先级最低的cgroup组中内存使用最大的进程，结束该进程，回收内存。 ● 当OOM时，对于优先级相等的cgroup组，会根据组的内存使用量进行二次排序，选择内存使用最大的进行OOM操作。 **说明** ● 使用memory.qos_level的前提条件为memcg_qos_enable取值须为1或2。 ● 新创建的cgroup组的memory.qos_level值默认会继承父节点的memory.qos_level的值，但是子节点的优先级不受父节点的限制。 ● 如果修改cgroup组父节点的优先级，子节点的优先级会自动调整，和父节点保持一致。	整数形式，取值范围为-1024~1023，默认值为0。

接口配置示例

按如下所示创建6个cgroup子节点A、B、C、D、E、F，配置memcg_qos_enable接口，并通过memory.qos_level接口设置OOM的优先级，优先级取值如图所示。

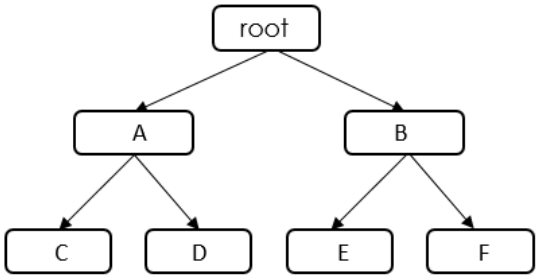


表 10-1 数据规划

cgroup组	memory.qos_level取值	说明
A	**-8**	当在root中进行OOM操作时，内核遍历root所有cgroup组，最终选择优先级最低的A、E。由于A和E优先级相同，内核继续对A和E使用的内存进行比较。 ● 如果设置memcg_qos_enable=1，优先对A/E中内存使用量大的一个cgroup组回收内存，结束cgroup组中的所有进程，并回收内存。 ● 如果设置memcg_qos_enable=2，优先对A/E中内存使用量最大的一个进程回收内存。
B	10	
C	1	
D	2	
E	**-8**	
F	3	

1. 关闭系统参数vm.panic_on_oom。
```
sysctl -w vm.panic_on_oom=0
```
2. 开启memcg OOM优先级策略功能。
```
echo 1 > /proc/sys/vm/memcg_qos_enable
```
3. 运行以下命令创建两个cgroup节点 A、B，并分别设置A、B节点的memcg OOM优先级值为-8、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. 运行以下命令分别在A节点下创建C、D子节点，在B节点下创建E、F子节点， 并分别设置C、D、E、F子节点的memcg OOM优先级值为1、2、-8、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 内核 memory 的多级内存回收策略

需求背景

在容器高密度混合部署场景中，IO读写较多的离线业务消耗大量page cache，导致系统空闲内存降低，达到全局空闲内存水位线后触发全局内存回收，使得在线任务申请内存时进入内存回收的慢路径，引发时延抖动。

为解决此问题，HCE OS 2.0新增支持多级内存回收策略。申请内存时，设置内存警示值，可触发内存回收任务，保证可用内存空间；回收内存时，设置多级内存保护水位线，保护任务可用性。

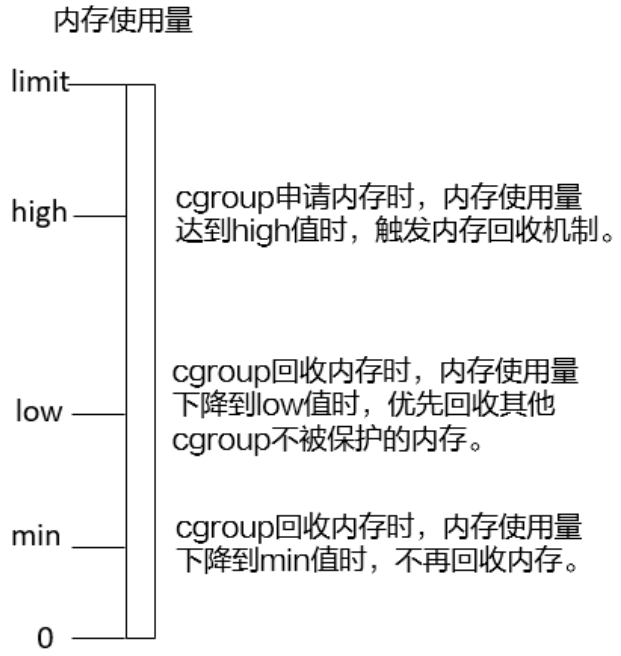
约束与限制

memory.min和memory.low只在叶子节点生效。创建memory cgroup时，会将父节点的memory.min和memory.low清零。

多级内存回收接口说明

memory.min、memory.low和memory.high接口在非根的memory cgroup下面默认存在，可以向文件内写值配置，也可以读取当前配置。合理的取值大小顺序为memory.min≤memory.low<memory.high，三者可独立使用，也可联合使用。

内存回收机制如下图。



接口	说明
memory.min	硬保护内存保护值，默认值为0。系统没有可回收内存的时候，也不会回收在该值边界及以下的内存。读写说明如下： ● 读该接口可以查看硬保护内存大小，单位为byte。 ● 写该接口可以设置硬保护内存大小，单位不做限制。 ● 配置范围：0-memory.limit_in_bytes。

接口	说明
memory.low	尽力而为的内存保护值，默认值为0。 系统优先回收未被保护的cgroup组的内存。如果内存还不足，再回收memory.min到memory.low之间的内存。 读写说明如下： ● 读该接口可以查看Best-effort内存保护值，单位为byte。 ● 写该接口可以设置Best-effort内存保护值，单位不做限制。 ● 配置范围：0-memory.limit_in_bytes。
memory.high	内存回收警示值，默认为max。当cgroup组内存使用量达到high值，会触发对该cgroup及子节点的同步内存回收任务，将内存尽量限制在high之下，避免触发memory.limit限制导致OOM。读写说明如下： ● 读该接口可以查看Throttle limit内存保护值，单位为byte。 ● 写该接口可以设置Throttle limit内存保护值，单位不做限制。 ● 配置范围：0-memory.limit_in_bytes

接口示例

按如下所示创建6个cgroup节点A、B、C、D、E、F，配置memory.min接口，示例说明多级内存回收策略。

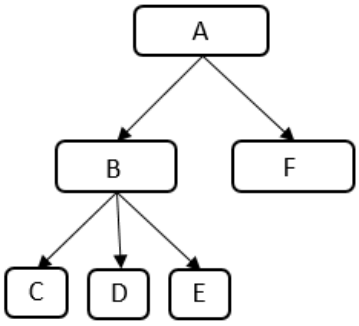


表 10-2 数据规划

cgroup组	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. 创建cgroup节点A，并配置memory.limit_in_bytes=200M。
```
mkdir /sys/fs/cgroup/memory/A
echo 200M > /sys/fs/cgroup/memory/A/memory.limit_in_bytes
```
2. 创建cgroup节点B。
```
mkdir /sys/fs/cgroup/memory/A/B
```
3. 创建cgroup节点C，并配置memory.min=75M，在当前节点创建申请50M cache的进程。
```
mkdir /sys/fs/cgroup/memory/A/B/C
echo 75M > /sys/fs/cgroup/memory/A/B/C/memory.min
```
4. 创建cgroup节点D，并配置memory.min=25M，在当前节点创建申请50M cache的进程。
```
mkdir /sys/fs/cgroup/memory/A/B/D
echo 25M > /sys/fs/cgroup/memory/A/B/D/memory.min
```
5. 创建cgroup节点E，并配置memory.min=0，在当前节点创建申请50M cache的进程。
```
mkdir /sys/fs/cgroup/memory/A/B/E
```
6. 创建cgroup节点F，并配置memory.min=125M，申请125M保护内存，触发内存回收。
```
mkdir /sys/fs/cgroup/memory/A/F
echo 125M > /sys/fs/cgroup/memory/A/F/memory.min
```

