

鲲鹏加速引擎

开发者指南

文档版本

07

发布日期

2020-12-14



华为技术有限公司



版权所有 © 华为技术有限公司 2020。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI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，华为公司对本文档内容不做任何明示或默示的声明或保证。

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 产品简介	1
2 逻辑架构	2
3 安装、升级和卸载	4
3.1 安装加速引擎软件	4
3.1.1 安装简介	4
3.1.2 安装前准备	7
3.1.3 安装 OpenSSL	10
3.1.4 通过 rpm 包安装加速引擎软件	11
3.1.5 通过 deb 包安装加速引擎软件	15
3.1.6 通过源码方式安装加速引擎软件	18
3.2 安装后测试	21
3.2.1 OpenSSL 加速器引擎性能测试	21
3.2.2 zlib 压缩库测试	23
3.3 升级加速引擎软件	23
3.3.1 通过 rpm 包升级加速引擎软件	24
3.3.2 通过 deb 包升级加速引擎软件	25
3.4 卸载加速引擎软件	27
3.4.1 卸载通过 rpm 包安装的加速引擎软件	27
3.4.2 卸载通过 deb 包安装的加速引擎软件	28
3.4.3 卸载通过源码方式安装的加速引擎软件	29
4 使用案例	31
4.1 加速引擎的应用	31
4.1.1 加速引擎使用说明	31
4.1.2 KAE 引擎使用示例代码	31
4.1.3 通过 OpenSSL 配置文件 openssl.cnf 使用 KAE 引擎	32
4.1.4 KAE 引擎加速 Nginx 应用	33
4.1.5 基于 dm-crypt 的透明分区/磁盘加密	33
4.1.6 加速引擎在 KVM 虚拟机中的使用	36
4.1.7 加速引擎在 docker 中的使用	37
4.1.8 MD5 硬件加速调优	39
4.2 zlib 加速库的使用	39
5 常用操作	40

5.1 加速器日志查询.....	40
6 故障处理.....	42
6.1 初始化失败.....	42
6.2 安装完加速器引擎之后，查找不到加速器设备.....	43
6.3 升级加速器驱动失败.....	44
6.4 操作系统安装完 OpenSSL 新版本后提示相关接口符号未找到.....	44
A 附录.....	46
A.1 术语.....	46
A.2 缩略语.....	47
A.3 修订记录.....	47

1 产品简介

加速引擎是TaiShan 200服务器基于Kunpeng 920处理器提供的硬件加速解决方案，包含了KAE加解密、KAEzip分别用于加速SSL/TLS应用和数据压缩，可以显著降低处理器消耗，提高处理器效率。此外，加速引擎对应用层屏蔽了其内部实现细节，用户通过OpenSSL、zlib标准接口即可以实现快速迁移现有业务。

KAE加解密是鲲鹏加速引擎的加解密模块，使用鲲鹏硬加速模块实现RSA/SM3/SM4/DH/MD5/AES算法，结合无损用户态驱动框架，提供高性能对称加解密、非对称加解密算法能力，兼容openssl1.1.1a及其之后版本，支持同步&异步机制。

目前支持主要支持以下算法：

- 摘要算法SM3/MD5，支持异步模型。
- 对称加密算法SM4，支持异步模型，支持CTR/XTS/CBC/ECB/OFB模式。
- 对称加密算法AES，支持异步模型，支持ECB/CTR/XTS/CBC模式。
- 非对称算法RSA，支持异步模型，支持Key Sizes 1024/2048/3072/4096。
- 密钥协商算法DH，支持异步模型，支持Key Sizes 768/1024/1536/2048/3072/4096。

KAEzip是鲲鹏加速引擎的压缩模块，使用鲲鹏硬加速模块实现deflate算法，结合无损用户态驱动框架，提供高性能gzip/zlib格式压缩接口。

- 支持zlib/gzip数据格式，符合RFC1950/RFC1952标准规范。
- 支持deflate算法。
- 支持同步模式。
- 单处理器（Kunpeng 920）最大压缩带宽7GB/s，最大解压带宽8GB/s。
- 支持的压缩比 ≈ 2 与zlib1.2.11接口保持一致。

通过加速引擎可以实现不同场景下应用性能的提升，例如在Web服务应用场景下，通过KAE加速RSA算法可以加速握手连接；在智能安防场景下，通过KAE加速SM4对称加解密算法加速视频流数据存储；在分布式存储场景下，通过zlib加速库加速数据压缩和解压。

📖 说明

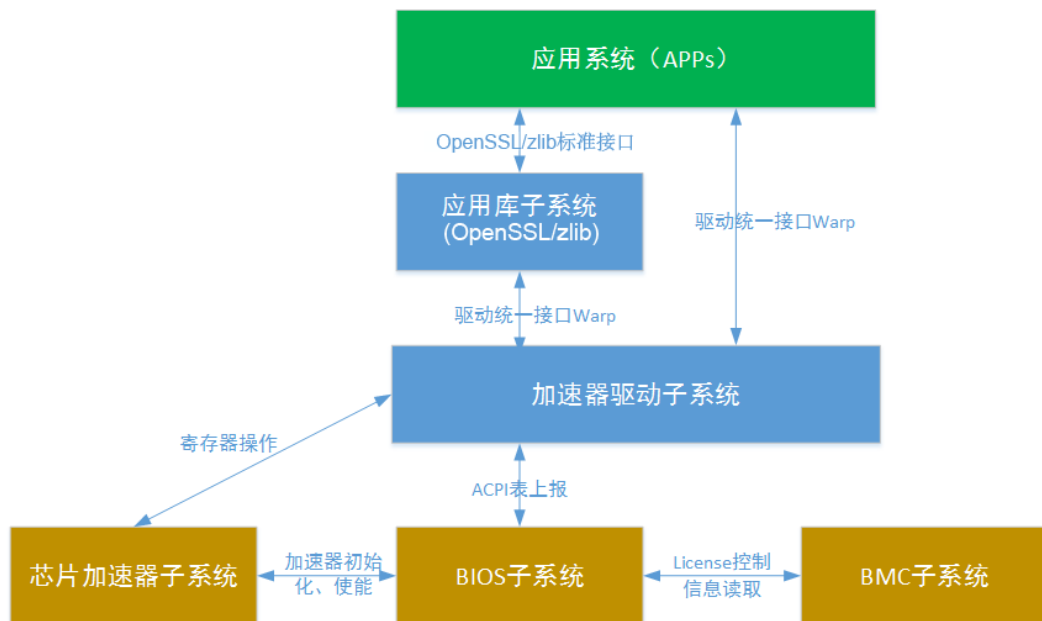
KAE加解密和KAEzip的具体相关操作和应用见下文统一描述。

2 逻辑架构

系统逻辑架构如图2-1所示。

- 芯片加速器子系统、BIOS子系统和BMC子系统为TaiShan硬件产品自带子系统。
- 加速器驱动子系统，向上层提供各子加速器模块统一的驱动接口，是本系统的核心子系统。
- 应用库子系统（OpenSSL/zlib），向上层应用提供标准接口。
- 应用系统（APPs），指上层应用系统，包括大数据应用、Web应用等，属于用户层面系统。

图 2-1 系统逻辑架构



子系统名称	功能描述
芯片加速器子系统	集成在Kunpeng 920处理器中，提供加速器的能力，对上层提供寄存器接口。该子系统是加速器的硬件实现，不直接开放给客户。

子系统名称	功能描述
BIOS子系统	单板BIOS软件系统，主要负责根据License决定对加速器哪些模块初始化，并上报加速器ACPI表到内核（加速器驱动子系统处理）。
BMC子系统	服务器BMC软件系统，在这里主要负责对加速器License的管理。
加速器驱动子系统	本系统的核心，向上层提供各子加速器模块统一的驱动接口。
应用库子系统	应用库子系统包括OpenSSL加速器引擎、zlib替代库等，向上层提供标准接口。
应用系统	用户系统，通过调用应用库子系统或驱动子系统实现加速器的功能。

3 安装、升级和卸载

- 3.1 安装加速引擎软件
- 3.2 安装后测试
- 3.3 升级加速引擎软件
- 3.4 卸载加速引擎软件

3.1 安装加速引擎软件

3.1.1 安装简介

在表3-1中选择其中一种方式进行安装。

表 3-1 加速引擎软件安装方式

安装方式	安装说明	当前支持系统	优缺点
rpm安装	为了方便用户使用，提供了部分商用OS的rpm方式安装包。rpm可用于管理Linux各项套件的程序，可以使用rpm --help 命令获得更多帮助。	<ul style="list-style-type: none">● CentOS 7.6 4.14.0-115.el7a.0.1.aarch64 version● SUSE 15.1 4.12.14-195-default.aarch64 version● EulerOS 2.8 4.19.36-vhulk1907.1.0.h410.eulerosv2r8.aarch64 version● NeoKylin7.6 4.14.0-115.5.1.el7a.06.aarch64 version● BCLinux-R7-U6-Server-aarch64 version● Kylin 4.0.2 (juniper) 4.15.0-70-generic version● Kylin release 4.0.2 (SP2) 4.19.36-vhulk1907.1.0.h403.ky4.aarch64 version● UniKylin Linux release 3(Core) 4.18.0-80.ky3.kb21.hw.aarch64 version	优点：安装后可以直接使用，不需要做编译及安装等操作。 缺点：支持范围有限，支持指定的操作系统。

安装方式	安装说明	当前支持系统	优缺点
dpkg安装	为了方便用户使用，提供了部分商用OS的dpkg方式安装包。dpkg可以使用dpkg --help命令获得更多帮助。	<ul style="list-style-type: none">• Kylin 4.0.2 (juniper) 4.15.0-70-generic version• Kylin release 4.0.2 (SP2) 4.19.36-vhulk1907.1.0.h403.ky4.aarch64 version• Ubuntu 18.04.1 LTS 4.15.0-29-generic version	优点：安装后可以直接使用，不需要做编译及安装等操作。 缺点：支持范围有限，支持指定的操作系统。

安装方式	安装说明	当前支持系统	优缺点
源码安装	通用的源码安装方式，使用configure进行编译及安装配置，使用make进行源码编译，使用make install进行安装。	<ul style="list-style-type: none">• CentOS 7.6 4.14.0-115.el7a.0.1.aarch64 version• SUSE 15.1 4.12.14-195-default.aarch64 version• EulerOS 2.8 4.19.36-vhulk1907.1.0.h410.eulerosv2r8.aarch64 version• NeoKylin7.6 4.14.0-115.5.1.el7a.06.aarch64 version• BCLinux-R7-U6-Server-aarch64 version• Kylin 4.0.2 (juniper) 4.15.0-70-generic version• Kylin release 4.0.2 (SP2) 4.19.36-vhulk1907.1.0.h403.ky4.aarch64 version• UniKylin Linux release 3(Core) 4.18.0-80.ky3.kb21.hw.aarch64 version• Ubuntu 18.04.1 LTS 4.15.0-29-generic version	优点：支持范围广，支持大部分Linux OS，支持修改源码进行编译及安装。 缺点：操作复杂，需要做一些额外的配置。

3.1.2 安装前准备

环境要求

- TaiShan 200服务器，开启加速引擎功能

📖 说明

非虚拟化场景使用加速器建议关闭SMMU，具体操作请参考《[BIOS 参数参考 \(鲲鹏920处理器\)](#)》。

- CPU: Kunpeng 920
- iBMC版本: V365及以上
- BIOS版本: V105及以上
- 操作系统:
 - CentOS 7.6 4.14.0-115.el7a.0.1.aarch64 version
 - SUSE 15.1 4.12.14-195-default.aarch64 version
 - EulerOS 2.8 4.19.36-vhulk1907.1.0.h410.eulerosv2r8.aarch64 version
 - NeoKylin7.6 4.14.0-115.5.1.el7a.06.aarch64 version
 - BCLinux-R7-U6-Server-aarch64 version
 - Kylin 4.0.2 (juniper) 4.15.0-70-generic version
 - Kylin release 4.0.2 (SP2) 4.19.36-vhulk1907.1.0.h403.ky4.aarch64 version
 - UniKylin Linux release 3(Core) 4.18.0-80.ky3.kb21.hw.aarch64 version
 - Ubuntu 18.04.1 LTS 4.15.0-29-generic version
- 远程SSH登录工具已经在本地安装

获取 License

安装鲲鹏加速器引擎之前需要先安装相应的License，License安装成功之后，操作系统才能识别到加速器设备。

📖 说明

TaiShan K系列服务器硬件加速引擎已默认开启，无需申请License。

具体License申请使用操作可参考《[华为服务器 iBMC 许可证 使用指导](#)》，文档链接如下：<https://support.huawei.com/enterprise/zh/management-software/ibmc-pid-8060757?category=operation-maintenance>。

查看操作系统是否有加速器设备的方法可以通过lspci 命令进行查看，如下所示。

📖 说明

不同的操作系统lspci查出的加速器描述信息可能不同，除了通过关键字进行过滤，用户还可以查看是否存在如下加速器的sbdf号信息。

```
[root@localhost modules]# lspci | grep HPRE
79:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd. HiSilicon HPRE Engine (rev 21)
b9:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd. HiSilicon HPRE Engine (rev 21)
[root@localhost modules]#
[root@localhost modules]# lspci | grep RDE
78:01.0 RAID bus controller: Huawei Technologies Co., Ltd. HiSilicon RDE Engine (rev 21)
b8:01.0 RAID bus controller: Huawei Technologies Co., Ltd. HiSilicon RDE Engine (rev 21)
[root@localhost modules]#
[root@localhost modules]# lspci | grep ZIP
75:00.0 Processing accelerators: Huawei Technologies Co., Ltd. HiSilicon ZIP Engine (rev 21)
b5:00.0 Processing accelerators: Huawei Technologies Co., Ltd. HiSilicon ZIP Engine (rev 21)
[root@localhost modules]#
[root@localhost modules]# lspci | grep SEC
76:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd. HiSilicon SEC Engine (rev
```

```
21)
b6:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd. HiSilicon SEC Engine (rev
21)
[root@localhost modules]#
```

