

Atlas 300 AI 加速卡
1.0.0

DDK 安装指南（ EulerOS ）（ 型号 3000 ）

文档版本 01
发布日期 2020-03-30



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 简介.....	1
2 软件包获取.....	2
3 检验软件包完整性.....	3
4 环境准备.....	4
5 DDK 安装.....	8
5.1 安装.....	8
5.2 常用操作.....	9
5.2.1 卸载 DDK.....	9
5.2.2 查询 DDK 版本.....	9
6 FAQ.....	10
6.1 安装 DDK 过程中提示 Warning: DDK functions except TE have been successfully installed.....	10
7 附录.....	11
7.1 软件包说明.....	11
7.2 开源第三方库说明.....	12

1 简介

本文介绍独立部署DDK（Device Development Kit）的安装过程以及常见问题处理方法，适用于基于后台命令行的方式进行开发的场景。

DDK为用户提供基于NPU的数字开发者套件。DDK可以用于构建相关工程的编译环境。不同的发布包里集成了不同NPU形态的DDK。当前版本的DDK集成了TE、DVPP、流程编排等组件的依赖库和头文件，用户可以通过makefile编译相应的工程文件。

2 软件包获取

在工具安装前，请准备如下软件包。

DDK安装包：请访问<https://support.huawei.com/enterprise>，在搜索栏中搜索对应产品名进入产品节点下，单击“软件”页签，获取DDK安装包。

软件包说明如表2-1所示。

表 2-1 软件包说明

安装包	包内容	软件包完整性校验文件
MSpore_DDK-{version}-aarch64.eulerOS-aarch64.eulerOS-aarch64.miniOS.tar.gz	DDK安装包。	MSpore_DDK-{version}-aarch64.eulerOS-aarch64.eulerOS-aarch64.miniOS.tar.gz.asc

3 检验软件包完整性

为了防止软件包在传递过程或存储期间被恶意篡改，下载软件包时需下载对应的数字签名文件用于完整性验证。

在软件包下载之后，请参考《OpenPGP签名验证指南》，对从Support网站下载的软件包进行PGP数字签名校验。如果校验失败，请不要使用该软件包，先联系华为技术支持工程师解决。

使用软件包安装/升级之前，也需要按上述过程先验证软件包的数字签名，确保软件包未被篡改。

运营商客户请访问：<https://support.huawei.com/carrier/digitalSignatureAction>

企业客户请访问：<https://support.huawei.com/enterprise/zh/tool/pgp-verify-TL1000000054>

4 环境准备

环境要求

安装DDK的环境，所要求的硬件及操作系统要满足以下条件。

表 4-1 EulerOS 系统版本信息

类别	版本限制	获取方式	注意事项
操作系统	EulerOS-V2.0SP8	请联系华为技术支持获取如下推荐的版本： EulerOS-V2.0SP8-aarch64-dvd.iso	建议使用内核版本为4.19.36-vhulk1907.1.0.h410.eulerosv2r8.aarch64的EulerOS。
Python	Python2: 2.7+ Python3: 3.5+	请参见 安装依赖 。	-

准备 DDK 安装用户（可选）

您可以使用任意用户（root或非root用户）安装DDK。

- 若使用root用户安装，则不需要操作该章节。
- 若使用已存在的非root用户安装，须保证该用户对\$HOME目录具有读写以及可执行权限。
- 若使用新的非root用户安装，请参考如下步骤进行创建，如下操作请在root用户下执行。本手册以该种场景为例执行DDK的安装。
 - 执行以下命令创建DDK安装用户并设置该用户的\$HOME目录，并确保用户对\$HOME目录具有读写以及可执行权限。

```
useradd -d /home/username -m username
```
 - 执行以下命令设置密码。

```
passwd username
```

说明

`username`为安装DDK的用户名，该用户的umask值不能大于0027：

- 若要查看umask的值，则执行命令：**umask**
- 若要修改umask的值，则执行命令：**umask 新的取值**

若创建的非root用户为**HwHiAiUser**，且DDK部署在Host侧服务器，使用该用户进行离线模型转换时，日志默认输入到host侧/var/dlog目录中，您也可以通过设置如下环境变量，使日志默认输出到屏幕上。

```
export SLOG_PRINT_TO_STDOUT=1
```