返回结果：

C节点memory.min=75M，memory.usage_in_bytes=50M。

D节点memory.min=25M，memory.usage_in_bytes=25M。

E节点memory.min=0，memory.usage_in_bytes=0。

B节点memory.usage_in_bytes=75M。

10.3 内核 cpu cgroup 的多级混部调度

需求背景

在业务混部场景中，Linux内核调度器需要为高优先级任务赋予更多的调度机会，并需要把低优先级任务对内核调度带来的影响降到最低。原有的在线、离线两级混部调度无法满足业务需求。

为解决此问题，HCE OS 2.0内核cpu cgroup支持多级混部调度，提供cgroup接口/sys/fs/cgroup/cpu/cpu.qos_level将任务调度级别扩展到5个级别，支持用户对每个cgroup组单独设置优先级。

约束与限制

内核cpu cgroup的多级混部调度基于5.10.0-60.18.0.50.r692_16.hce2.x86_64内核版本开发，当前cpu.qos_level仅支持cgroup-v1， 不支持cgroup-v2。

多级混部调度接口说明

cpu.qos_level的生效规则：

- CFS调度器自上而下逐层选择task_group，同一个父节点内的子节点之间cpu.qos_level生效。

- 子cgroup创建时默认继承父cgroup的cpu.qos_level，支持重新配置cpu.qos_level值。
- 同优先级的qos_level之间的资源竞争服从CFS调度器的策略。
- 同一个cpu上，qos_level < 0 的任务始终会被qos_level >= 0的任务无条件抢占，不受层级约束。

在调度高优先级任务时：

- 在线任务可无条件抢占离线任务，在多核调度时，在线任务可优先抢占其他核上的离线任务。超线程（Hyper Thread）场景，优先级为2的在线任务可驱逐SMT上的离线任务。
- 高优先级的任务被唤醒时获得一定的时间片加速，可立刻抢占低优先级的任务（忽略CFS的最小运行时间片），获得更好的低时延响应。

表 10-3 cpu.qos_level 接口说明

接口	说明
cpu.qos_level	配置cgroup的cpu优先级。取值类型为整数形式，取值范围 为[-2, 2]，默认值为0。 ● cpu.qos_level >= 0 标识cgroup组内任务为在线任务，在线任务可无条件抢占离线任务。 优先级 0 < 1 < 2，同为在线业务的任务，高优先级的在线相比低优先级的在线可获取更多的CPU资源抢占机会。 ● cpu.qos_level < 0 标识cgroup组内的任务为离线任务，-1优先级高于-2。 -1级别的任务比-2级别的任务可获得更多的CPU资源抢占机会。 父节点为离线任务时，子节点只能继承父节点的优先级，不支持修改为其他优先级。

接口示例

按如下所示创建3个cgroup节点A、B、C，配置并查看qos_level接口。

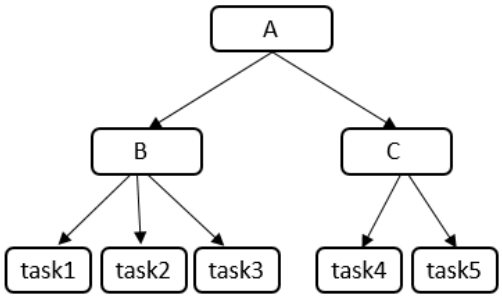


表 10-4 数据规划

cgroup组	cpu.qos_level
A	1
B	-2
C	2

1. 创建cgroup A及子节点B、C，依次设置A、B、C的cpu调度优先级为1、-2、2。

   cgroup A和cgroup C中的任务可无条件抢占cgroup B任务的CPU资源，cgroup C优先级大于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. 将task1、task2、task3进程加入cgroup B。

   task1、task2、task3进程加入cgroup B后，task1、task2、task3进程的cpu调度优先级为-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. 将task4、task5进程加入cgroup C。

   task4、task5进程加入cgroup C后，task4、task5进程的cpu调度优先级为2。

   ```
   echo $PID4 > /sys/fs/cgroup/cpu/A/C/tasks
   echo $PID5 > /sys/fs/cgroup/cpu/A/C/tasks
   ```

4. 查看cgroup B的cpu调度优先级及进程。

   ```
   [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. 查看cgroup C的cpu调度优先级及进程。

   ```
   [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 内核异常事件分析指南

背景说明

HCE OS运行时，不可避免的会出现一些内核事件，例如**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**、**MCE (Machine Check Exception)**、**fatal signal**、**warning**、**panic**。本节为您介绍这些内核事件的原理及触发方法。

soft lockup

soft lockup是内核检测CPU在一定时间内（默认20秒）没有发生调度切换时，上报的异常。

- 原理

  soft lockup利用Linux内核watchdog机制触发。内核会为每一个CPU启动一个优先级最高的FIFO实时内核线程watchdog，名称为watchdog/0、watchdog/1以此类推。这个线程会定期调用watchdog函数喂狗，默认每4秒执行一次。同时喂狗过后会重置一个hrtimer定时器在2倍的watchdog_thresh时间后到期。watchdog_thresh是内核参数，对应默认超时时间为20秒。

  在超时时间内，如果内核线程watchdog没被调度，hrtimer定时器到期，即触发内核打印类似如下的soft lockup异常。

  ```
  BUG: soft lockup - CPU#3 stuck for 23s! [kworker/3:0:32]
  ```

- 触发方法

  关闭中断或关闭抢占，软件执行死循环。

RCU(Read-Copy Update) stall

RCU stall是一种rcu宽限期内rcu相关内核线程没有得到调度的异常。

- 原理

  在RCU机制中，reader不用等待，可以任意读取数据，RCU记录reader的信息；writer更新数据时，先复制一份副本，在副本上完成修改，等待所有reader退出后，再一次性地替换旧数据。

  writer需要等所有reader都停止引用“旧数据”才能替换旧数据。这相当于给了这些reader一个优雅退出的宽限期，这个等待的时间被称为grace-period，简称GP。