获取软件

📖 说明

根据实际安装方式，选择表3-2或表3-3其中一个表格中的软件包下载即可。

表 3-2 加速引擎各种 OS 类型相关的 rpm 或 deb 软件包

软件包名称	软件包说明	获取方法
uacce-版本号-1.OS类型.aarch64.rpm uacce-版本号-1.OS类型.aarch64.deb	统一加速器框架，包含内容：uacce.ko、hisi_qm.ko 内核模块	<ul style="list-style-type: none"> • https://github.com/kunpengcompute/KAE • https://github.com/kunpengcompute/KAEdriver • https://github.com/kunpengcompute/KAEzip
hisi_hpre-版本号-1.OS类型.aarch64.rpm hisi_hpre-版本号-1.OS类型.aarch64.deb	依赖：uacce rpm包 包含内容：hisi_hpre.ko内核模块 支持：RSA/DH算法	
hisi_sec2-版本号-1.OS类型.aarch64.rpm hisi_sec2-版本号-1.OS类型.aarch64.deb	依赖：uacce rpm包 包含内容：hisi_sec2.ko内核模块 支持：AES/MD5/SM3/SM4 算法	
hisi_rde-版本号-1.OS类型.aarch64.rpm hisi_rde-版本号-1.OS类型.aarch64.deb	依赖：uacce rpm包 包含内容：hisi_rde.ko内核模块 支持：FlexEC算法	
hisi_zip-版本号-1.OS类型.aarch64.rpm hisi_zip-版本号-1.OS类型.aarch64.deb	依赖：uacce rpm包 包含内容：hisi_zip.ko内核模块 支持：zlib/gzip	
libwd-版本号-1.OS类型.aarch64.rpm libwd-版本号-1.OS类型.aarch64.deb	包含内容：libwd.so动态链接库 提供接口给KAE引擎	
libkae-版本号-1.OS类型.aarch64.rpm libkae-版本号-1.OS类型.aarch64.deb	依赖：libwd rpm包 包含内容：libkae.so动态库 支持：SM3/SM4/RSA/AES/MD5/DH等算法	

软件包名称	软件包说明	获取方法
libkaezip-版本号-1.OS类型.aarch64.rpm libkaezip-版本号-1.OS类型.aarch64.deb	依赖: libwd rpm/deb包 包含内容: libkaezip.so动态库 支持: 压缩解压算法	

表 3-3 加速引擎源码包关键目录说明

源码包名称	源码包说明	获取方法
KAE	该目录包含全部加速引擎 OpenSSL相关代码, 包含 KAE引擎。	<ul style="list-style-type: none"> • https://github.com/kunpengcompute/KAE
KAEdriver	该目录包含加速引擎驱动代码; 包含uacce、hpre、zip、rde、sec2等内核模块以及用户态驱动libwd。	<ul style="list-style-type: none"> • https://github.com/kunpengcompute/KAEdriver
KAEzip	该目录包含了zlib的patch, 该patch文件实现了硬件解压和压缩算法。	<ul style="list-style-type: none"> • https://github.com/kunpengcompute/KAEzip

3.1.3 安装 OpenSSL

前提条件

- 已安装与系统版本对应的kernel-devel
使用如下命令查询当前内核版本号
 - `uname -r`
- 已安装perl、bzip2
使用如下命令查询perl、bzip2的版本号
 - `perl --version`
 - `bzip2 --version`
- 已安装gcc、make工具, 不同gcc版本下呈现出的性能数据存在差异, 推荐使用7.4.1及以上版本, make推荐使用3.82及以上版本
使用如下命令查询gcc、make的版本号
 - `gcc --version`
 - `make --version`
- 已安装automake、autoconf、libtool相关软件
使用如下命令查询automake、autoconf、libtool的版本号
 - `automake --version`
 - `autoconf --version`

```
- libtool --version
```

尚未安装的软件可通过以下说明进行操作。

📖 说明

不同操作系统默认安装软件命令不同，CentOS与EulerOS使用yum工具，SUSE使用zypper工具。请确认系统已连接网络或配置了镜像源，以SUSE为例，安装gcc命令如下所示，其余操作系统的安装方式相同。

```
mount -o loop /dev/sr0 /mnt #挂载操作镜像
zypper ar file:///mnt local #配置本地zypper镜像源
zypper install gcc
```

安装步骤

步骤1 使用SSH远程登录工具，将OpenSSL源码包拷贝到自定义路径下，OpenSSL版本为1.1.1a或以上版本。

📖 说明

OpenSSL源码包可从<https://www.openssl.org/source/old/1.1.1/>下载。

步骤2 使用SSH远程登录工具，进入Linux操作系统命令行界面。

步骤3 在下载好的OpenSSL源码目录下，执行如下命令（以下均使用默认路径/usr/local进行安装）。

```
./config -Wl,-rpath,/usr/local/lib //该命令会根据编译平台及环境自动生成Makefile文件，可以通过./config --prefix 指定安装路径，-Wl,-rpath参数指定OpenSSL运行时依赖libcrypto、libssl库的路径。
make
make install
```

OpenSSL默认安装在/usr/local下，更加具体的安装指导请参考OpenSSL源码目录下的README文档。

----结束

设置环境变量

通过以下命令导出环境变量：如果用户指定安装路径，则下面/usr/local应根据实际安装路径进行修改。

```
export OPENSSL_ENGINES=/usr/local/lib/engines-1.1 #该环境变量为了指定挂载到OpenSSL中的引擎路径
```

安装后检查

步骤1 执行cd命令，进入到/usr/local/bin目录下。

步骤2 执行./openssl version命令查看版本信息。

```
[root@localhost bin]# ./openssl version
OpenSSL 1.1.1a 20 Nov 2018 //显示该格式内容说明安装成功
```

----结束

3.1.4 通过 rpm 包安装加速引擎软件

前提条件

- 请在指定版本的操作系统上进行rpm方式安装，若无法安装请按源码方式进行安装。

- rpm工具能正常使用。
- OpenSSL 1.1.1a或以上版本已正确安装，具体请参见[3.1.3 安装OpenSSL](#)。

安装步骤

📖 说明

hisi_hpre,hisi_sec2,hisi_rde,hisi_zip驱动软件包依赖于uacce软件包，libkaezip、libkae引擎软件包依赖libwd软件包。

- 如果仅加速RSA/DH算法建议只需要安装uacce、hisi_hpre、libwd、libkae软件包。
 - 如果仅加速AES/MD5/SM3/SM4算法建议只需要安装uacce、hisi_sec2、libwd、libkae软件包。
 - 如果仅加速zlib库的压缩算法建议只需要安装uacce、hisi_zip、libwd、libkaezip软件包。
- 相关rpm提供算法功能请参考[表3-2](#)。

步骤1 使用SSH远程登录工具，将加速引擎软件包拷贝到自定义路径下。

步骤2 使用SSH远程登录工具，以root帐号进入Linux操作系统命令行界面。

步骤3 执行rpm -ivh命令安装加速驱动软件包，如OpenSSL已按上述默认方式进行安装，则可以通过rpm -ivh *.rpm 安装所有加速引擎软件包，否则请按[步骤4](#)进行安装。

```
[root@localhost home]# rpm -ivh *.rpm
Preparing...                               ##### [100%]
checking installed modules
Updating / installing...
 1:uacce-1.0.1-1.centos7.6                 ##### [ 14%]
modules installed
 2:libwd-1.0.1-1.centos7.6                 ##### [ 29%]
 3:libkae-1.0.1-1.centos7.6               ##### [ 43%]
checking installed modules
 4:hisi_hpre-1.0.1-1.centos7.6            ##### [ 57%]
modules installed
checking installed modules
 5:hisi_rde-1.0.1-1.centos7.6             ##### [ 71%]
modules installed
checking installed modules
 6:hisi_sec2-1.0.1-1.centos7.6            ##### [ 86%]
modules installed
checking installed modules
 7:hisi_zip-1.0.1-1.centos7.6             ##### [100%]
modules installed
```

步骤4 如果已经按[步骤3](#)安装rpm软件，请跳过该步骤。以下示例为安装uacce-1.0.1软件包。

```
[root@localhost home]# rpm -ivh uacce-1.0.1-1.centos7.6.aarch64.rpm
Preparing...                               ##### [100%]
checking installed modules
Updating / installing...
 1:uacce-1.0.1-1.centos7.6                 ##### [100%]
modules installed
```

依次安装hisi_hpre，hisi_sec2，hisi_rde，hisi_zip驱动软件包和libwd，libkae引擎软件包。安装libkae引擎软件包时需要通过--prefix指定OpenSSL引擎的路径，如下：

```
[root@localhost home]# rpm -ivh libkae-1.0.1-1.centos7.6.aarch64.rpm --prefix=/usr/local/openssl/lib/engines-1.1 #即OpenSSL安装路径选择为/usr/local/openssl时
Preparing...                               ##### [100%]
Updating / installing...
 1:libkae-1.0.1-1.centos7.6               ##### [100%]
```

步骤5 查看rpm软件是否已正常安装到系统内。

```
[root@localhost rpm]# rpm -ql uacce
/lib/modules/4.14.0-115.el7a.0.1.aarch64/extra/hisi_qm.co
```



```
/lib/modules/4.14.0-115.el7a.0.1.aarch64/extra/uacce.ko
[root@localhost rpm]#
[root@localhost rpm]# rpm -ql hisi_sec2 hisi_hpre hisi_rde hisi_zip
/lib/modules/4.14.0-115.el7a.0.1.aarch64/extra/hisi_sec2.ko
/etc/modprobe.d/hisi_sec2.conf
/lib/modules/4.14.0-115.el7a.0.1.aarch64/extra/hisi_hpre.ko
/etc/modprobe.d/hisi_hpre.conf
/lib/modules/4.14.0-115.el7a.0.1.aarch64/extra/hisi_rde.ko
/etc/modprobe.d/hisi_rde.conf
/lib/modules/4.14.0-115.el7a.0.1.aarch64/extra/hisi_zip.ko
/etc/modprobe.d/hisi_zip.conf
[root@localhost rpm]# [root@localhost rpm]# ls -al /lib/modules/`uname -r`/extra
total 3672
drwxr-xr-x. 2 root root 118 Nov 17 21:56 .
drwxr-xr-x. 7 root root 4096 Nov 17 21:56 ..
-rw-r--r--. 1 root root 681104 Nov 12 17:32 hisi_hpre.ko
-rw-r--r--. 1 root root 618888 Nov 12 17:32 hisi_qm.ko
-rw-r--r--. 1 root root 844728 Nov 12 17:32 hisi_rde.ko
-rw-r--r--. 1 root root 729304 Nov 12 17:32 hisi_sec2.ko
-rw-r--r--. 1 root root 396784 Nov 12 17:32 hisi_zip.ko
-rw-r--r--. 1 root root 467160 Nov 12 17:32 uacce.ko
[root@localhost rpm]# ls -al /etc/modprobe.d/
total 36
drwxr-xr-x. 2 root root 140 Nov 17 21:56 .
drwxr-xr-x. 127 root root 8192 Nov 17 22:14 ..
-rw-r--r--. 1 root root 166 Oct 30 2018 firewalld-sysctls.conf
-rw-r--r--. 1 root root 44 Nov 17 21:56 hisi_hpre.conf
-rw-r--r--. 1 root root 43 Nov 17 21:56 hisi_rde.conf
-rw-r--r--. 1 root root 61 Nov 17 21:56 hisi_sec2.conf
-rw-r--r--. 1 root root 43 Nov 17 21:56 hisi_zip.conf
-rw-r--r--. 1 root root 674 Jul 4 2018 tuned.conf
[root@localhost rpm]#
```

步骤6 重启系统或通过命令行手动依次加载加速器驱动到内核，并查看是否加载成功。如果操作系统是SUSE请先按说明进行操作。

```
[root@localhost rpm]# lsmod | grep uacce
[root@localhost rpm]# modprobe uacce
[root@localhost rpm]# modprobe hisi_sec2 #加载hisi_sec2驱动时将根据/etc/modprobe.d/hisi_sec2.conf 下的
配置文件加载到内核
[root@localhost rpm]# modprobe hisi_hpre #加载hisi_hpre驱动时将根据/etc/modprobe.d/hisi_hpre.conf 下的
配置文件加载到内核
[root@localhost rpm]# modprobe hisi_rde #加载hisi_rde驱动时将根据/etc/modprobe.d/hisi_rde.conf 下的配
置文件加载到内核
[root@localhost rpm]# modprobe hisi_zip #加载hisi_zip驱动时将根据/etc/modprobe.d/hisi_zip.conf 下的配置
文件加载到内核
[root@localhost rpm]# lsmod | grep uacce
uacce 36864 3 hisi_sec2,hisi_qm,hisi_hpre,hisi_rde,hisi_zip
```

📖 说明

- 在SUSE操作系统内加载外部驱动需要先配置“/etc/modprobe.d/10-unsupported-modules.conf”，设置“allow_unsupported_modules”的值为“1”。
- 加速器引擎代码包含内核驱动、用户态驱动、KAE引擎和zlib库四个模块。其中内核驱动与用户态驱动为必选项，KAE引擎与zlib库按实际需求选择安装。