- 更改文件描述符限制大小。
 - a. 使用root用户打开系统配置文件：“/etc/security/limits.conf”，命令为：

```
vi /etc/security/limits.conf
```
 - b. 在limits.conf文件末尾添加如下内容：

```
* soft nproc 65535
* hard nproc 65535
* soft nofile 65535
* hard nofile 65535
```

说明

“*”代表所有用户，建议替换为DDK安装用户名。

- c. 保存文件并退出：**:wq!**。

配置 DDK 安装用户权限（可选）

当使用非root用户安装时，需要操作该章节，否则请忽略。

DDK安装前需要下载相关依赖软件，下载依赖软件需要使用**sudo yum**等权限，请以root用户执行如下操作。

1. 执行以下命令打开“/etc/sudoers”文件：

```
chmod u+w /etc/sudoers
vi /etc/sudoers
```
2. 在该文件“root ALL=(ALL:ALL) ALL”下面增加如下内容：

```
username ALL=(ALL:ALL) NOPASSWD:SETENV: /usr/bin/yum
```

“username”为执行安装脚本的普通用户名。
3. 添加完成后，执行**:wq!**保存文件。
4. 执行以下命令取消“/etc/sudoers”文件的写权限。

```
chmod u-w /etc/sudoers
```

说明

请确保“/etc/sudoers”文件的最后一行为“#includedir /etc/sudoers.d”，如果没有该信息，请手动添加。

配置本地源

由于EulerOS镜像存在小版本间差异，为避免兼容性差异，建议使用本地镜像源进行安装。

请以root用户执行如下操作。

步骤1 上传EulerOS-V2.0SP8-aarch64-dvd.iso至服务器中，执行下列命令进行镜像挂载。

命令：**mount -o loop 欧拉镜像.iso 镜像挂载路径**

示例：**mount -o loop EulerOS-V2.0SP8-aarch64-dvd.iso /mnt**

步骤2 进入源管理文件路径/etc/yum.repos.d，备份repo文件。

命令：**cp 源管理文件.repo 源管理文件.repo.bak**

示例：**cp euleros_aarch64.repo euleros_aarch64.repo.bak**

步骤3 替换源管理文件如下。

命令：**vim源管理文件.repo**

示例：**vim euleros_aarch64.repo**

```
[base]
name=EulerOS-2.0SP8 base
baseurl=file:///mnt
enabled=1
gpgcheck=0
```

说明

如果用镜像挂载路径非/mnt，请自行替换baseurl的对应值。

步骤4 更新软件信息。

```
yum clean all
yum makecache
```

----结束

安装依赖

请使用**su - username**命令切换到DDK安装用户执行如下操作，安装DDK工具依赖的python等组件。

步骤1 检查系统是否安装python2及python3。

DDK安装依赖python2及python3环境，使用命令**python -V**、**python3 -V**检查系统是否安装python2及python3。

若返回如下信息则说明已经安装。否则执行**步骤2**，安装python2及python3。

```
python 2.7.15
python 3.7.0
```

步骤2 安装python2及python3。

1. 检索源列表中是否存在可用python2及python3。

```
sudo yum list "*python2*"
sudo yum list "*python3*"
```

2. 如果源列表中存在python2.aarch64及python3.aarch64镜像软件（不同镜像源软件名可能有所差异）。请执行如下命令，安装python2及python3。

命令：**sudo yum install 镜像软件名称**

示例：**sudo yum install python3.aarch64**

步骤3 安装python2及python3依赖。

1. 执行如下命令，查找已安装的numpy及decorator安装库：

```
rpm -qa | grep numpy  
rpm -qa | grep decorator
```

回显信息如下，表示存在python2及python3的numpy及decorator安装库。若存在未安装的numpy及decorator安装库，请执行[步骤3.2](#)~[步骤3.3](#)，安装缺失的numpy及decorator安装库。

```
python3-decorator-4.3.0-1.eulerosv2r8.noarch  
python3-numpy-1.15.1-1.eulerosv2r8.aarch64  
python2-decorator-4.3.0-1.eulerosv2r8.noarch  
python2-numpy-1.15.1-1.eulerosv2r8.aarch64
```