  当reader长时间没有退出，writer等待的时间超过宽限期时，即上报RCU Stall。

- 触发方法

  内核在Documentation/RCU/stallwarn.txt文档列出了可能触发RCU stall的场景：cpu在rcu reader临界区一直循环，cpu在关闭中断或关闭抢占场景中一直循环等。

hung task

当内核检测到进程处于D状态超过设定的时间时，上报hung task异常。

- 原理

  进程其中一个状态是TASK_UNINTERRUPTIBLE，也叫D状态，处于D状态的进程只能被wake_up唤醒。内核引入D状态时，是为了让进程等待IO完成。正常情况下，IO正常处理，进程不应该长期处于D状态。

  hung task检测进程长期处于D状态的原理，内核会创建一个线程khungtaskd，用来定期遍历系统中的所有进程，检查是否存在处于D状态超过设置时长（默认120秒）的进程。如果存在这样的进程，则打印并上报相关警告和进程堆栈。如果配置了hung_task_panic（通过proc或内核启动参数配置），则直接发起panic。

- 触发方法

  创建内核线程，设成D状态，scheduler释放时间片。

global OOM

Linux的OOM killer特性是一种内存管理机制，在系统可用内存较少的情况下，内核为保证系统还能够继续运行下去，会选择杀掉一些进程释放掉一些内存。

- 原理

  通常oom_killer的触发流程是：内核为某个进程分配内存，当发现当前物理内存不够时，触发OOM。OOM killer遍历当前所有进程，根据进程的内存使用情况进行打分，然后从中选择一个分数最高的进程，终止进程释放内存。

  OOM killer的处理主要集中在mm/oom_kill.c，核心函数为out_of_memory，函数处理流程为：

  a. 通知系统中注册了oom_notify_list的模块释放一些内存，如果从这些模块中释放出了一些内存，直接结束oom killer流程；如果回收失败，进入下一步。
  b. 触发oom killer通常是由当前进程进行内存分配所引起。如果当前进程已经挂起了一个SIG_KILL信号或者正在退出，直接选中当前进程，终止进程释放内存；否则进入下一步。
  c. 检查panic_on_oom系统管理员的设置，决定OOM时是进行oom killer还是panic。如果选择panic，则系统崩溃并重启；如果选择oom killer，进入下一步。
  d. 进入oom killer，检查系统设置，系统管理员可设置终止当前尝试分配内存、引起OOM的进程或其它进程。如果选择终止当前进程，oom killer结束；否则进入下一步。
  e. 调用select_bad_process选中合适进程，然后调用oom_kill_process终止选中的进程。如果select_bad_process没有选出任何进程，内核进入panic。

- 触发方法

  执行占用大内存的程序，直到内存不足。

cgroup OOM

- 与global OOM的区别

  cgroup OOM与global OOM的内存范围不同。当FS cgroup组内进程使用内存超过了设置的上限，cgroup通过KILL相应的进程来释放内存。

- 触发方法

  执行占用大内存的程序，直到内存不足。

page allocation failure

page allocation failure是申请空闲页失败时，系统上报的错误。当程序申请某个阶数（order）的内存，但系统内存中，没有比申请阶数高的空闲页，即触发内核报错。

- 原理

  Linux使用伙伴系统（buddy system）内存分配算法。将所有的空闲页表（一个页表的大小为4K）分别链接到包含了11个元素的数组中，数组中的每个元素将大小相同的连续页表组成一个链表，页表的数量为1、2、4、8、16、32、64、128、256、512、1024，所以一次性可以分配的最大连续内存为1024个连续的4k页表，即4MB的内存。

  假设申请一个包括256个页表的内存，指定阶数order为6，系统会依次查找数组中的第9、10、11个链表，上一个为空，表示没有此阶数的空闲内存，查找下一个，直到最后一个链表。

如果所有链表均为空，申请失败，则内核上报错误page allocation failure。输出报错信息，描述申请阶数为6的内存页失败：

```
page allocation failure:order:6
```

- 触发方法

  用alloc_pages连续申请高阶数内存页（例如order=10），不释放，直到申请失败。

list corruption

list corruption是内核检查链表有效性失败的报错，报错类型分为list_add corruption和list_del corruption。

- 原理

  内核提供list_add和list_del，对传入的链表先检查链表的有效性（valid），检查通过后，修改链表增加或删除节点。如果检查链表失败，则上报corruption错误。检查和报错代码在内核lib/list_debug.c。

```
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);
```

  这种错误通常为内存异常操作导致，例如内存踩踏、内存损坏等。

- 触发方法

  用list.h的内核标准接口创建链表，非法修改链表节点的prev或next指针，再调用内核list_add/list_del接口。

Bad mm_struct

Bad mm_struct错误通常是由于内核中的一个或多个mm_struct数据结构被破坏或损坏所导致。

- 原理

  mm_struct是Linux内核中的一个重要数据结构，用于跟踪进程的虚拟内存区域。如果该数据结构被破坏，可能会导致进程崩溃或系统崩溃。这种错误通常内存异常导致，例如mm_struct区域的内存被踩踏、内存越界等。

- 触发方法

  无人为触发方法，当硬件错误，或者Linux系统内核代码错误时会触发Bad mm_struct。

I/O error

Linux I/O error报错通常表示输入/输出操作失败，在网卡、磁盘等IO设备驱动异常，或文件系统异常都可能打印这个错误。

- 原理

  错误原因取决于代码执行失败的条件。常见的触发异常的原因是硬件故障、磁盘损坏、文件系统错误、驱动程序问题、权限问题等。例如当系统尝试读取或写入磁盘上的数据时，如果发生错误，就会出现I/O错误。

- 触发方法

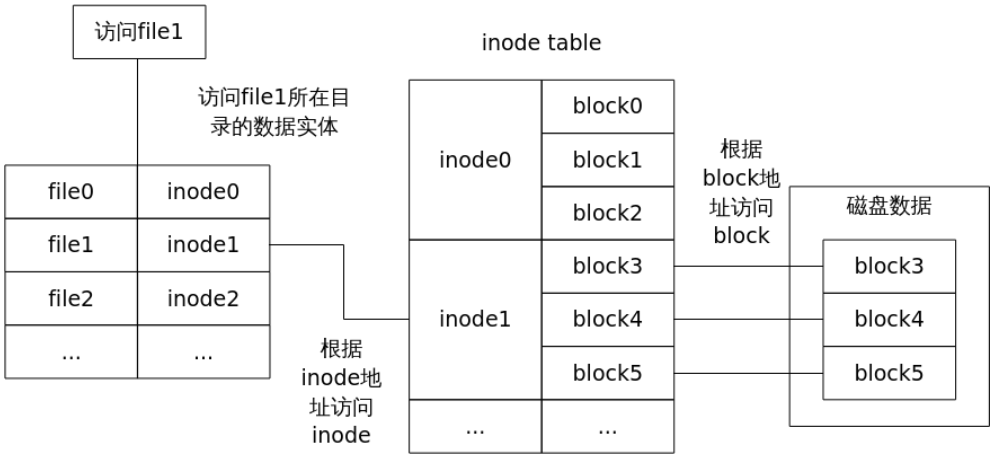
  系统读写磁盘过程，拔出磁盘，导致磁盘数据损坏。

EXT4-fs error

EXT4-fs error是由于ext4格式的文件系统中，文件节点的错误导致。

- 原理

  文件储存的最小存储单位叫做“扇区”（sector），连续多个扇区组成“块”（block）。inode节点储存文件的元信息，包括文件的创建者、创建日期、大小、属性、实际存储的数据块（block number）。EXT4格式的inode信息校验失败会触发EXT4-fs error。