---结束

安装后检查

步骤1 使用rpm -qa | grep查看加速引擎软件包。

```
[root@localhost root]# rpm -qa uacce hisi_sec2 hisi_hpre hisi_zip hisi_rde
hisi_rde-1.0.1-1.centos7.6.aarch64 //显示该格式内容说明安装成功
hisi_sec2-1.0.1-1.centos7.6.aarch64
uacce-1.0.1-1.centos7.6.aarch64
hisi_hpre-1.0.1-1.centos7.6.aarch64
hisi_zip-1.0.1-1.centos7.6.aarch64
```

步骤2 查看虚拟文件系统下对应的加速器设备：

```
[root@localhost rpm]# ls -al /sys/class/uacce/
total 0
lrwxrwxrwx. 1 root root 0 Nov 14 03:45 hisi_hpre-2 -> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/
uacce/hisi_hpre-2
lrwxrwxrwx. 1 root root 0 Nov 14 03:45 hisi_hpre-3 -> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/
uacce/hisi_hpre-3
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_rde-4 -> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-4
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_rde-5 -> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-5
lrwxrwxrwx. 1 root root 0 Nov 14 08:39 hisi_sec-0 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/
uacce/hisi_sec-0
lrwxrwxrwx. 1 root root 0 Nov 14 08:39 hisi_sec-1 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/
uacce/hisi_sec-1
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_zip-6 -> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/
uacce/hisi_zip-6
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_zip-7 -> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/
uacce/hisi_zip-7
```

步骤3 查看加速器是否生效，以验证RSA性能为例，命令如下：

```
[root@localhost rpm]# cd /usr/local/bin/

[root@localhost bin]# ./openssl speed rsa2048

sign verify sign/s verify/s

rsa 2048 bits 0.001381s 0.000035s 724.1 28601.0

[root@localhost bin]# ./openssl speed -engine kae rsa2048

engine "kae" set.

sign verify sign/s verify/s

rsa 2048 bits 0.000175s 0.000021s 5730.1 46591.8
```

通过RSA性能命令可以看到指定kae引擎之后，RSA的性能得到明显提升。另外，除上述方法，在跑RSA性能命令过程中，可以在新的终端上同时查看hisi_hpre设备的硬件队列资源情况如下，相同的，SM3/SM4算法可以查看hisi_sec2的硬件队列消耗情况。

```
[root@localhost ~]# cat /sys/class/uacce/hisi_hpre-*/attrs/available_instances

256
```

255 #从256变为255，说明RSA算消耗了HRPE加速器一个硬件单元队列，说明加速器引擎已生效

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

步骤4 （可选）查看zlib库加速器是否生效，用户如果没有安装hisi_zip软件可以忽略该步骤。通过ldd命令查看zlib加速库是否链接到libwd和libkaezip，如果有返回如下信息，说明zlib加速库安装成功。同样的，用户的进程也可以通过ldd 命令查看是否使用libwd和libkaezip。

```
[root@localhost /]# ldd /usr/local/kaezip/lib/libz.so.1.2.11
linux-vdso.so.1 => (0x0000ffff80280000)
libc.so.6 => /lib64/libc.so.6 (0x0000ffff80080000)
libwd.so.1 => /lib64/libwd.so.1 (0x0000ffff80040000)
/lib/ld-linux-aarch64.so.1 (0x0000ffff80290000)
libkaezip.so => /usr/local/kaezip/lib/libkaezip.so (0x0000ffff80830000)
[root@localhost /]#
```

----结束

3.1.5 通过 deb 包安装加速引擎软件

前提条件

- 请在指定版本的操作系统上通过deb包进行安装，若无法安装请按源码方式进行安装。
- dpkg工具能正常使用。
- OpenSSL 1.1.1a或以上版本已正确安装，具体请参见[3.1.3 安装OpenSSL](#)。

安装步骤

📖 说明

hisi_hpre,hisi_sec2,hisi_rde,hisi_zip驱动软件包依赖于uacce软件包；libkae,libkaezip引擎软件包依赖libwd软件包。

- 如果仅加速RSA和DH算法建议只需要安装uacce、hisi_hpre、libwd、libkae软件包。
 - 如果仅加速AES/MD5/SM3/SM4国密算法建议只需要安装uacce、hisi_sec2、libwd、libkae软件包。
 - 如果仅加速zlib库的压缩算法建议只需要安装uacce、hisi_zip、libwd、libkaezip软件包。
- 各软件包提供算法功能请参考[3.1.1-表 加速器软件包安装方式](#)。

步骤1 使用SSH远程登录工具，将加速引擎软件包拷贝到自定义路径下。

步骤2 使用SSH远程登录工具，以root帐号进入Linux操作系统命令行界面。

步骤3 执行dpkg -i命令安装加速驱动软件包，例如可以通过dpkg -i *.deb安装所有加速引擎软件包。

```
[root@localhost home]# dpkg -i *.deb
```

步骤4 （可选）如果已经按**步骤3**安装deb软件，请跳过该步骤。如果需要单独安装软件包，请参考以下安装uacce-1.3.1软件包示例进行安装。

```
[root@localhost home]# dpkg -i uacce-1.3.1-1.ubuntu18.04.arm64.deb
uacce modules installed
```

依次安装hisi_hpre，hisi_sec2，hisi_rde，hisi_zip驱动软件包和libwd，libkae，libkaezip引擎软件包。安装libkae或者libkaezip引擎软件包时需要先安装libwd软件包。

步骤5 查看deb软件是否已正常安装到系统内。

```
[root@localhost deb]# dpkg -L uacce
./
/lib
/lib/modules
/lib/modules/4.15.0-70-generic
/lib/modules/4.15.0-70-generic/extra
/lib/modules/4.15.0-70-generic/extra/uacce.ko
/lib/modules/4.15.0-70-generic/extra/hisi_qm.ko
[root@localhost deb]# dpkg -L hisi-sec2 hisi-hpre hisi-rde hisi-zip
./
/lib
/lib/modules
/lib/modules/4.15.0-70-generic
/lib/modules/4.15.0-70-generic/extra
/lib/modules/4.15.0-70-generic/extra/hisi_sec2.ko
/etc
/etc/modprobe.d
/etc/modprobe.d/hisi_sec2.conf
./
```

```
/lib
/lib/modules
/lib/modules/4.15.0-70-generic
/lib/modules/4.15.0-70-generic/extra
/lib/modules/4.15.0-70-generic/extra/hisi_hpre.ko
/etc
/etc/modprobe.d
/etc/modprobe.d/hisi_hpre.conf

./
/lib
/lib/modules
/lib/modules/4.15.0-70-generic
/lib/modules/4.15.0-70-generic/extra
/lib/modules/4.15.0-70-generic/extra/hisi_rde.ko
/etc
/etc/modprobe.d
/etc/modprobe.d/hisi_rde.conf

./
/lib
/lib/modules
/lib/modules/4.15.0-70-generic
/lib/modules/4.15.0-70-generic/extra
/lib/modules/4.15.0-70-generic/extra/hisi_zip.ko
/etc
/etc/modprobe.d
/etc/modprobe.d/hisi_zip.conf
[root@localhost deb]# ls -al /lib/modules/`uname -r`/extra
total 348
drwxr-xr-x 2 root root 4096 Apr 15 17:12 .
drwxr-xr-x 6 root root 4096 Apr 15 17:12 ..
-rw-r--r-- 1 root root 49080 Apr 15 09:15 hisi_hpre.ko
-rw-r--r-- 1 root root 85728 Apr 15 09:15 hisi_qm.ko
-rw-r--r-- 1 root root 59720 Apr 15 09:15 hisi_rde.ko
-rw-r--r-- 1 root root 65536 Apr 15 09:15 hisi_sec2.ko
-rw-r--r-- 1 root root 42600 Apr 15 09:15 hisi_zip.ko
-rw-r--r-- 1 root root 40408 Apr 15 09:15 uacce.ko
[root@localhost deb]# ls -al /etc/modprobe.d/
total 52
drwxr-xr-x 2 root root 4096 Apr 15 17:12 .
drwxr-xr-x 94 root root 4096 Apr 15 09:15 ..
-rw-r--r-- 1 root root 325 Jan 28 2018 blacklist-ath_pci.conf
-rw-r--r-- 1 root root 1667 Nov 13 2018 blacklist.conf
-rw-r--r-- 1 root root 210 Jan 28 2018 blacklist-firewire.conf
-rw-r--r-- 1 root root 697 Jan 28 2018 blacklist-framebuffer.conf
-rw-r--r-- 1 root root 583 Jan 28 2018 blacklist-rare-network.conf
-rw-r--r-- 1 root root 45 Apr 15 09:15 hisi_hpre.conf
-rw-r--r-- 1 root root 44 Apr 15 09:15 hisi_rde.conf
-rw-r--r-- 1 root root 61 Apr 15 09:15 hisi_sec2.conf
-rw-r--r-- 1 root root 44 Apr 15 09:15 hisi_zip.conf
-rw-r--r-- 1 root root 347 Jan 28 2018 iwlwifi.conf
-rw-r--r-- 1 root root 379 Jul 2 2018 mdadm.conf
```

步骤6 重启系统或通过命令行手动依次加载加速器驱动到内核，并查看是否加载成功。

```
[root@localhost deb]# lsmod | grep uacce
[root@localhost deb]# modprobe uacce
[root@localhost deb]# modprobe hisi_sec2 #加载hisi_sec2驱动时将根据/etc/modprobe.d/hisi_sec2.conf 下的
配置文件加载到内核
[root@localhost deb]# modprobe hisi_hpre #加载hisi_hpre驱动时将根据/etc/modprobe.d/hisi_hpre.conf 下的
配置文件加载到内核
[root@localhost deb]# modprobe hisi_rde #加载hisi_rde驱动时将根据/etc/modprobe.d/hisi_rde.conf 下的配
置文件加载到内核
[root@localhost deb]# modprobe hisi_zip #加载hisi_zip驱动时将根据/etc/modprobe.d/hisi_zip.conf 下的配置
文件加载到内核
[root@localhost deb]# lsmod | grep uacce
uacce          28672  1 hisi_qm
[root@localhost deb]# lsmod | grep hisi_qm
```

```
hisi_qm          65536  4  hisi_sec2,hisi_zip,hisi_hpre,hisi_rde
uacce           28672  1  hisi_qm
```

📖 说明

加速器引擎代码包含内核驱动、用户态驱动、KAE引擎和zlib库四个模块。其中内核驱动与用户态驱动为必选项，KAE引擎与zlib库按实际需求选择安装。

----结束

安装后检查

步骤1 使用dpkg -s查看加速引擎软件包。

```
[root@localhost root]# dpkg -s uacce hisi-sec2 hisi-hpre hisi-zip hisi-rde
Package: uacce
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Unified/User-space-access-intended Accelerator Framework.
URL: https://support.huawei.com

Package: hisi-sec2
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Huawei Hisilicon SEC Accelerator Driver.
URL: https://support.huawei.com

Package: hisi-hpre
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Huawei Hisilicon HPRE Accelerator Driver.
URL: https://support.huawei.com

Package: hisi-zip
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Huawei Hisilicon ZIP Accelerator Driver.
URL: https://support.huawei.com

Package: hisi-rde
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Huawei Hisilicon RDE Accelerator Driver.
URL: https://support.huawei.com
```

步骤2 查看/sys/目录下对应的加速器设备：

```
[root@localhost deb]# ls -al /sys/class/uacce/  
total 0  
drwxr-xr-x 2 root root 0 Apr 15 17:42 .  
drwxr-xr-x 65 root root 0 Apr 15 16:49 ..  
lrwxrwxrwx 1 root root 0 Apr 15 17:42 hisi_hpre-0 -> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/  
uacce/hisi_hpre-0  
lrwxrwxrwx 1 root root 0 Apr 15 17:42 hisi_hpre-1 -> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/  
uacce/hisi_hpre-1  
lrwxrwxrwx 1 root root 0 Apr 15 17:42 hisi_rde-4 -> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-4  
lrwxrwxrwx 1 root root 0 Apr 15 17:42 hisi_rde-5 -> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-5  
lrwxrwxrwx 1 root root 0 Apr 15 17:42 hisi_sec2-2 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/  
uacce/hisi_sec2-2  
lrwxrwxrwx 1 root root 0 Apr 15 17:42 hisi_sec2-3 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/  
uacce/hisi_sec2-3  
lrwxrwxrwx 1 root root 0 Apr 15 17:42 hisi_zip-6 -> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/  
uacce/hisi_zip-6  
lrwxrwxrwx 1 root root 0 Apr 15 17:42 hisi_zip-7 -> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/  
uacce/hisi_zip-7
```

步骤3 通过OpenSSL命令验证加速器是否生效，以验证RSA性能为例，命令如下：

```
[root@localhost deb]# cd /usr/local/bin/  
[root@localhost bin]# ./openssl speed rsa2048  
sign verify sign/s verify/s  
rsa 2048 bits 0.001381s 0.000035s 724.1 28601.0  
[root@localhost bin]# ./openssl speed -engine kae rsa2048  
engine "kae" set.  
sign verify sign/s verify/s  
rsa 2048 bits 0.000175s 0.000021s 5730.1 46591.8
```

通过RSA性能命令可以看到指定kae引擎之后，RSA的性能得到明显提升。另外，除上述方法，在跑RSA性能命令过程中，可以在新的终端上同时查看hpre设备的硬件队列资源情况如下：

```
[root@localhost ~]# cat /sys/class/uacce/hisi_hpre-*/attrs/available_instances  
256  
255 #从256变为255，说明RSA算消耗了HRPE加速器一个硬件单元队列，说明加速器引擎已生效
```