  内核ext4校验使用checksum校验inode信息，当出现分区表错误、磁盘硬件损坏时，内核返回-EIO错误码，系统上报EXT4-fs error checksum invalid错误。

- 触发方法

  使用磁盘过程中强行拔盘，重新接入读盘。

MCE (Machine Check Exception)

Machine Check Exception (MCE) 是CPU发现硬件错误时触发的异常（exception），上报中断号是18，异常的类型是abort。

- 原理

导致MCE的原因主要有：总线故障、内存ECC校验错、cache错误、TLB错误、内部时钟错误等。不仅硬件故障会引起MCE，不恰当的BIOS配置、firmware bug、软件bug也有可能引起MCE。

MCE中断上报，操作系统检查一组寄存器称为Machine-Check MSR，根据寄存器的错误码执行对应的处理函数（函数实现依赖不同的芯片架构实现）。

- 触发方法

  无人为触发方法，当总线故障、内存ECC校验错、cache错误、TLB错误、内部时钟错误等时会触发MCE。

fatal signal

fatal signal指信号处理方式不能被设置为忽略或执行自定义处理函数的信号类型，包括SIGKILL、SIGSTOP、SIGILL等。

- 原理

  Linux信号（signal）机制，用于系统中进程间通讯，是一种异步的通知机制。当一个信号发送给一个进程，而操作系统中断了进程正常的控制流程时，任何非原子操作都将被中断。

  如果SIG符合条件，即为fatal信号：

```
#define sig_fatal(t, signr) \
    (!siginmask(signr, SIG_KERNEL_IGNORE_MASK|SIG_KERNEL_STOP_MASK) && \
```

- 触发方法

  用户态程序执行非法指令、kill -9杀进程。

warning

Warning是操作系统在运行时，检测到需要立即注意的内核问题（issue），而采取的上报动作，打印发生时的调用栈信息。上报后，系统继续运行。

- 原理

  Warning是通过调用WARN、WARN_ON、WARN_ON_ONCE等宏来触发的。

  导致Warning的原因有多种，需要根据调用栈回溯，找到调用Warning宏的具体原因。Warning宏并不会导致系统运行状态发生改变，也不提供处理Warning的指导。

- 触发方法

  根据系统调用构造Warnning条件。

panic

Kernel panic是指操作系统在监测到内部的致命错误，并无法安全处理此错误时采取的动作。内核触发到某种异常情况，运行kernel_panic函数，并尽可能把异常发生时获取的全部信息打印出来。

- 原理

  导致异常的原因多种多样，通过异常打印的调用信息，找到调用kernel_panic的原因。常见的原因包括内核堆栈溢出、内核空间的除0异常、内存访问越界、内核陷入死锁等。

- 触发方法

内核态读0地址。

11 XGPU 共享技术

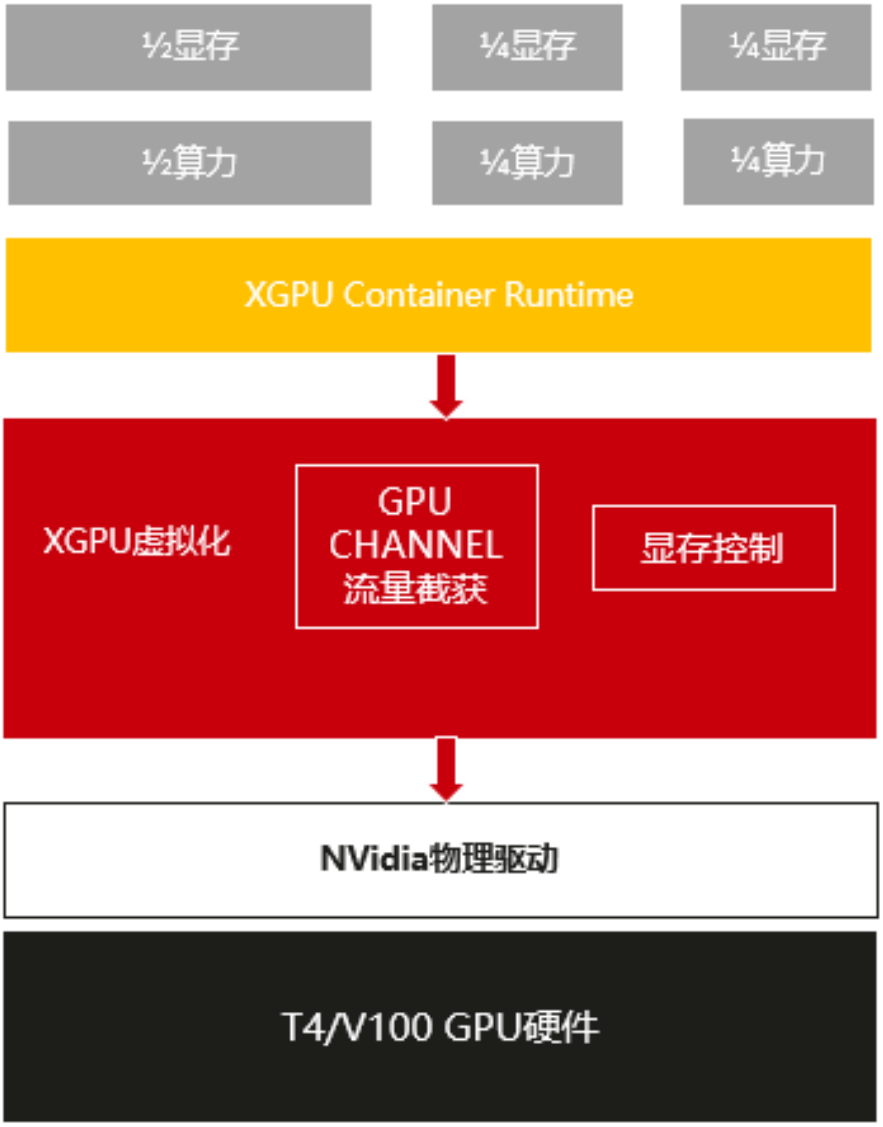
11.1 XGPU 共享技术概述

XGPU共享技术是华为云基于内核虚拟GPU开发的共享技术。XGPU服务可以隔离GPU资源，实现多个容器共用一张显卡，从而实现业务的安全隔离，提高GPU硬件资源的利用率并降低使用成本。

XGPU 共享技术架构

XGPU通过自研的内核驱动为容器提供虚拟的GPU设备，在保证性能的前提下隔离显存和算力，为充分利用GPU硬件资源进行训练和推理提供有效保障。您可以通过命令方便地配置容器内的虚拟GPU设备。