步骤4 （可选）查看zlib库加速引擎是否生效，用户如果没有安装hisi_zip软件可以忽略该步骤。通过ldd命令查看zlib加速库是否链接到libwd和libkaezip，如果有返回如下信息，说明zlib加速库安装成功。同样的，用户的进程也可以通过ldd命令查看是否使用libwd和libkaezip。

```
[root@localhost /]# ldd /usr/local/kaezip/lib/libz.so.1.2.11  
linux-vdso.so.1 => (0x0000ffff80280000)  
libc.so.6 => /lib64/libc.so.6 (0x0000ffff80080000)  
libwd.so.1 => /lib64/libwd.so.1 (0x0000ffff80040000)  
/lib/ld-linux-aarch64.so.1 (0x0000ffff80290000)  
libkaezip.so => /usr/local/kaezip/lib/libkaezip.so (0x0000ffff80830000)  
[root@localhost /]#
```

----结束

3.1.6 通过源码方式安装加速引擎软件

前提条件

- 安装前的系统环境已满足[3.1.2 安装前准备](#)中的环境要求。
- OpenSSL 1.1.1a或以上版本已正确安装，具体请参见[3.1.3 安装OpenSSL](#)。

安装步骤

步骤1 使用远程登录工具，将加速引擎源码包拷贝到自定义路径下。

📖 说明

代码包含内核驱动、用户态驱动、基于OpenSSL的KAE引擎和zlib库四个模块。
其中内核驱动与用户态驱动为必选项，KAE引擎与zlib库按实际需求选择安装。

步骤2 使用SSH远程登录工具，以root帐号进入Linux操作系统命令行界面。

步骤3 安装内核驱动。

在下载好的KAEdriver目录下，执行如下命令：

```
cd kae_driver
make
make install
```

加速器驱动编译生成uacce.ko、hisi_qm.ko、hisi_sec2.ko、hisi_hpre.ko、hisi_zip.ko、hisi_rde.ko；安装路径为：
lib/modules/`uname -r`/extra

📖 说明

由于SUSE及CentOS内核目录为lib/modules/`uname -r`/，驱动安装的目录为lib/modules/`uname -r`/extra（`uname -r`命令获取当前运行内核信息）。如果其他OS不是该目录，需要修改Makefile文件中install指定的内核路径。

install:

\$(shell mkdir -p /lib/modules/`uname -r`/extra)修改为\$(shell mkdir -p 内核路径/extra)

步骤4 安装用户态驱动。

在KAEdriver目录下，编译安装Warpdrive驱动开发库。

```
cd warpdriver
sh autogen.sh
./configure
make
make install
```

其中./configure 可以加--prefix选项用于指定加速器用户态驱动需要安装的位置，用户态驱动动态库文件为libwd.so。Warpdrive默认安装路径为/usr/local，动态库文件在/usr/local/lib下。

📖 说明

KAE引擎需要使用到OpenSSL的动态库与Warpdrive的动态库。Warpdrive源码安装路径选择需要与OpenSSL安装路径保持一致，使得KAE引擎可以通过LD_LIBRARY_PATH能够同时找到这两个动态库。

步骤5 重启系统或通过命令行手动依次加载加速器驱动到内核，并查看是否加载成功。如果操作系统是SUSE请按先说明进行操作。

```
[root@localhost rpm]# lsmod | grep uacce
[root@localhost rpm]# modprobe uacce
[root@localhost rpm]# modprobe hisi_sec2 #加载hisi_sec2驱动时将根据/etc/modprobe.d/hisi_sec2.conf 下的
配置文件加载到内核
[root@localhost rpm]# modprobe hisi_hpre #加载hisi_hpre驱动时将根据/etc/modprobe.d/hisi_hpre.conf 下的
配置文件加载到内核
[root@localhost rpm]# modprobe hisi_rde #加载hisi_rde驱动时将根据/etc/modprobe.d/hisi_rde.conf 下的配
置文件加载到内核
[root@localhost rpm]# modprobe hisi_zip #加载hisi_zip驱动时将根据/etc/modprobe.d/hisi_zip.conf 下的配置
文件加载到内核
[root@localhost rpm]# lsmod | grep uacce
uacce          36864  3 hisi_sec2,hisi_qm,hisi_hpre,hisi_rde,hisi_zip
```

📖 说明

在SUSE操作系统内加载外部驱动需要先配置/etc/modprobe.d/10-unsupported-modules.conf。设置allow_unsupported_modules 的值为1。

步骤6 编译安装加速器KAE引擎。

```
cd KAE
chmod +x configure
./configure
make clean && make
make install
```

其中./configure 可以加--prefix选项用于指定KAE 引擎的安装路径，KAE引擎动态库文件为libkae.so。

推荐通过默认方式安装KAE引擎。默认安装路径为/usr/local，动态库文件在/usr/local/lib/engines-1.1下。

📖 说明

如果libwd和OpenSSL安装方式不是按默认进行安装，则可以通过以下命令指定OpenSSL和libwd的安装路径：

```
./configure --openssl_path=/usr/local/openssl --wd_path=/usr/local/libwd
```

其中/usr/local/openssl和/usr/local/libwd是OpenSSL和libwd的一个示例安装路径。

步骤7 （可选）编译安装zlib压缩库。

编译安装之前需要从[zlib官网](#)下载zlib-1.2.11.tar.gz，并拷贝到KAEzip/open_source。

```
cd KAEzip
sh setup.sh install
```

zlib加速库安装在/usr/local/kaezip。

----结束

安装后检查

步骤1 执行cd命令，进入到/usr/local/lib目录或者用户自定义安装目录下。

步骤2 执行ls -al命令，查看软连接状态。

命令脚本和信息回显如下所示：

```
[root@localhost ~]# ls -al /usr/local/lib/ |grep libwd
lrwxrwxrwx. 1 root root    14 Jun 25 11:16 libwd.so -> libwd.so.1.0.1
lrwxrwxrwx. 1 root root    14 Jun 25 11:16 libwd.so.0 -> libwd.so.1.0.1
-rwxr-xr-x. 1 root root 137280 Jun 24 11:37 libwd.so.1.0.1 //软连接及so存在，说明libwd安装成功
[root@localhost ~]# ls -al /usr/local/lib/engines-1.1/
lrwxrwxrwx. 1 root root    48 Jun 25 11:21 kae.so -> /usr/local/openssl/lib/engines-1.1/kae.so.1.0.1
lrwxrwxrwx. 1 root root    48 Jun 25 11:21 kae.so.0 -> /usr/local/openssl/lib/engines-1.1/kae.so.1.0.1
-rwxr-xr-x. 1 root root 212192 Jun 24 11:37 kae.so.1.0.1
//软连接及so存在，说明KAE引擎安装成功
[root@localhost ~]# ll /usr/local/zlib/lib
-rw-r--r-. 1 root root 161162 Jul 10 12:54 libz.a
lrwxrwxrwx. 1 root root    14 Jul 10 12:54 libz.so -> libz.so.1.2.11
lrwxrwxrwx. 1 root root    14 Jul 10 12:54 libz.so.1 -> libz.so.1.2.11
-rwxr-xr-x. 1 root root 146656 Jul 10 12:54 libz.so.1.2.11
//libz的软连接和so文件存在，说明zlib库安装成功
```

步骤3 查看虚拟文件系统下对应的加速器设备：

```
[root@localhost rpm]# ls -al /sys/class/uacce/
total 0
lrwxrwxrwx. 1 root root 0 Nov 14 03:45 hisi_hpre-2 -> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/
```



```
uacce/hisi_hpre-2
lrwxrwxrwx. 1 root root 0 Nov 14 03:45 hisi_hpre-3 -> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/
uacce/hisi_hpre-3
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_rde-4 -> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-4
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_rde-5 -> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-5
lrwxrwxrwx. 1 root root 0 Nov 14 08:39 hisi_sec-0 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/
uacce/hisi_sec-0
lrwxrwxrwx. 1 root root 0 Nov 14 08:39 hisi_sec-1 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/
uacce/hisi_sec-1
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_zip-6 -> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/
uacce/hisi_zip-6
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_zip-7 -> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/
uacce/hisi_zip-7
```

步骤4 通过OpenSSL命令验证加速器是否生效，以验证RSA性能为例，命令如下：

```
[root@localhost rpm]# cd /usr/local/bin/
[root@localhost bin]# ./openssl speed rsa2048
      sign verify sign/s verify/s
rsa 2048 bits 0.001381s 0.000035s  724.1  28601.0
[root@localhost bin]# ./openssl speed -engine kae rsa2048
engine "kae" set.
      sign verify sign/s verify/s
rsa 2048 bits 0.000175s 0.000021s  5730.1  46591.8
```

通过RSA性能命令可以看到指定kae引擎之后，RSA的性能得到明显提升。另外，除上述方法，在跑RSA性能命令过程中，我们可以在新的终端上同时查看hpre设备的硬件队列资源情况如下：

```
[root@localhost ~]# cat /sys/class/uacce/hisi_hpre-*/attrs/available_instances
256
255 #从256变为255，说明RSA算消耗了HRPE加速器一个硬件单元队列，说明加速器引擎已生效
[root@localhost ~]#
```

步骤5 （可选）查看zlib库加速器是否生效，用户如果没有安装hisi_zip软件可以忽略该步骤。通过ldd命令查看zlib加速库是否链接到libwd库，如果有返回如下信息，说明zlib加速库安装成功。同样的，用户的进程也可以通过ldd 命令查看是否使用libwd库。

```
[root@localhost /]# ldd /usr/local/zlib/lib/libz.so.1.2.11
linux-vdso.so.1 => (0x0000ffff80280000)
libc.so.6 => /lib64/libc.so.6 (0x0000ffff80080000)
libwd.so.1 => /lib64/libwd.so.1 (0x0000ffff80040000)
/lib/ld-linux-aarch64.so.1 (0x0000ffff80290000)
[root@localhost /]#
```

----结束

3.2 安装后测试

3.2.1 OpenSSL 加速器引擎性能测试

用户可以通过以下命令测试部分加速器功能。

📖 说明

以下测试案例数据来源于原生CentOS7.6。

- 使用OpenSSL的软件算法测试RSA性能。
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed rsa2048
...
 sign verify sign/s verify/s
rsa 2048 bits 0.001384s 0.000035s 724.1 28365.8.
- 使用KAE引擎的测试RSA性能。

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae rsa2048
....
      sign  verify  sign/s  verify/s
rsa 2048 bits 0.000355s 0.000022s  2819.0  45478.4
```

说明

使用KAE引擎加速后签名性能从724.1 sign/s提升到2819sign/s。

- 使用OpenSSL的软件算法测试异步RSA性能。
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -async_jobs 36 rsa2048
....
 sign verify sign/s verify/s
rsa 2048 bits 0.001318s 0.000032s 735.7 28555
- 使用KAE引擎的测试异步RSA性能。
linux-rmw4:/usr/local/bin # ./openssl speed -engine kae -elapsed -async_jobs 36 rsa2048
....
 sign verify sign/s verify/s
rsa 2048 bits 0.000018s 0.000009s 54384.1 105317.0

说明

使用KAE引擎加速后异步RSA签名性能从735.7 sign/s提升到 54384.1sign/s。

- 使用OpenSSL的软件算法测试SM4 CBC模式性能。
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp sm4-cbc
You have chosen to measure elapsed time instead of user CPU time.
....
Doing sm4-cbc for 3s on 10240 size blocks: 2196 sm4-cbc's in 3.00s
type 51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304 bytes8388608 bytes
sm4-cbc 82312.53k 85196.80k 85284.18k 85000.85k 85284.18k 85261.26k
- 使用KAE引擎的测试SM4 CBC模式性能。
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae -evp sm4-cbc
engine "kae" set.
You have chosen to measure elapsed time instead of user CPU time.
...
Doing sm4-cbc for 3s on 1048576 size blocks: 11409 sm4-cbc's in 3.00s
...
type 51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304 bytes8388608 bytes
sm4-cbc 383317.33k 389427.20k 395313.15k 392954.73k 394264.58k 394264.58k

说明

使用KAE加速后SM4 CBC模式在输入数据块大小为8MB时，从82312.53k/s提升到383317.33k/s。

- 使用OpenSSL的软件算法测试SM3模式性能。
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp sm3
You have chosen to measure elapsed time instead of user CPU time.
Doing sm3 for 3s on 102400 size blocks: 1536 sm3's in 3.00s
....
type 51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304 bytes8388608 bytes
sm3 50568.53k 52428.80k 52428.80k 52428.80k 52428.80k 52428.80k
- 使用KAE引擎测试SM3模式性能。
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae -evp sm3
engine "kae" set.
You have chosen to measure elapsed time instead of user CPU time.
Doing sm3 for 3s on 102400 size blocks: 19540 sm3's in 3.00s
....
type 51200 bytes 102400 bytes 1048576 bytes 2097152 bytes 4194304 bytes 8388608 bytes
sm3 648243.20k 666965.33k 677030.57k 678778.20k 676681.05k 668292.44k

说明

使用KAE加速后SM3算法在输入数据块大小为8MB时，从52428.80 k/s提升到668292.44k/s。

- 使用OpenSSL软件算法测试AES算法CBC模式异步性能。

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp aes-128-cbc -async_jobs 4
You have chosen to measure elapsed time instead of user CPU time.
Doing aes-128-cbc for 3s on 51200 size blocks: 65773 aes-128-cbc's in 3.00s
Doing aes-128-cbc for 3s on 102400 size blocks: 32910 aes-128-cbc's in 3.00s
....
type          51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304 bytes8388608 bytes
aes-128-cbc   1122525.87k 1123328.00k 1120578.22k 1121277.27k 1119879.17k 1115684.86k
```