图 11-1 XGPU 共享技术架构图



产品优势

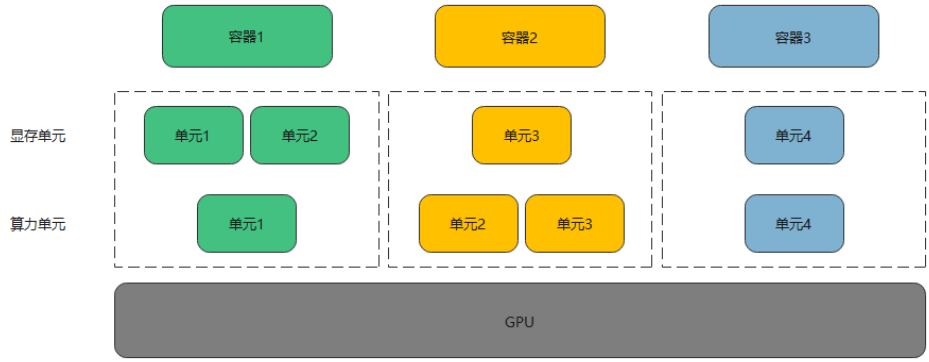
- 节约成本

  随着显卡技术的不断发展，单张GPU卡的算力越来越强，同时价格也越来越高。但在很多的业务场景下，一个AI应用并不需要一整张的GPU卡。XGPU的出现让多个容器共享一张GPU卡，从而实现业务的安全隔离，提升GPU利用率，节约用户成本。

- 可灵活分配资源

  XGPU实现了物理GPU的资源任意划分，您可以按照不同比例灵活配置。

  - 支持按照显存和算力两个维度划分，您可以根据需要灵活分配。



  - XGPU支持只隔离显存而不隔离算力的策略，同时也支持基于权重的算力分配策略。算力支持最小1%粒度的划分，推荐最小算力不低于4%。
- 兼容性好

  不仅适配标准的Docker和Containerd工作方式，而且兼容Kubernetes工作方式。
- 操作简单

  无需重编译AI应用，运行时无需替换CUDA库。

11.2 安装并使用 XGPU

本章节介绍如何安装和使用XGPU服务。

约束限制

- XGPU功能仅在Nvidia Tesla T4、V100上支持。
- HCE OS内核版本为5.10及以上版本。
- GPU实例已安装535.54.03版本的NVIDIA驱动。
- GPU实例已安装18.09.0-300或更高版本的docker。
- XGPU服务的隔离功能不支持以UVM的方式申请显存，即调用CUDA API cudaMallocManaged()，更多信息，请参见**NVIDIA官方文档**。请使用其他方式申请显存，例如调用cudaMalloc()等。
- 受GPU虚拟化技术的限制，容器内应用程序初始化时，通过nvidia-smi监测工具监测到的实时算力可能超过容器可用的算力上限。
- 当CUDA应用程序创建时，会在GPU卡上申请一小部分UVM显存（在Nvidia Tesla T4上大约为3 MiB），这部分显存属于管理开销，不受XGPU服务管控。
- 暂不支持同时在裸机环境以及该环境直通卡的虚拟机中同时使用

安装 XGPU 服务

安装XGPU服务请联系客服。

推荐您通过云容器引擎服务使用XGPU虚拟化服务，相关操作请参见**GPU虚拟化**。

XGPU 服务使用示例

影响XGPU服务的环境变量如下表所示，您可以在创建容器时指定环境变量的值。容器引擎可以通过XGPU服务获得算力和显存。

表 11-1 影响 XGPU 服务的环境变量

环境变量名称	取值类型	说明	示例
GPU_IDX	Integer	指定容器可使用的GPU显卡。	为容器分第一张显卡： GPU_IDX=0
GPU_CONTAINER_MEM	Integer	设置容器内可使用的显存大小，单位 MiB。	为容器分配的显存大小为5120MiB： GPU_CONTAINER_MEM=5120
GPU_CONTAINER_QUOTA_PERCENT	Integer	指定显卡算力分配百分比。 算力支持最小1%粒度的划分，推荐最小算力不低于4%。	为容器分配50%的算力比例： GPU_CONTAINER_QUOTA_PERCEN=50
GPU_POLICY	Integer	指定GPU使用的算力隔离的策略。 • 0：不隔离算力，即原生调度。 • 1：固定算力调度。 • 2：平均调度。 • 3：抢占调度。 • 4：权重抢占调度。 • 5：混合调度。 • 6：权重弱调度。 算力隔离策略示例详见**XGPU算力调度示例**。	设置算力隔离策略为固定算力调度： GPU_POLICY=1
GPU_CONTAINER_PRIORITY	Integer	指定容器的优先级。 • 0：低优先级 • 1：高优先级	创建高优先级容器： GPU_CONTAINER_PRIORITY=1

以nvidia的docker创建两个容器为例，介绍XGPU服务的使用方法，数据规划如下。

表 11-2 数据规划

参数	容器1	容器2	说明
GPU_IDX	0	0	指定两个容器使用第一张显卡。
GPU_CONTAINER_QUOTA_PERCENT	50	30	为容器1分配50%算力，为容器2分配30%算力。

参数	容器1	容器2	说明
GPU_CONTAINER_MEM	5120	1024	为容器1分配5120MiB显存，为容器2分配1024MiB显存。
GPU_POLICY	1	1	设置第一张显卡使用固定算力调度策略。
GPU_CONTAINER_PRIORITY	1	0	指定容器1为高优先级容器，容器2为低优先级容器。

配置示例：

```
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
```

查看 procfs 节点

XGPU服务运行时会在/proc/xgpu下生成并自动管理多个procfs节点，您可以通过procfs节点查看和配置XGPU服务相关的信息。下面介绍各procfs节点的用途。

1. 执行以下命令，查看节点信息。

   ```
   ls /proc/xgpu/
   0 container  version
   ```

   目录内容说明如下表所示：

目录	读写类型	说明
0	读写	XGPU服务会针对GPU实例中的每张显卡生成一个的目录，并使用数字作为目录名称，例如0、1、2。本示例中只有一张显卡，对应的目录ID为0。
container	读写	XGPU服务会针对运行在GPU实例中的每个容器生成一个的目录。
version	只读	XGPU的版本。

2. 执行以下命令，查看第一张显卡对应的目录内容。

   ```
   ls /proc/xgpu/0/
   max_inst meminfo policy quota utilization_line utilization_rate xgpu1 xgpu2
   ```

   目录内容说明如下表所示：

目录	读写类型	说明
max_inst	读写	用于设置容器的最大数量，取值范围为1~25。仅在没有容器运行的情况下可修改。
meminfo	只读	此显卡总共可用的显存大小。

目录	读写类型	说明
policy	读写	指定GPU使用的算力隔离的策略，默认值为1。 ● 0：不隔离算力，即原生调度。 ● 1：固定算力调度。 ● 2：平均调度。 ● 3：抢占调度。 ● 4：权重抢占调度。 ● 5：混合调度。 ● 6：权重弱调度。 算力隔离策略示例详见**XGPU算力调度示例**。
quota	只读	算力总权重。
utilization_line	读写	在离线混部的算力压制水位线。 当GPU整卡利用率超过该值时，在线容器完全压制离线容器，否则在线容器部分压制离线容器。
utilization_rate	只读	GPU整卡利用率。
xgpuIndex	读写	属于此显卡的xgpu子目录。 本示例中，属于第1张显卡的xgpu为xgpu1和xgpu2