- 使用的KEA引擎测试AES算法CBC模式异步性能。

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp aes-128-cbc -async_jobs 4 -engine kae
engine "kae" set.
You have chosen to measure elapsed time instead of user CPU time.
Doing aes-128-cbc for 3s on 51200 size blocks: 219553 aes-128-cbc's in 3.00s
Doing aes-128-cbc for 3s on 102400 size blocks: 117093 aes-128-cbc's in 3.00s
....
type          51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304 bytes8388608 bytes
aes-128-cbc   3747037.87k 3996774.40k 1189085.18k 1196774.74k 1196979.11k 1199570.94k
```

📖 说明

- AES仅支持数据长度为256KB及以下场景的异步使用。
- 使用KAE加速后AES算法在输入数据块为100K大小时，从1123328.00k/s提升到3996774.40k/s。

3.2.2 zlib 压缩库测试

用户可以在zlib-1.2.11下执行make check命令测试zlib加速库功能。

```
cd KAEzip/open_source/zlib-1.2.11
make check
```

在test目录下编译性能测试工具测试加速库性能。测试性能时，要通过导出环境变量LD_LIBRARY_PATH来启用zlib加速库。

```
export LD_LIBRARY_PATH=/usr/local/kaezip/lib:$ LD_LIBRARY_PATH
cd KAEzip/test
make
[root@localhost test]# ./kaezip_perf
usage:
-m: multi process
-l: stream length(KB)
-n: loop times
-d: compress or decompress
example: ./kaezip_perf -m 2 -l 1024 -n 1000
         ./kaezip_perf -d -m 2 -l 1024 -n 1000

default input parameter used
kaezip perf input parameter: multi process 2, stream length: 1024(KB), loop times: 1000
kaezip compress perf result:
time used: 509004 us, speed = 4.024 GB/s

[root@localhost test]# ./kaezip_perf -d
kaezip perf parameter: multi process 2, stream length: 1024(KB), loop times: 1000
kaezip decompress perf result:
time used: 810318 us, speed = 2.527 GB/s
```

3.3 升级加速引擎软件

如果使用源码方式安装加速器引擎不涉及升级操作，可以跳过本章节。

3.3.1 通过 rpm 包升级加速引擎软件

使用场景

当需要更新加速引擎软件版本时可以使用rpm -Uvh方式进行升级。

操作步骤

- 步骤1** 下载最新版本的加速引擎软件包。
- 步骤2** 使用SSH远程登录工具，以root帐号进入Linux操作系统命令行界面。
- 步骤3** 在存放软件包的路径下使用rpm -Uvh 命令升级加速器驱动包及引擎库包。

命令脚本和信息回显如下所示。

```
root@localhost output]# rpm -Uvh *.rpm
Preparing... ##### [100%]
checking installed modules
uacce modules start to install
Updating / installing...
 1:uacce-1.2.1-1.centos7.6 ##### [ 7%]
uacce modules installed
 2:libwd-1.2.1-1.centos7.6 ##### [ 14%]
 3:libkae-1.2.1-1.centos7.6 ##### [ 21%]
checking installed modules
hisi_hpre modules start to install
 4:hisi_hpre-1.2.1-1.centos7.6 ##### [ 29%]
hisi_hpre modules installed
checking installed modules
hisi_rde modules start to install
 5:hisi_rde-1.2.1-1.centos7.6 ##### [ 36%]
hisi_rde modules installed
checking installed modules
hisi_sec2 modules start to install
 6:hisi_sec2-1.2.1-1.centos7.6 ##### [ 43%]
hisi_sec2 modules installed
checking installed modules
hisi_zip modules start to install
 7:hisi_zip-1.2.1-1.centos7.6 ##### [ 50%]
hisi_zip modules installed
Cleaning up / removing...
 8:libkae-1.2.0-1.centos7.6 ##### [ 57%]
 9:hisi_zip-1.2.0-1.centos7.6 ##### [ 64%]
hisi_zip modules uninstalled
10:hisi_sec2-1.2.0-1.centos7.6 ##### [ 71%]
hisi_sec2 modules uninstalled
11:hisi_rde-1.2.0-1.centos7.6 ##### [ 79%]
hisi_rde modules uninstalled
12:hisi_hpre-1.2.0-1.centos7.6 ##### [ 86%]
hisi_hpre modules uninstalled
13:uacce-1.2.0-1.centos7.6 ##### [ 93%]
uacce modules uninstalled
14:libwd-1.2.0-1.centos7.6 ##### [100%]
[root@localhost output]#
```

- 步骤4** 重启系统或通过命令行手动卸载旧版本驱动，然后加载新版本驱动，并查看是否加载成功。

1. 卸载旧驱动

```
[root@localhost output]# lsmod | grep uacce
uacce          262144  5 hisi_zip,hisi_rde,hisi_hpre,hisi_sec2,hisi_qm
[root@localhost output]#
[root@localhost output]# rmmod hisi_hpre
[root@localhost output]# rmmod hisi_sec2
[root@localhost output]# rmmod hisi_rde
```

```
[root@localhost output]# rmmod hisi_zip
[root@localhost output]# rmmod hisi_qm
[root@localhost output]# rmmod uacce
[root@localhost output]# lsmod | grep uacce
[root@localhost output]#
```

2. 加载新驱动

```
[root@localhost rpm]# modprobe uacce
[root@localhost rpm]# modprobe hisi_sec2 #加载hisi_sec2驱动时将根据/etc/modprobe.d/
hisi_sec2.conf 下的配置文件加载到内核
[root@localhost rpm]# modprobe hisi_hpre #加载hisi_hpre驱动时将根据/etc/modprobe.d/
hisi_hpre.conf 下的配置文件加载到内核
[root@localhost rpm]# modprobe hisi_rde #加载hisi_rde驱动时将根据/etc/modprobe.d/hisi_rde.conf 下
的配置文件加载到内核
[root@localhost rpm]# modprobe hisi_zip #加载hisi_zip驱动时将根据/etc/modprobe.d/hisi_zip.conf 下
的配置文件加载到内核
[root@localhost rpm]# lsmod | grep uacce
uacce          36864  3 hisi_sec2,hisi_qm,hisi_hpre,hisi_rde,hisi_zip
```

步骤5 使用rpm -qa | grep查看升级后的软件版本。

```
[root@localhost root]# rpm -qa uacce hisi_sec2 hisi_hpre hisi_zip hisi_rde
hisi_rde-1.2.1-1.centos7.6.aarch64 //显示升级后的版本号说明升级成功
hisi_sec2-1.2.1-1.centos7.6.aarch64
uacce-1.2.1-1.centos7.6.aarch64
hisi_hpre-1.2.1-1.centos7.6.aarch64
hisi_zip-1.2.1-1.centos7.6.aarch64
```

----结束

3.3.2 通过 deb 包升级加速引擎软件

使用场景

当需要更新加速引擎软件版本时可以使用deb包方式进行升级。

操作步骤

步骤1 下载最新版本的加速引擎软件包。

步骤2 使用SSH远程登录工具，以root帐号进入Linux操作系统命令行界面。

步骤3 运行dpkg -r 命令先卸载加速引擎软件包。

说明

由于存在依赖关系，卸载uacce驱动软件包前须先卸载hisi_hpre， hisi_sec2， hisi_rde,hisi_zip驱动软件包， 卸载libwd前须先卸载libkae引擎软件包。

命令脚本如下所示。

```
[root@localhost output]# dpkg -r hisi-sec2
```

步骤4 在存放软件包的路径下使用dpkg -i *.deb命令重新安装加速器驱动包及引擎库包。

步骤5 重启系统或通过命令行手动卸载旧版本驱动，然后加载新版本驱动，并查看是否加载成功。

1. 卸载旧驱动：

```
[root@localhost output]# lsmod | grep uacce
uacce          262144  5 hisi_zip,hisi_rde,hisi_hpre,hisi_sec2,hisi_qm
[root@localhost output]#
[root@localhost output]# rmmod hisi_hpre
[root@localhost output]# rmmod hisi_sec2
[root@localhost output]# rmmod hisi_rde
```

```
[root@localhost output]# rmmod hisi_zip
[root@localhost output]# rmmod hisi_qm
[root@localhost output]# rmmod uacce
[root@localhost output]# lsmod | grep uacce
[root@localhost output]#
```

2. 加载新驱动:

```
[root@localhost rpm]# modprobe uacce
[root@localhost rpm]# modprobe hisi_sec2 #加载hisi_sec2驱动时将根据/etc/modprobe.d/
hisi_sec2.conf 下的配置文件加载到内核
[root@localhost rpm]# modprobe hisi_hpre #加载hisi_hpre驱动时将根据/etc/modprobe.d/
hisi_hpre.conf 下的配置文件加载到内核
[root@localhost rpm]# modprobe hisi_rde #加载hisi_rde驱动时将根据/etc/modprobe.d/hisi_rde.conf 下
的配置文件的配置文件加载到内核
[root@localhost rpm]# modprobe hisi_zip #加载hisi_zip驱动时将根据/etc/modprobe.d/hisi_zip.conf 下
的配置文件的配置文件加载到内核
[root@localhost rpm]# lsmod | grep uacce
uacce          36864  3 hisi_sec2,hisi_qm,hisi_hpre,hisi_rde,hisi_zip
```

步骤6 使用dpkg -s查看升级后的软件版本。

```
[root@localhost root]# dpkg -s uacce hisi-sec2 hisi-hpre hisi-zip hisi-rde
Package: uacce
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Unified/User-space-access-intended Accelerator Framework.
URL: https://support.huawei.com

Package: hisi-sec2
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Huawei Hisilicon SEC Accelerator Driver.
URL: https://support.huawei.com

Package: hisi-hpre
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Huawei Hisilicon HPRE Accelerator Driver.
URL: https://support.huawei.com

Package: hisi-zip
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
Maintainer: HiSilicon Tech. Co., Ltd.
Architecture: arm64
Version: 1.3.1
Description: This package contains the Huawei Hisilicon ZIP Accelerator Driver.
URL: https://support.huawei.com

Package: hisi-rde
Status: install ok installed
Priority: optional
Section: utils
Installed-Size: 512
```

```
Maintainer: HiSilicon Tech. Co., Ltd.  
Architecture: arm64  
Version: 1.3.1  
Description: This package contains the Huawei Hisilicon RDE Accelerator Driver.  
URL: https://support.huawei.com
```

----结束

3.4 卸载加速引擎软件

3.4.1 卸载通过 rpm 包安装的加速引擎软件

使用场景

用户不再使用通过rpm包安装的加速引擎软件，或想进行新版本加速引擎软件的安装。

操作步骤

步骤1 使用SSH远程登录工具，以root帐号进入Linux操作系统命令行界面。

步骤2 卸载加速引擎软件包并检查卸载情况。

1. 运行rpm -e命令卸载加速引擎软件包。

📖 说明

由于存在依赖关系，卸载uacce驱动软件包前须先卸载hisi_hpre，hisi_sec2，hisi_rde,hisi_zip驱动软件包，卸载libwd前须先卸载libkae引擎软件包。

命令脚本如下所示。

```
[root@localhost home]# rpm -e hisi_sec2
```

2. 使用rpm -qa |grep 软件包名命令可以查询是否卸载成功。

命令脚本如下所示。

```
[root@localhost home]# rpm -qa |grep hisi_sec2
```

如下示例演示了卸载所有加速引擎软件包。

```
[root@localhost output]# rpm -e hisi_hpre  
hisi_hpre modules uninstalled  
[root@localhost output]# rpm -e hisi_sec2  
hisi_sec2 modules uninstalled  
[root@localhost output]# rpm -e hisi_zip  
hisi_zip modules uninstalled  
[root@localhost output]# rpm -e hisi_rde  
hisi_rde modules uninstalled  
[root@localhost output]#  
[root@localhost output]# rpm -e uacce  
uacce modules uninstalling  
uacce modules uninstalled  
[root@localhost output]# rpm -e libwd libkae libkaezip  
[root@localhost output]#  
[root@localhost output]# rpm -qa uacce hisi_hpre hisi_sec2 hisi_rde hisi_zip  
[root@localhost output]# rpm -qa libwd libkae libkaezip  
[root@localhost output]#
```

步骤3 重启系统或通过命令行手动将已加载到内核的驱动卸载掉，并查看是否卸载成功。

```
[root@localhost rpm]# lsmod | grep uacce  
uacce          36864  3 hisi_sec2,hisi_qm,hisi_hpre,hisi_rde,hisi_zip  
[root@localhost output]# lsmod | grep uacce  
uacce          262144  5 hisi_zip,hisi_rde,hisi_hpre,hisi_sec2,hisi_qm
```

```
[root@localhost output]#  
[root@localhost output]# rmmod hisi_hpre  
[root@localhost output]# rmmod hisi_sec2  
[root@localhost output]# rmmod hisi_rde  
[root@localhost output]# rmmod hisi_zip  
[root@localhost output]# rmmod hisi_qm  
[root@localhost output]# rmmod uacce  
[root@localhost output]# lsmod | grep uacce  
[root@localhost output]#
```

---结束

3.4.2 卸载通过 deb 包安装的加速引擎软件

使用场景

用户不再使用通过deb包安装的加速引擎软件，或想进行新版本加速引擎软件的安装。

操作步骤

步骤1 使用SSH远程登录工具，以root帐号进入Linux操作系统命令行界面。

步骤2 卸载加速引擎软件包并检查卸载情况。

1. 运行dpkg -r 命令卸载加速引擎软件包。

📖 说明

由于存在依赖关系，卸载uacce驱动软件包前须先卸载hisi_hpre, hisi_sec2, hisi_rde,hisi_zip驱动软件包，卸载libwd前须先卸载libkae引擎软件包。

命令脚本如下所示。

```
[root@localhost output]# dpkg -r hisi-sec2
```

2. 运行命令查询软件包是否卸载成功。

命令脚本如下所示。

```
[root@localhost output]# dpkg -r libwd libkae libkaezip
```

如下示例演示了卸载所有加速引擎软件包。

```
[root@localhost output]# dpkg -r hisi-hpre  
(Reading database ... 79192 files and directories currently installed.)  
Removing hisi-hpre (1.3.2) ...  
hisi_hpre modules uninstalling  
hisi_hpre modules uninstalled  
[root@localhost output]# dpkg -r hisi-sec2  
(Reading database ... 79190 files and directories currently installed.)  
Removing hisi-sec2 (1.3.2) ...  
hisi_sec2 modules uninstalling  
hisi_sec2 modules uninstalled  
[root@localhost output]# dpkg -r hisi-zip  
(Reading database ... 79188 files and directories currently installed.)  
Removing hisi-zip (1.3.2) ...  
hisi_zip modules uninstalling  
hisi_zip modules uninstalled  
[root@localhost output]# dpkg -r hisi-rde  
(Reading database ... 79186 files and directories currently installed.)  
Removing hisi-rde (1.3.2) ...  
hisi_rde modules uninstalling  
hisi_rde modules uninstalled  
[root@localhost output]# dpkg -r uacce  
(Reading database ... 79184 files and directories currently installed.)  
Removing uacce (1.3.2) ...  
uacce modules uninstalling
```



```
uacce modules uninstalled
[root@localhost output]# dpkg -r libwd libkae libkaezip
(Reading database ... 79181 files and directories currently installed.)
Removing libkae (1.3.2) ...
Removing libkaezip (1.3.2) ...
Removing libwd (1.3.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
[root@localhost output]# dpkg -L uacce hisi-hpre hisi-sec2 hisi-rde hisi-zip
Package 'uacce' does not contain any files (!)