3. 执行以下命令，查看container目录内容。

```
ls /proc/xgpu/container/
9418 9582
```

目录内容说明如下表所示：

目录	读写类型	说明
containerID	读写	容器的ID。 使用XGPU创建容器的时候分配的ID，每个容器对应一个ID。

4. 执行以下命令，查看containerID目录内容。

```
ls /proc/xgpu/container/9418/
xgpu1
ls /proc/xgpu/container/9582/
xgpu2
```

目录内容说明如下表所示：

目录	读写类型	说明
xgpuIndex	读写	属于此容器的xgpu子目录。 本示例中，属于容器9418的xgpu为xgpu1，属于容器9582的xgpu为xgpu2。

5. 执行以下命令，查看xgpuIndex目录内容。
```
ls /proc/xgpu/container/9418/xgpu1/
meminfo priority quota
```
目录内容说明如下表所示：

目录	读写类型	说明
meminfo	只读	此XGPU分配的可见显存大小和当前剩余可用显存大小。 如3308MiB/5120MiB, 64% free，指分配了5120MiB，剩余64%可使用。
priority	读写	用于设置容器的优先级，默认值为0。 ● 0：低优先级 ● 1：高优先级 该功能用于在线离线混合使用场景，高优先级容器可以抢占低优先级容器的算力。
quota	只读	此XGPU分配的算力百分比。 如50，指此XGPU分配了显卡50%的算力。

了解procfs节点的用途后，您可以在GPU实例中执行命令进行切换调度策略、查看权重等操作，示例命令如下表所示。

命令	效果
echo 1 > /proc/xgpu/0/policy	修改第一张显卡的调度策略为权重调度。
cat /proc/xgpu/container/$containerID/$xgpuIndex/meminfo	查看指定容器里xgpu分配的显存大小。
cat /proc/xgpu/container/$containerID/$xgpuIndex/quota	查看指定容器里xgpu分配的算力权重。
cat /proc/xgpu/0/quota	查看第一张显卡上剩余可用的算力权重。
echo 20 > /proc/xgpu/0/max_inst	设置第一张显卡最多可以创建20个容器。
echo 1 > /proc/xgpu/container/$containerID/$xgpuIndex/priority	设置指定容器里xgpu的优先级为高优先级。
echo 40 > /proc/xgpu/0/utilization_line	设置第一张显卡的在离线混部算力压制水位线为40%。

升级 XGPU 服务

XGPU服务采用冷升级的方式。

1. 关闭所有运行中的容器。
   docker ps -q | xargs -I {} docker stop {}
2. 升级xgpu的rpm包。
   rpm -U hce_xgpu

卸载 XGPU 服务

1. 关闭所有运行中的容器。
   docker ps -q | xargs -I {} docker stop {}
2. 卸载xgpu的rpm包。
   rpm -e hce_xgpu

通过 xgpu-smi 工具监控容器

您可以通过xgpu-smi工具查看XGPU容器的相关信息，包括容器ID、算力使用、分配情况、显存使用及分配情况等。

xgpu-smi的监控展示信息如下所示：

```
+=============================================================================+
|  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 |
+-----------------------------------------------------------------------------+
```