Package 'hisi-hpre' does not contain any files (!)

Package 'hisi-sec2' does not contain any files (!)

Package 'hisi-rde' does not contain any files (!)

Package 'hisi-zip' does not contain any files (!)
[root@localhost output]# dpkg -Llibwd libkae libkaezip
/usr
/usr/local
/usr/local/include

/usr
/usr/local
/usr/local/lib
/usr/local/lib/engines-1.1

Package 'libkaezip' does not contain any files (!)
[root@localhost output]#
```

步骤3 重启系统或通过命令行手动将已加载到内核的驱动卸载掉，并查看是否卸载成功。

```
[root@localhost rpm]# lsmod | grep uacce
uacce          36864  3 hisi_sec2,hisi_qm,hisi_hpre,hisi_rde,hisi_zip
[root@localhost output]# lsmod | grep uacce
uacce          262144  5 hisi_zip,hisi_rde,hisi_hpre,hisi_sec2,hisi_qm
[root@localhost output]#
[root@localhost output]# rmmod hisi_hpre
[root@localhost output]# rmmod hisi_sec2
[root@localhost output]# rmmod hisi_rde
[root@localhost output]# rmmod hisi_zip
[root@localhost output]# rmmod hisi_qm
[root@localhost output]# rmmod uacce
[root@localhost output]# lsmod | grep uacce
[root@localhost output]#
```

----结束

3.4.3 卸载通过源码方式安装的加速引擎软件

使用场景

用户不再使用通过源码安装的加速引擎软件，或想进行新版本加速引擎软件的安装。

操作步骤

步骤1 使用SSH远程登录工具，以root帐号进入Linux操作系统命令行界面。

步骤2 源码方式安装的加速器驱动包及加速器引擎库包通过make uninstall 命令进行卸载。

- 卸载驱动：

```
cd kae_driver
make uninstall
```
- 卸载warpdriver驱动开发库：

```
cd warpdriver
make uninstall
```

- 卸载引擎：
cd KAE
make uninstall

- 卸载zlib加速库：
cd KAEzip
sh setup.sh uninstall

步骤3 重启系统或通过命令行手动将已加载到内核的驱动卸载掉，并查看是否卸载成功。

```
[root@localhost rpm]# lsmod | grep uacce
uacce          36864  3 hisi_sec2,hisi_qm,hisi_hpre,hisi_rde,hisi_zip
[root@localhost output]# lsmod | grep uacce
uacce          262144  5 hisi_zip,hisi_rde,hisi_hpre,hisi_sec2,hisi_qm
[root@localhost output]#
[root@localhost output]# rmmod hisi_hpre
[root@localhost output]# rmmod hisi_sec2
[root@localhost output]# rmmod hisi_rde
[root@localhost output]# rmmod hisi_zip
[root@localhost output]# rmmod hisi_qm
[root@localhost output]# rmmod uacce
[root@localhost output]# lsmod | grep uacce
[root@localhost output]#
```

----结束

4 使用案例

4.1 加速引擎的应用

4.2 zlib加速库的使用

4.1 加速引擎的应用

4.1.1 加速引擎使用说明

- 如果用户未购买引擎许可证，建议用户不要通过kae引擎调用相应算法，否则可能会影响OpenSSL加密算法的性能。
- SM4-XTS模式仅支持内核态使用，具体使用方法请参考[4.1.5 基于dm-crypt的透明分区/磁盘加密](#)。
- SM4同步性能在小包场景下（包长小于2K）性能比异步性能好；如果使用场景多为小包场景，推荐使用同步模型。
- AES已在aarch64平台上实现软件指令集加速，硬件加速在中包或大包场景下（包长16K~256K）异步性能相比OpenSSL才具明显优势，推荐在该场景下中使用硬加速。
- 缩解压算法协议格式仅支持zlib/gzip，刷新模式仅支持Z_NO_FLUSH/Z_SYNC_FLUSH/Z_FULL_FLUSH/Z_FINISH。
- SM4、AES异步模式支持数据长度为256KB及以下，数据长度大于256KB将自动切换同步模式。
- MD5算法无法防止碰撞攻击，不适用与安全性认证，如SSL公开密钥认证或是数字签名等用途。
- SM3/SM4算法默认开启，用户可以通过openssl.cnf文件开启或关闭这两种算法。

4.1.2 KAE 引擎使用示例代码

```
#include <stdio.h>
#include <stdlib.h>

/* OpenSSL headers */
#include <openssl/bio.h>
#include <openssl/ssl.h>
#include <openssl/err.h>
#include <openssl/engine.h>
```

```
int main(int argc, char **argv)
{
    /* Initializing OpenSSL */
    SSL_load_error_strings();
    ERR_load_BIO_strings();
    OpenSSL_add_all_algorithms();

    /*You can use ENGINE_by_id Function to get the handle of the Huawei Accelerator Engine*/
    ENGINE *e = ENGINE_by_id("kae");
    /*使能加速引擎异步功能，可选配置，设置为“0”表示不使能，设置为“1”表示使能，默认使能异步功能*/
    ENGINE_ctrl_cmd_string(e, "KAE_CMD_ENABLE_ASYNC", "1", 0)
    ENGINE_init(e);
    /*指定引擎用于RSA加解密，如果初始时使用ENGINE_set_default_RSA(ENGINE *e);则无需传入e*/
    RSA *rsa = RSA_new_method(e);
    /*The user code*/
    .....

    ENGINE_free(e);
}
```

用户还可以在初始化阶段指定crypto相应算法使用KAE引擎，其他算法不需要使用KAE引擎，这样对已有的代码修改量将更小，只需要在初始化的某个阶段设置一下即可。

```
int ENGINE_set_default_RSA(ENGINE *e);
int ENGINE_set_default_DH(ENGINE *e);
int ENGINE_set_default_ciphers(ENGINE *e);
int ENGINE_set_default_digests(ENGINE *e);
int ENGINE_set_default(ENGINE *e, unsigned int flags);
```

更多使用API方法请访问OpenSSL官网，链接如下：

https://www.openssl.org/docs/man1.1.0/man3/ENGINE_set_default_ciphers.html

4.1.3 通过 OpenSSL 配置文件 openssl.cnf 使用 KAE 引擎

通过配置文件方式使用KAE加速器，可以使用户的APP在非常小的修改量情况下使用加速器功能，仅需调用一次初始化API，如下所示：

```
OPENSSL_init_crypto(OPENSSL_INIT_LOAD_CONFIG, NULL);
```

新建openssl.cnf 需要添加如下配置信息

```
openssl_conf=openssl_def
[openssl_def]
engines=engine_section
[engine_section]
kae=kae_section
[kae_section]
engine_id=kae
dynamic_path=/usr/local/lib/engines-1.1/kae.so
KAE_CMD_ENABLE_ASYNC=1 #可选配置，0表示不使能异步功能，1表示使能异步功能，默认使能
KAE_CMD_ENABLE_SM3=1 #可选配置，0表示不使能SM3加速功能，1表示使能SM3加速功能，默认使能
KAE_CMD_ENABLE_SM4=1 #可选配置，0表示不使能SM4加速功能，1表示使能SM4加速功能，默认使能
default_algorithms=ALL #表示所有算法优先查找引擎，若引擎不支持，则切换OpenSSL进行计算
init=1
```

导出OPENSSL_CONF环境变量：

```
export OPENSSL_CONF=/home/app/openssl.cnf #该路径为openssl.cnf存放路径
```

使用OpenSSL配置文件示例如下：

```
#include <stdio.h>
#include <stdlib.h>
```

```
/* OpenSSL headers */
#include <openssl/bio.h>
#include <openssl/err.h>
#include <openssl/engine.h>
int main(int argc, char **argv)
{
    /* Initializing OpenSSL */

    ERR_load_BIO_strings();
    /* Load openssl configure */
    OPENSSL_init_crypto(OPENSSL_INIT_LOAD_CONFIG, NULL);
    /*The user code*/
    .....
```

4.1.4 KAE 引擎加速 Nginx 应用

Tengine是阿里巴巴基于开源nginx推出的具有异步功能的web服务引擎，以Tengine为例介绍KAE引擎如何使能Nginx加速。具体操作过程请参见《Tengine 2.2.2 移植指南》中的[源码编译安装](#)。

验证

如果有需要，请参考《HTTPS RSA加速 特性指南》中的[编译和配置](#)。

步骤1

----结束

4.1.5 基于 dm-crypt 的透明分区/磁盘加密

dm-crypt向上呈现为一个device mapper机制的target device，经过映射挂载后就可以作为透明加密分区/磁盘使用。dm-crypt算法注册在Crypto模块中，hisi_sec2驱动安装后，SM4-XTS算法会注册到Crypto模块中，使用LUKS（Linux Unified Key Setup）进行配置即可实现硬件加解密。一个加密盘操作要占用24个队列，当前加速器限制开放256*2的队列数，如果需要操作更多数量的加密盘，需要先开启所有1024*2的加速器队列，开启方法，修改/etc/modprobe.d/hisi_sec2.conf配置文件中pf_q_num参数，重启生效。

环境要求

- 已安装hisi_sec2驱动，安装方法请参考[3.1 安装加速引擎软件](#)。
- 为提升SM4-XTS算法性能，请将LUKS工具Cryptsetup升级至2.2.0版本。

📖 说明

操作系统自带cryptsetup软件可能无法正在使用SM4-XTS算法加密磁盘，需要进行升级，请下载cryptsetup-2.2.0源码到环境中，以EulerOS2.8为例，升级cryptsetup命令如下：

```
[root@localhost cryptsetup-2.2.0-rc1]# yum install libuuid-devel
[root@localhost cryptsetup-2.2.0-rc1]# yum install device-mapper-devel
[root@localhost cryptsetup-2.2.0-rc1]# yum install popt-devel
[root@localhost cryptsetup-2.2.0-rc1]# yum install json-c-devel
[root@localhost cryptsetup-2.2.0-rc1]# yum install libblkid-devel2.2.0
[root@localhost cryptsetup-2.2.0-rc1]# ./configure
[root@localhost cryptsetup-2.2.0-rc1]# make && make install
```

其中，libuuid-devel、device-mapper-devel、popt-devel、popt-devel、libblkid-devel2.2.0为cryptsetup依赖软件包。

加密分区/磁盘

- 步骤1** 执行`dd if=/dev/random of=/home/EncryptKeyFile bs=4k count=1`命令，生成keyfile文件。

```
linux-lhzu:~ # dd if=/dev/random of=/home/EncryptKeyFile bs=4k count=1
0+1 records in
0+1 records out
115 bytes copied, 0.00010976 s, 1.0 MB/s
```

- 步骤2** 执行`cryptsetup --batch-mode --cipher sm4-xts-plain64 --key-size 256 --hash sha256 --sector-size=4096 --type=luks2 --key-file /home/EncryptKeyFile luksFormat /dev/sdb`命令，加密分区/磁盘。

```
linux-lhzu:~ # cryptsetup --batch-mode --cipher sm4-xts-plain64 --key-size 256 --hash sha256 --sector-size=4096 --type=luks2 --key-file /home/EncryptKeyFile luksFormat /dev/sdb
```

- 步骤3** 执行`cryptsetup --key-file /home/EncryptKeyFile luksOpen /dev/sdb sx_disk`命令，映射分区/磁盘。

```
linux-lhzu:~ # cryptsetup --key-file /home/EncryptKeyFile luksOpen /dev/sdb sx_disk
```

- 步骤4** 执行`lsblk`指令，查看分区/磁盘是否加密。

```
linux-lhzu:~ # lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0  0  5.5G  1 loop /os_lhl
sda         8:0  0  2.2T  0 disk
├─sda1      8:1  0   1G  0 part /boot/efi
├─sda2      8:2  0  2.2T  0 part
├─vg_os-swap 254:0  0   20G  0 lvm  [SWAP]
├─vg_os-root 254:1  0  2.2T  0 lvm  /
sdb         8:16  0 278.5G  0 disk
├─sx_disk   254:2  0 278.5G  0 crypt #crypt表明分区/磁盘已加密
```

- 步骤5** 执行`mkfs.xfs /dev/mapper/sx_disk`命令，格式化分区/磁盘。

```
linux-lhzu:~ # mkfs.xfs /dev/mapper/sx_disk
meta-data=/dev/mapper/sx_disk isize=512  agcount=16, agsize=4562368 blks
         =                       sectsz=512  attr=2, projid32bit=1
         =                       crc=1      finobt=1, sparse=0, rmapbt=0, reflink=0
data     =                       bsize=4096  blocks=72997376, imaxpct=25
         =                       sunit=64   swidth=64 blks
naming   =version 2              bsize=4096  ascii-ci=0 ftype=1
log      =internal log          bsize=4096  blocks=35648, version=2
         =                       sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096  blocks=0, rtextents=0
```

- 步骤6** 执行`mkdir /home/sec_test`命令，创建挂载点目录。

```
linux-lhzu:~ # mkdir /home/sec_test
```

- 步骤7** 执行`mount /dev/mapper/sx_disk /home/sec_test`命令，挂载分区/磁盘到目录。

```
linux-lhzu:~ # mount /dev/mapper/sx_disk /home/sec_test/
linux-lhzu:~ # df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        63G   0   63G   0% /dev
tmpfs           63G   0   63G   0% /dev/shm
tmpfs           63G  28M   63G   1% /run
tmpfs           63G   0   63G   0% /sys/fs/cgroup
/dev/mapper/vg_os-root 2.2T  18G  2.1T   1% /
/dev/sda1       1022M 172K 1022M   1% /boot/efi
tmpfs           13G  20K   13G   1% /run/user/472
tmpfs           13G   0   13G   0% /run/user/0
/dev/loop0      5.5G  5.5G   0 100% /os_lhl
/dev/mapper/sx_disk 279G 317M 279G   1% /home/sec_test
```

- 步骤8** 执行`cd /home/sec_test;ll`命令，确认目录可正常访问。

```
linux-lhzu:~ # cd /home/sec_test;ll
```

步骤9 执行lsblk命令，查看分区/磁盘是否已加密，并且和目录是否正确对应。

```
linux-lhzu:/home/sec_test # lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0  0  5.5G  1 loop /os_lhl
sda         8:0  0  2.2T  0 disk
├─sda1      8:1  0   1G  0 part /boot/efi
└─sda2      8:2  0  2.2T  0 part
   └─vg_os-swap 254:0  0   20G  0 lvm  [SWAP]
      └─vg_os-root 254:1  0  2.2T  0 lvm  /
sdb         8:16  0 278.5G  0 disk
└─sx_disk   254:2  0 278.5G  0 crypt /home/sec_test
```

步骤10 执行cryptsetup status /dev/mapper/sx_disk命令，查看分区/磁盘加密详细信息。

```
linux-lhzu:/home # cryptsetup status /dev/mapper/sx_disk
/dev/mapper/sx_disk is active and is in use.
type: LUKS1
cipher: sm4-xts-plain64
keysize: 256 bits
key location: dm-crypt
device: /dev/sdb
sector size: 512
offset: 4096 sectors
size: 583979008 sectors
mode: read/write
```

步骤11 执行**步骤2**到**步骤10**，对多个分区/磁盘进行加密。

----结束

删除加密的分区/磁盘

步骤1 执行umount -l /home/sec_test命令，卸载分区/磁盘的挂载目录。

📖 说明

执行指令前，用户必须先退出挂载目录。

当有多个分区/磁盘挂载时，需要多次执行该命令进行目录卸载。

```
linux-lhzu:/home # umount -l /home/sec_test
```

步骤2 执行lsblk命令，确认已卸载分区/磁盘的挂载目录。

```
linux-lhzu:/home # lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0  0  5.5G  1 loop /os_lhl
sda         8:0  0  2.2T  0 disk
├─sda1      8:1  0   1G  0 part /boot/efi
└─sda2      8:2  0  2.2T  0 part
   └─vg_os-swap 254:0  0   20G  0 lvm  [SWAP]
      └─vg_os-root 254:1  0  2.2T  0 lvm  /
sdb         8:16  0 278.5G  0 disk
└─sx_disk   254:2  0 278.5G  0 crypt
```

步骤3 执行cryptsetup luksClose sx_disk命令，关闭映射。

📖 说明

需要多次执行该命令关闭所有映射。

```
linux-lhzu:/home # cryptsetup luksClose sx_disk
```

步骤4 执行ll /dev/mapper指令，查看映射是否关闭。

```
linux-lhzu:/home # ll /dev/mapper/
total 0
crw---- 1 root root 10, 236 Jul 31 22:27 control
```

```
lrwxrwxrwx 1 root root 7 Jul 31 22:27 vg_os-root -> ../dm-1
lrwxrwxrwx 1 root root 7 Jul 31 22:27 vg_os-swap -> ../dm-0
```

----结束

4.1.6 加速引擎在 KVM 虚拟机中的使用

支持在KVM虚拟机使用加速引擎，加速器设备遵循PCIe规范，在操作系统内呈现为PCIe设备，并支持SR-IOV能力。每个加速器提供了1024个队列，单个PF默认使用256个队列，其余768个队列预留给VF使用。VF队列数量=（1024-PF队列数量）/ VF个数，余数队列会加到最后一个VF上。

推荐一个PF虚拟化出8个VF数目。

环境要求

- HostOS上已经建立虚拟机。
- HostOS上已经安装加速器内核驱动，虚拟机上已经安装加速引擎软件包，安装方法请参考[3.1 安装加速引擎软件](#)。

在 HostOS 上进行虚拟化配置

步骤1 执行ls -al /sys/class/uacce查询HostOS环境中安装的加速器和对应的bdf号。

```
hisi_hpre-0 -> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/uacce/hisi_hpre-0
hisi_hpre-1 -> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/uacce/hisi_hpre-1
hisi_rde-2 -> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-2
hisi_rde-3 -> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-3
hisi_sec-6 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/uacce/hisi_sec-6
hisi_sec-7 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/uacce/hisi_sec-7
hisi_zip-4 -> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/uacce/hisi_zip-4
hisi_zip-5 -> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/uacce/hisi_zip-5
```

步骤2 虚拟化加速器VF(以hisi_sec设备为例，各虚拟出3个VF，对应hisi_sec - 8 ~ hisi_sec - 13)。

```
echo 3 > /sys/devices/pci0000:74/0000:74:01.0/0000:76:00.0/sriov_numvfs
echo 3 > /sys/devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/sriov_numvfs
```

```
hisi_hpre-0 -> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/uacce/hisi_hpre-0
hisi_hpre-1 -> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/uacce/hisi_hpre-1
hisi_rde-2 -> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-2
hisi_rde-3 -> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-3
hisi_sec-10 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.3/uacce/hisi_sec-10
hisi_sec-11 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.1/uacce/hisi_sec-11
hisi_sec-12 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.2/uacce/hisi_sec-12
hisi_sec-13 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.3/uacce/hisi_sec-13
hisi_sec-6 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/uacce/hisi_sec-6
hisi_sec-7 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/uacce/hisi_sec-7
hisi_sec-8 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.1/uacce/hisi_sec-8
hisi_sec-9 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.2/uacce/hisi_sec-9
hisi_zip-4 -> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/uacce/hisi_zip-4
hisi_zip-5 -> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/uacce/hisi_zip-5
```

----结束

在虚拟机上配置加速器

步骤1 执行virsh edit vm1命令，编辑虚拟机vm1的配置文件。

步骤2 在配置文件中添加vcpu配置（以配置4个core为例）。

```
<cputune>
<vcupin vcpu='0' cpuset='4'/>
```



```
<vcpupin vcpu='1' cpuset='5'/>
<vcpupin vcpu='2' cpuset='6'/>
<vcpupin vcpu='3' cpuset='7'/>
<emulatorpin cpuset='4-7'/>
</cputune>
```

经过上述的配置后，虚拟机进程运行会固在指定的主机的物理CPU上。

步骤3 虚拟机配置VF。

```
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address bus='0x76' slot='0x00' function='0x1'/>
  </source>
</hostdev>
```

经过上述的配置后，虚拟机挂载了加速器虚拟出来的一个VF。

📖 说明

- hisi_sec设备SBDF号以0000:7x:xx.x，其对应CPU0上设备；以0000:bx:xx.x为开头，对应CPU1上设备。
- 为保证性能稳定，推荐虚拟机上核选取对应CPU上的core，同时VF也选择对应加速器上虚拟出来的VF。
- HostOS对单个虚拟机上VF挂载个数存在上限，默认为11个。

步骤4 执行virsh start vm1启动虚拟机。

```
[root@vm2 ~]# virsh start vm1
error: Failed to start domain vm1
error: internal error: Unknown PCI header type '127'
```

如果启动虚拟机失败，并提示上述错误，则操作以下命令，对挂载的VF进行解绑操作，然后重新启动虚拟机。

```
echo 0000:76:00.1 > /sys/bus/pci/drivers/hisi_sec/unbind
echo vfio-pci > /sys/devices/pci0000:74/0000:74:01.0/0000:76:00.1/driver_override
echo 0000:76:00.1 > /sys/bus/pci/drivers_probe
```

步骤5 登陆虚拟机查询设备。

```
linux-4myg:~ # ls /sys/class/uacce/
hisi_sec-0
```

说明挂载的VF已经在虚拟机上读取成功。

----结束

4.1.7 加速引擎在 docker 中的使用

支持在docker中使用加速引擎，加速器设备遵循PCIe规范，在操作系统内呈现为PCIe设备，并支持SR-IOV能力。每个加速器提供了1024个队列，单个PF默认使用256个队列，其余768个队列预留VF使用。VF队列数量= (1024-PF队列数量) / VF个数，余数队列会加到最后一个VF上。

推荐一个PF虚拟化出8个VF数目。

环境要求

- HostOS上已经建立docker容器。
- HostOS上已经安装加速器内核驱动，安装方法请参考[3.1 安装加速引擎软件](#)。

在 HostOS 上进行虚拟化配置

步骤1 执行 `ls -al /sys/class/uacce` 查询 HostOS 环境中安装的加速器和对应的 bdf 号。

```
hisi_hpre-0 -> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/uacce/hisi_hpre-0
hisi_hpre-1 -> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/uacce/hisi_hpre-1
hisi_rde-2 -> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-2
hisi_rde-3 -> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-3
hisi_sec-6 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/uacce/hisi_sec-6
hisi_sec-7 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/uacce/hisi_sec-7
hisi_zip-4 -> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/uacce/hisi_zip-4
hisi_zip-5 -> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/uacce/hisi_zip-5
```

步骤2 虚拟化加速器 VF (以 hisi_sec 设备为例, 各虚拟出 3 个 VF, 对应 hisi_sec - 8 ~ hisi_sec - 13)。

```
echo 3 > /sys/devices/pci0000:74/0000:74:01.0/0000:76:00.0/sriov_numvfs
echo 3 > /sys/devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/sriov_numvfs
```

```
hisi_hpre-0 -> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/uacce/hisi_hpre-0
hisi_hpre-1 -> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/uacce/hisi_hpre-1
hisi_rde-2 -> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-2
hisi_rde-3 -> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-3
hisi_sec-10 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.3/uacce/hisi_sec-10
hisi_sec-11 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.1/uacce/hisi_sec-11
hisi_sec-12 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.2/uacce/hisi_sec-12
hisi_sec-13 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.3/uacce/hisi_sec-13
hisi_sec-6 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/uacce/hisi_sec-6
hisi_sec-7 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/uacce/hisi_sec-7
hisi_sec-8 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.1/uacce/hisi_sec-8
hisi_sec-9 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.2/uacce/hisi_sec-9
hisi_zip-4 -> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/uacce/hisi_zip-4
hisi_zip-5 -> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/uacce/hisi_zip-5
```

步骤3 启动 docker 容器, 并给容器分配加速器 VF

```
docker run -it -v /usr:/usr/ --device=/dev/hisi_sec-8:/dev/hisi_sec-2:rwm -m 8192m --cpuset-cpus="4-7"
90b5058926a2 /bin/bash
```

```
[root@localhost lv]# docker run -it -v /usr:/usr/ --device=/dev/hisi_sec-2:/dev/hisi_sec-2:rwm
[root@5171399e2 /]#
[root@5171399e2 /]#
[root@5171399e2 /]#
```

参数说明:

- l : 让 docker 分配一个伪终端并绑定在容器的标准输入上

- t : 让容器的标准输入保持打开

- v : 让宿主机的目录挂载到镜像里, 冒号前为宿主机目录, 必须为绝对路径, 冒号后为镜像内挂载的路径

--device: 指定容器使用宿主机的设备, 冒号前为宿主机上创建的 VF 设备, 冒号后为容器内目录, r, w, m 使容器拥有对设备的读、写、创建设备文件的权限

-m: 限制容器使用最大内存数量

--cpuset-cpus: 指定容器在哪些 cpu 内核上运行

90b5058926a2: 为镜像 id, 也可换成镜像名, 查看命令为 `docker images`

/bin/bash 启动容器的 bash

----结束

4.1.8 MD5 硬件加速调优

详细信息请参见《Ceph对象存储 调优指南（鲲鹏920）》中的[MD5硬件加速调优](#)。

4.2 zlib 加速库的使用

参照[3.1.6 通过源码方式安装加速引擎软件](#)章节编译并安装好hisi_zip内核模块、warpdrive用户态驱动、zlib库。应用层可以通过下面两种方式链接到zlib加速压缩库：

- 方式一：应用层在编译阶段指定运行时加载zlib.so的位置，通过以下编译选项进行链接：