您可使用**xgpu-smi -h**命令查看xgpu-smi工具的帮助信息。

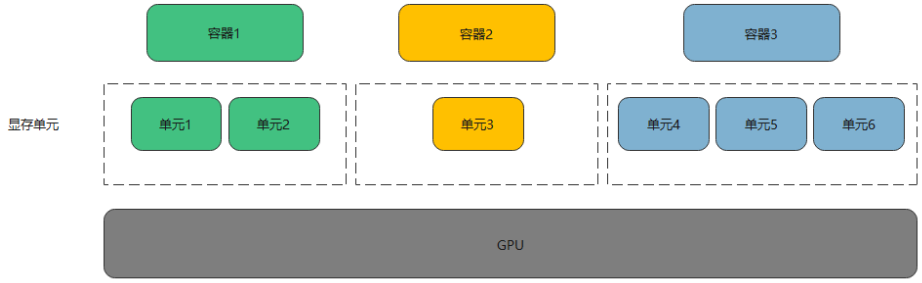
```
[root@localhost ~]#
[root@localhost ~]# xgpu-smi -h
xgpu-smi provides monitoring information about xgpu containers.
The data is presented in either a plain text or a json format, via stdout or a file.

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 XGPU 算力调度示例

当使用XGPU服务创建XGPU时，XGPU服务会按照最大容器数量（**max_inst**）为每张显卡设置时间片（X ms）用于为容器分配GPU算力，以单元1、单元2…单元N表示。本节max_inst以20为例，介绍使用不同调度策略时对算力的调度示例。

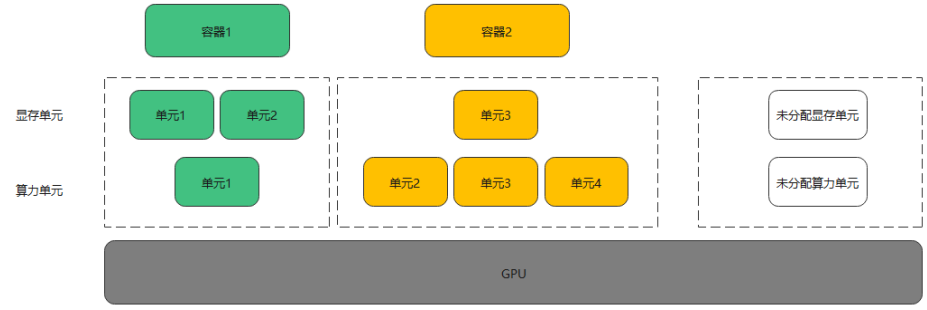
原生调度（policy=0）

原生调度表示使用NVIDIA GPU本身的算力调度方式。在原生调度策略下XGPU只用来做显存的隔离。



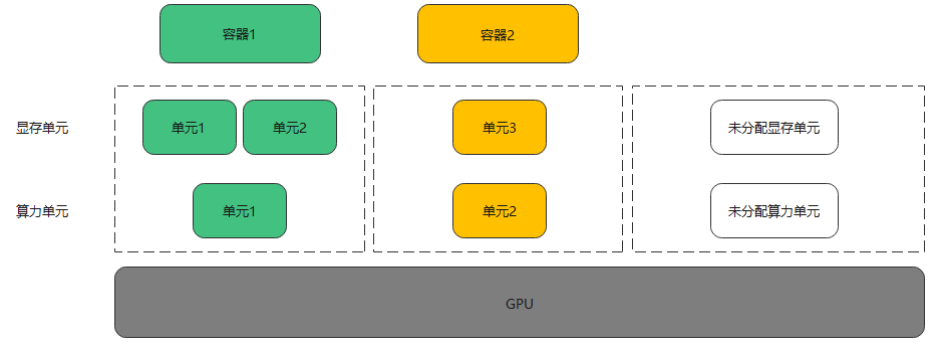
固定算力调度（policy=1）

固定算力调度表示以固定的算力百分比为容器分配算力。例如为容器1和容器2分别分配5%和15%的算力，如下图所示。



平均调度（policy=2）

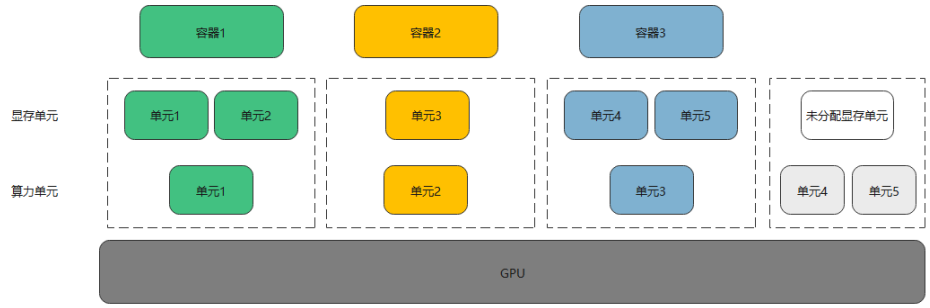
平均调度表示每个容器固定获得1/max_inst的算力。以max_inst=20为例，每个容器固定获得1/max_inst，即5%的算力，如下图所示。



抢占调度（policy=3）

抢占调度表示每个容器固定获得1个时间片，XGPU服务会从算力单元1开始调度。但如果某个算力单元没有分配给某个容器，或者容器内没有进程打开GPU设备，则跳过调度切换到下一个时间片。图中灰色部分的算力单元表示被跳过不参与调度。

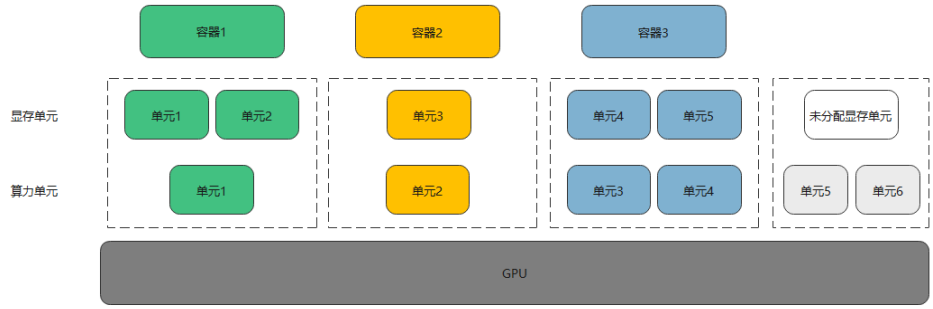
本例中容器1、2、3占用的实际算力百分比均为33.33%。



权重抢占调度（policy=4）

权重抢占调度表示按照每个容器的算力比例为容器分配时间片。XGPU服务会从算力单元1开始调度，但如果某个算力单元没有分配给某个容器，则跳过调度切换到下一个时间片。例如为容器1、2、3分别分配5%、5%、10%的算力，则容器1、2、3分别占用1、1、2个算力单元。图中灰色部分的算力单元表示被跳过不参与调度。

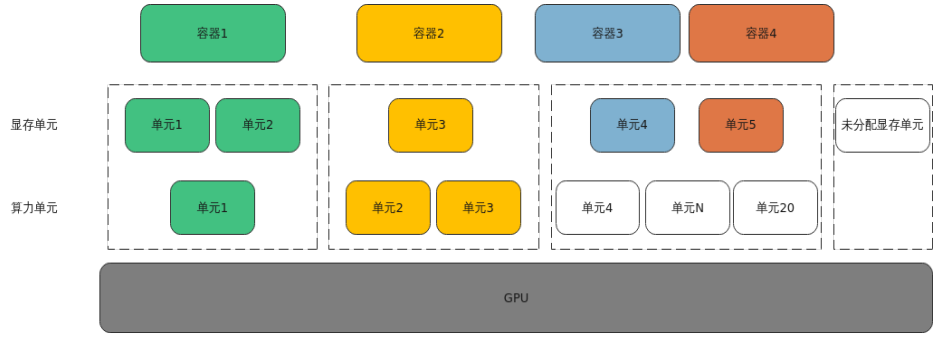
本例中容器1、2、3占用的实际算力百分比为25%、25%、50%。



混合调度（policy=5）

混合调度表示单张GPU卡支持单显存隔离和算力显存隔离类型。其中算力显存隔离的容器其隔离效果同固定算力（policy=1）完全一致，单显存隔离的容器共享算力显存隔离的容器分配后剩余的GPU算力。以max_inst=20为例，容器1、2为算力显存隔离容器，其分配的算力分别为5%、10%，容器3、4为单显存隔离的容器，则容器1、2分别占用1、2个算力单元，容器3、4共享剩余17个算力单元。此外，当容器2中没有进程打开GPU设备时，则容器1、2分别占用1、0个算力单元，容器3、4共享剩余19个算力单元。

在混合调度下，根据GPU_CONTAINER_QUOTA_PERCENT是否为0来区分容器是否开启算力隔离，GPU_CONTAINER_QUOTA_PERCENT为0的所有容器共享GPU的空闲算力。



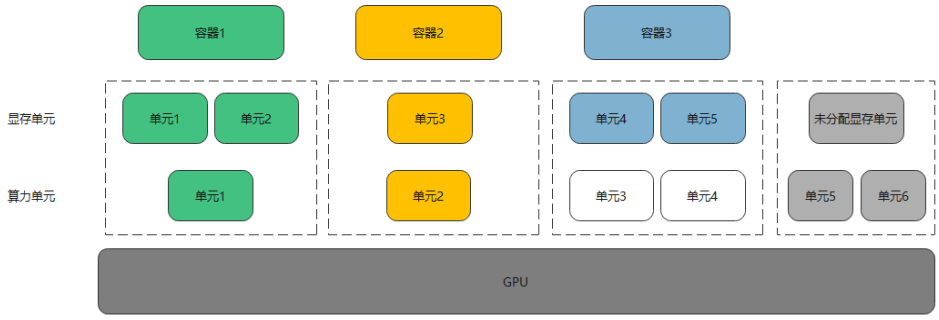
说明

混合调度策略不支持高优先级容器。

权重弱调度（policy=6）

权重弱调度表示按照每个容器的算力比例为容器分配时间片，隔离性弱于权重抢占调度。XGPU服务会从算力单元1开始调度，但如果某个算力单元没有分配给某个容器，或者容器内没有进程打开GPU设备，则跳过调度切换到下一个时间片。例如为容器1、2、3分别分配5%、5%、10%的算力，则容器1、2、3分别占用1、1、2个算力单元。图中白色部分的算力单元表示容器3的空闲算力，图中白色部分和灰色部分的算力单元表示被跳过不参与调度。

本例中容器1、2、3占用的实际算力百分比为50%、50%、0%。



说明

权重弱调度涉及空闲算力的抢占和抢回，因此容器在空闲和忙碌之间切换时会影响其他容器的算力，该算力波动属于正常情况。当某个容器从空闲切换到忙碌时，其抢回算力的时延不超过100ms。

12 HCE OS 的 REPO 源配置与软件安装

HCE OS采用RPM包形式管理软件，并且提供了与系统配套的官方REPO源来发布软件包及其更新。您可通过dnf/yum命令实现常见的软件管理功能，包括安装、升级、卸载等。

官方 repo 源配置

通过弹性云服务器购买的HCE OS默认镜像，在**/etc/yum.repos.d/hce.repo**文件中会默认配置官方repo源。以HCE 2.0版本为例，其内容如下：

```
[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
```

其中各字段含义如下：

- name：对repo源的描述。
- baseurl：仓库所在的服务器地址，支持http://、ftp://、file://三种格式。
- enabled：是否启用该软件仓库，1表示启用，0表示禁用。
- gpgcheck：是否进行gpg校验，1表示启用校验，0表示禁用校验。
- gpgkey：公钥保存的地址，用于gpg校验。

⚠ 注意

修改该文件可能会对系统的软件安装、升级产生影响，不建议修改该文件。

第三方 repo 源配置

如果要新增第三方repo源，可按下述过程进行配置（以openEuler社区的镜像源为例）：

1. 在**/etc/yum.repos.d/**目录新增**openEuler.repo**文件（名称可以自定义，文件后缀需以.repo结尾）。使用**vim /etc/yum.repo.d/openEuler.repo**命令进行编辑。
2. 配置仓库名字，如[openEuler]，仓库名必须唯一，可根据实际情况进行调整。
3. 配置name选项，如openEuler repository，表示仓库的具体描述，可根据实际情况进行调整。
4. 配置baseurl选项，此处为：**https://repo.openeuler.org/openEuler-22.03-LTS/OS/x86_64/**，表示软件包从该链接获取，具体可参考openEuler或者对应repo提供者的官方说明。
5. 配置gpgcheck选项，为1表示对安装的软件包进行gpg校验。
6. 配置enabled选项，为1表示启用该repo源。
7. 配置gpgkey选项，此处为：**https://repo.openeuler.org/openEuler-22.03-LTS/OS/x86_64/RPM-GPG-KEY-openEuler**，表示gpg校验使用的公钥来源于该链接。

最终**openEuler.repo**文件效果如下：

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

须知

可以通过配置中的priority字段控制repo源的优先级。如果优先使用HCE OS默认源，可在hce.repo配置中都加上priority=1（数值越小优先级越高），然后在第三方源配置中加上priority=2，数值根据实际情况进行调整。本文仅为示例，完整的openEuler仓库配置请参考**HCE OS获取openEuler扩展软件包**。

说明

如果要升级软件包，可参考用户指南**更新HCE OS系统和RPM包**。

yum/dnf 常见使用方式

HCE 1.1仅支持通过yum命令进行软件管理相关操作，HCE 2.0同时支持yum与dnf命令。常用的软件管理相关的命令如下：

功能	yum命令	dnf命令	示例
安装软件包	yum install <软件包>	dnf install <软件包>	安装gcc：yum install gcc
卸载软件包	yum remove <软件包>	dnf remove <软件包>	卸载gcc：yum remove gcc

功能	yum命令	dnf命令	示例
列出已安装的软件包	yum list installed	dnf list installed	列出系统所有的包：yum list installed
搜索软件包	yum search <软件包>	dnf search <软件包>	在repo源中搜索gcc包：yum search gcc
查询软件包信息	yum info <软件包>	dnf info <软件包>	查询gcc软件包信息：yum info gcc

13 修订记录

发布日期	更新说明
2024-03-30	第八次正式发布 新增 **HCE OS的REPO源配置与软件安装** 修改 **迁移操作**，补充备份相关描述。
2023-12-30	第七次正式发布 修改 • **XGPU算力调度示例**，补充权重弱调度（policy=6）内容。 • **动态加速**，补充动态应用加速工具字符交互界面指令内容。
2023-11-17	第六次正式发布 修改 • **制作Docker镜像并启动容器**，修改约束限制以及完善操作步骤内容。 • **评估软件兼容性**，补充ARM架构软件兼容性扫描示例内容。 • **迁移操作**，迁移工具安装包版本号更新以及修改系统迁移说明内容。 • **约束限制**，修改约束限制。 • **约束限制**，修改约束限制。 • **安装并使用XGPU**，修改约束限制。

发布日期	更新说明
2023-09-27	第五次正式发布 新增 • **内核异常事件分析指南** • **动态加速** 修改 • **约束限制**，兼容性支持ko格式。 • **评估软件兼容性**，支持扫描嵌套目录。 • **工具概述**，新增支持EulerOS 2.9/2.10和HCE OS 2.0 ARM操作系统的兼容性评估。 • **冲突包列表**，更新冲突包列表。 • **约束限制**，增加约束限制。 • **概述**，增加动态加速介绍，以及静态加速和动态加速的优缺点。 • **配置文件**，增加动态加速配置文件及参数说明。 • **安装并使用XGPU**、**XGPU算力调度示例**，增加混部调度相关描述。
2023-08-03	第四次正式发布 修改 • **准备工作**，增加motd_setup参数。 • **OSMT命令帮助信息**，增加--nocheck参数。 • **HCE OS获取openEuler扩展软件包**，更新操作步骤。 • **概述**、**静态加速**，支持**hce-wae-auto**命令优化应用程序，增加相关内容。 • **XGPU共享技术概述**，算力支持力度由5%优化至1%，修改对应描述；增加“XGPU功能仅在Nvidia Tesla T4、V100、A100、A30、A40、A800上支持”的约束限制。 • **安装并使用XGPU**，增加GPU_POLICY、GPU_CONTAINER_PRIORITY、max_inst、priority参数及相关内容；算力隔离的策略由0~1优化至0~4；增加“通过xgpu-smi工具监控容器”内容。 增加 • **制作Docker镜像并启动容器** • **配置文件** • **XGPU算力调度示例**

发布日期	更新说明
2023-03-30	第三次正式发布。 增加 • **Pod带宽管理工具** • **内核功能与接口** • **内核memory的OOM进程控制策略** • **内核memory的多级内存回收策略** • **内核cpu cgroup的多级混部调度** 修改 • **冲突包列表**，补充CentOS 8.4/8.5的冲突包列表。 • **静态加速**，更新操作步骤。 删除 • 应用加速工具中的“常见问题”章节 • perf工具
2023-01-17	第二次正式发布。 增加 • **约束限制** • **升级后续操作** • **工具类** 修改 • **冲突包列表**，增加CentOS 8.3冲突包列表。 • **使用dnf或yum命令升级**，更新**dnf list updates**命令，补充操作步骤3。 • **约束限制**，补充约束限制。 • **版本升级和回退**，优化操作步骤。 • **准备工作**，更新osmt.conf配置，补充参数项及描述。 • **osmt update命令更新**，优化操作步骤。 • **osmt-agent服务自动更新**，优化操作步骤。 • **回退RPM包**，补充回退说明。
2022-11-30	第一次正式发布。