```
-Wl,-rpath=/usr/local/kaezip/lib #其中/usr/local/kaezip/lib为一示例，表示新安装zlib库的路径
```

具体案例如下：

- 《Ceph块存储 调优指南（鲲鹏920）》中的[zlib硬件加速调优](#)
- 《Ceph对象存储 调优指南（鲲鹏920）》中的[zlib硬件加速调优](#)
- 《Ceph文件存储 调优指南（鲲鹏920）》中的[zlib硬件加速调优](#)

- 方式二：导出环境变量：

```
export LD_LIBRARY_PATH=/usr/local/kaezip/lib:$LD_LIBRARY_PATH
```

5 常用操作

5.1 加速器日志查询

5.1 加速器日志查询

加速器引擎涉及日志信息如表5-1所示。

表 5-1 日志信息

目录	文件名	文件内容说明
/var/log/	kae.log	OpenSSL引擎日志默认打印等级为error级别，如需要设置日志级别按照如下操作： 1. export KAE_CONF_ENV=/var/log/ 2. 在/var/log/下创建文件kae.cnf 3. 在kae.cnf 文件中设置如下： [LogSection] debug_level=error #取值内容none/error/info/warning/debug 说明 正常情况下不建议开启info或debug级别日志，否则会导致加速器性能的下降。
/var/log/	kaezip.log	Zlib加速库日志默认不打印，如需要设置日志级别按照如下操作： 1. export KAEZIP_CONF_ENV=/var/log/ 2. 在/var/log/下创建文件kaezip.cnf 3. 在kaezip.cnf 文件中设置如下： [LogSection] debug_level=error #取值内容none/error/info/warning/debug 说明 正常情况下不建议开启info或debug级别日志，否则会导致加速器性能的下降。

目录	文件名	文件内容说明
/var/log/	message/ syslog	<ul style="list-style-type: none">CentOS, SUSE, Euler等OS内核日志路径为/var/log/message。Ubuntu等OS内核日志路径为/var/log/syslog。 说明 或通过dmesg > /var/log/dmesg.log日志收集内核相关日志，包含驱动及内核态日志。

6 故障处理

- 6.1 初始化失败
- 6.2 安装完加速器引擎之后，查找不到加速器设备
- 6.3 升级加速器驱动失败
- 6.4 操作系统安装完OpenSSL新版本后提示相关接口符号未找到

6.1 初始化失败

故障现象

加速器引擎没有完全加载成功。

处理步骤

- 步骤1** 检查加速器驱动是否加载成功，运行lsmod 命令查看uacce.ko、qm.ko、sgl.ko、hisi_sec2.ko、hisi_hpre.ko、hisi_zip.ko、hisi_rde.ko是否在位。

```
[root@localhost ~]# lsmod | grep uacce
uacce                262144  2 hisi_hpre,hisi_qm,hisi_sec2,hisi_zip,hisi_rde
```

- 步骤2** 检查/usr/lib64（rpm方式安装时目录）或者/usr/local/lib（源码方式安装时目录）和OpenSSL安装目录是否有加速器引擎库，且建立正确的软连接。

```
[root@localhost home]# ll /usr/local/lib/engines-1.1/ |grep kae
#查询kae是否正确安装并建立软连接，如果有正确安装显示如下内容
[root@localhost home]# ll /usr/local/lib/engines-1.1/ |grep kae
lrwxrwxrwx. 1 root root    22 Nov 12 02:33 kae.so -> kae.so.1.0.1
lrwxrwxrwx. 1 root root    22 Nov 12 02:33 kae.so.0 -> kae.so.1.0.1
-rwxr-xr-x. 1 root root 112632 May 25 2019 kae.so.1.0.1
[[root@localhost home]#
[root@localhost home]# ll /usr/lib64/ | grep libwd
#查询wd是否正确安装并建立软连接，如果有正确安装显示如下内容
lrwxrwxrwx. 1 root root    14 Nov 12 02:33 libwd.so -> libwd.so.1.0.1
lrwxrwxrwx. 1 root root    14 Nov 12 02:33 libwd.so.0 -> libwd.so.1.0.1
-rwxr-xr-x. 1 root root 137120 May 25 2019 libwd.so.1.0.1
[root@localhost home]#
```

- 步骤3** 检查OpenSSL引擎库的路径是否能通过export命令进行导出。

```
[root@localhost home]# echo $OPENSSL_ENGINES
[root@localhost output]# export OPENSSL_ENGINES=/usr/local/lib/engines-1.1
```

```
[root@localhost output]# echo $OPENSSL_ENGINES
/usr/local/lib/engines-1.1
```

----结束

6.2 安装完加速器引擎之后，查找不到加速器设备

故障现象

安装完加速器引擎之后，查找不到加速器设备。

解决方法

步骤1 检查虚拟文件系统下是否有相应设备。正常情况下有如下相应的加速器设备。

```
[root@localhost rpm]# ls -al /sys/class/uacce/
total 0
lrwxrwxrwx. 1 root root 0 Nov 14 03:45 hisi_hpre-2 -> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/
uacce/hisi_hpre-2
lrwxrwxrwx. 1 root root 0 Nov 14 03:45 hisi_hpre-3 -> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/
uacce/hisi_hpre-3
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_rde-4 -> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-4
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_rde-5 -> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-5
lrwxrwxrwx. 1 root root 0 Nov 14 08:39 hisi_sec-0 -> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/
uacce/hisi_sec-0
lrwxrwxrwx. 1 root root 0 Nov 14 08:39 hisi_sec-1 -> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/
uacce/hisi_sec-1
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_zip-6 -> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/
uacce/hisi_zip-6
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_zip-7 -> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/
uacce/hisi_zip-7
```

步骤2 若要使用hpre设备但是在**步骤1**中未查询到，请按**6.1 初始化失败**排查加速引擎软件是否已正确安装。

步骤3 若**步骤2**已确认加速引擎软件正确安装，请排查通过lspci命令查看物理设备是否存在。

```
[root@localhost output]# lspci | grep HPRE
79:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd. HiSilicon HPRE Engine (rev 21)
b9:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd. HiSilicon HPRE Engine (rev 21)
[root@localhost output]#
[root@localhost output]# lspci | grep SEC
76:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd. HiSilicon SEC Engine (rev 21)
b6:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd. HiSilicon SEC Engine (rev 21)
[root@localhost output]#
[root@localhost output]# lspci | grep RDE
78:01.0 RAID bus controller: Huawei Technologies Co., Ltd. HiSilicon RDE Engine (rev 21)
b8:01.0 RAID bus controller: Huawei Technologies Co., Ltd. HiSilicon RDE Engine (rev 21)
[root@localhost output]#
[root@localhost output]# lspci | grep ZIP
75:00.0 Processing accelerators: Huawei Technologies Co., Ltd. HiSilicon ZIP Engine (rev 21)
b5:00.0 Processing accelerators: Huawei Technologies Co., Ltd. HiSilicon ZIP Engine (rev 21)
[root@localhost output]#
```

步骤4 若**步骤3**未查询到相应的物理设备，请确认以下，不分先后：

- 确认是否已导入加速器许可证，若未导入，请参考《[TaiShan 机架服务器 iBMC \(V300及以上\) 用户指南](#)》中“许可证管理”章节，导入加速器许可证。导入加速器许可证之后，需要掉电重启BMC，使能License。

- 确认BMC和BIOS版本是否支持加速器特性。

----结束

6.3 升级加速器驱动失败

故障现象

升级加速器驱动后，重启系统驱动版本仍为旧版本。

可能原因

在升级加速器驱动前，系统更新了其他驱动包，这些驱动包可能重新更新了引导文件系统initramfs，将未升级前的加速器驱动一起更新到了initramfs文件系统中。例如系统更新了网卡驱动，或者人为更新了initramfs文件系统，导致系统重启时优先从initramfs文件系统中加载加速器驱动。

处理步骤

升级加速器驱动版本后，通过执行dracut --force命令重新更新initramfs文件系统。

6.4 操作系统安装完 OpenSSL 新版本后提示相关接口符号未找到

故障现象

例如执行rpm命令时出现如下报错：

```
rpm: relocation error: /lib64/librpmio.so.8: symbol EVP_md2 version OPENSSL_1_1_0 not defined in file libcrypto.so.1.1 with link time reference
```

例如执行系统OpenSSL命令时出现如下报错：

```
/usr/bin/openssl: relocation error: /usr/bin/openssl: symbol EVP_md2 version OPENSSL_1_1_0 not defined in file libcrypto.so.1.1 with link time reference
```

可能原因

OpenSSL动态库路径导出到LD_LIBRARY_PATH或者OpenSSL动态库安装在/usr/local/lib下时——动态库搜索路径配置文件/etc/ld.so.conf中设定了/usr/local/lib，此时若系统工具或命令调用OpenSSL的动态库，会优先调用到用户安装的库而非系统原有库，导致冲突。

处理步骤

推荐按[3.1.3 安装OpenSSL](#)安装OpenSSL软件；按[3.1.4 通过rpm包安装加速引擎软件](#)、[3.1.5 通过deb包安装加速引擎软件](#)和[3.1.6 通过源码方式安装加速引擎软件](#)安装加速引擎软件包。如果系统需要将/usr/local/lib路径导出到LD_LIBRARY_PATH环境变量或配置到/etc/ld.so.conf中，则需要通过如下命令指定安装路径与动态库路径安装OpenSSL源码。


```
./config --prefix=/usr/local/openssl -Wl,-rpath,/usr/local/openssl/lib  
make  
make install
```

如果加速器为rpm方式安装，则作如下修改：

```
rpm -ivh libkae-1.0.1-1.euler2.0.aarch64.rpm --prefix=/usr/local/openssl/lib/engines-1.1
```

如果加速器为源码方式安装，则执行如下命令：

```
cd KAE  
chmod +x configure  
./configure --openssl_path=/usr/local/openssl  
make clean && make  
make install
```

A 附录

A.1 术语

表 A-1 术语

术语	术语解释
A	
AES	高级加密标准（Advanced Encryption Standard, AES），在密码学中又称Rijndael加密法，是美国联邦政府采用的一种区块加密标准。这个标准用来替代原先的DES，已经被多方分析且广为全世界所使用。经过五年的甄选流程，高级加密标准由美国国家标准与技术研究院（NIST）于2001年11月26日发布于FIPS PUB 197，并在2002年5月26日成为有效的标准。2006年，高级加密标准已然成为对称密钥加密中最流行的算法之一。
S	
SM3	SM3主要用于数字签名及验证、消息认证码生成及验证、随机数生成等，其算法公开。据国家密码管理局表示，其安全性及效率与SHA-256相当。
SM4	商用密码体系中，SM4主要用于数据加密，其算法公开，分组长度与密钥长度均为128bit，加密算法与密钥扩展算法都采用32轮非线性迭代结构，S盒为固定的8比特输入8比特输出。SM4.0中的指令长度被提升到大于64K（即64×1024）的水平，这是SM 3.0规格（渲染指令长度允许大于512）的128倍。
R	
RSA	RSA算法的名字以发明者的名字命名：Ron Rivest, Adi Shamir和Leonard Adleman。RSA属于非对称加密算法，自1977年提出，至今经历了各种攻击的考验，逐渐为人们接受，被认为是目前最优秀的非对称加密方案之一。
D	

术语	术语解释
DH	DH密钥交换算法 迪菲 - 赫尔曼密钥交换 (Diffie-Hellman key exchange, 简称“D-H”)是一种安全协议。它可以让双方在完全没有对方任何预先信息的条件下通过不安全信道建立起一个密钥。这个密钥可以在后续的通讯中作为对称密钥来加密通讯内容。

A.2 缩略语

表 A-2 缩略语

缩略语	英文全称	中文全称
H		
HPRE	High Performance RSA Engine	高性能RSA加速引擎
K		
KAE	Kunpeng Accelerator Engine	鲲鹏加速器引擎
S		
SEC	Security Engine	硬件安全加速引擎

A.3 修订记录

发布日期	修订记录
2020-12-14	第七次正式发布。 修改 3.1.6 通过源码方式安装加速引擎软件 。
2020-11-05	第六次正式发布。 <ul style="list-style-type: none">更新1 产品简介。修改4.1.4 KAE引擎加速Nginx应用。新增4.1.8 MD5硬件加速调优。修改4.2 zlib加速库的使用。
2020-10-15	第五次正式发布。 <ul style="list-style-type: none">1 产品简介中EBC改为ECB。3.1.3 安装OpenSSL中增加OpenSSL源码下载地址。
2020-07-22	第四次正式发布。 更改文档名称。

发布日期	修订记录
2020-06-30	<p>第三次正式发布。</p> <ul style="list-style-type: none">● 3.1.1 安装简介中增加dpkg安装的说明。● 修改3.1.2 安装前准备中的环境要求、软件包说明。● 3.1.5 通过deb包安装加速引擎软件中安装后检查增加OpenSSL命令验证是否生效的步骤。● 3.3 升级加速引擎软件中增加deb包的升级方式。● 3.4 卸载加速引擎软件增加deb包的卸载方式。● 4.1.1 加速引擎使用说明增加MD5和SM3/SM4的相关内容。● 修改4.1.3 通过OpenSSL配置文件openssl.cnf使用KAE引擎中配置文件参数说明。● 修改4.2 zlib加速库的使用中的链接方式说明。● 4.1.6 加速引擎在KVM虚拟机中的使用和4.1.7 加速引擎在docker中的使用中增加PF虚拟化出VF数量推荐说明。
2020-04-30	<p>第二次正式发布。 文档优化。</p>
2020-03-20	<p>第一次正式发布。</p>