2. 搜索可用的numpy及decorator安装库：

```
sudo yum list "*numpy*"  
sudo yum list "*decorator*"
```

回显信息如下，表示存在可用的numpy及decorator安装库。

```
python2-numpy.aarch64  
python2-decorator.noarch  
python3-numpy.aarch64  
python3-decorator.noarch
```

说明

若无法查询到python3的numpy安装库，请联系华为技术支持获取内核版本为4.19.36-vhulk1907.1.0.h529.eulerosv2r8.aarch64及以上的EulerOS镜像包。重新[配置本地源](#)。

3. 执行如下命令，安装缺失的python2及python3的numpy及decorator安装库。

命令：**sudo yum install *numpy*依赖库**

示例：**sudo yum install python3-numpy.aarch64**

命令：**sudo yum install *decorator*依赖库**

示例：**sudo yum install python3-decorator.noarch**

----结束

恢复源配置（可选）

后续如果不继续使用本地源，可执行如下操作步骤将源管理文件恢复。

步骤1 进入源管理文件路径/etc/yum.repos.d，恢复repo文件。

cp -f euleros_aarch64.repo.bak euleros_aarch64.repo

步骤2 取消镜像文件的挂载。

命令：**umount 挂载路径**

示例：**umount /mnt**

----结束

5 DDK 安装

本节介绍DDK的安装方法以及常用操作。

5.1 安装

5.2 常用操作

介绍DDK安装以及使用过程中的常用操作。

5.1 安装

前提条件

已完成[4 环境准备](#)和[3 检验软件包完整性](#)。

操作步骤

请切换到DDK安装用户执行如下操作。

步骤1 执行如下命令解压安装包。

```
tar -zxvf MSpore_DDK****tar.gz
```

操作时请将`MSpore_DDK****.tar.gz`替换为实际安装包，解压后文件说明请参见[表7-1](#)。

步骤2 执行如下命令安装DDK。

```
bash install.sh DDK安装目录
```

DDK安装目录在执行安装时会自动创建，例如安装目录设置为“\$HOME/tools/che/ddk”，则在安装过程中会自动创建“tools/che/ddk”目录。您也可以自己指定安装路径。

若出现“Successfully installed the DDK!”信息，则表明DDK安装成功。

说明

- 若安装时提示“Warning: DDK functions except TE have been successfully installed.”则参见[6.1 安装DDK过程中提示Warning: DDK functions except TE have been successfully installed.](#)解决。
- DDK安装完毕后，安装路径下“\$HOME/tools/che/ddk/ddk/toolchains/profiler”的Profiling功能不可用，如果用户需要使用Profiling功能，则请参见《Ascend 310 Mind Studio工具安装指南》手册（请根据Mind Studio安装服务器选择对应形态的安装手册）安装Mind Studio与DDK。

----结束

5.2 常用操作

介绍DDK安装以及使用过程中的常用操作。

5.2.1 卸载 DDK

如果您想更新DDK版本，则请卸载当前版本然后再按照5.1-安装步骤安装新的版本，本章节给出DDK的卸载方法。

进入DDK安装路径，本示例为“\$HOME/tools/che/ddk/ddk/scripts”，使用DDK安装用户执行如下命令卸载DDK：

```
./uninstall.sh
```

如果出现如下所示信息，则说明DDK卸载成功：

```
Info: ide_daemon pem uninstall succ  
Starting to remove ddk dir.
```

5.2.2 查询 DDK 版本

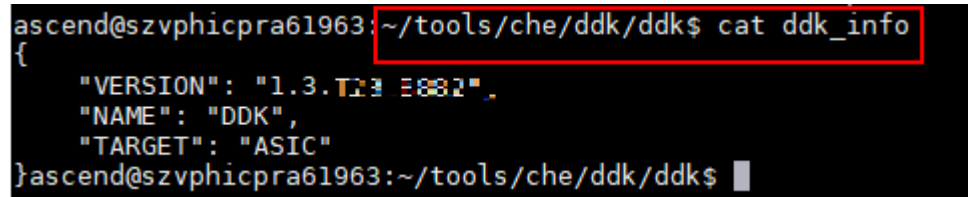
DDK安装成功后，您可以在DDK安装目录下查看DDK版本号。

例如本示例DDK安装目录为“\$HOME/tools/che/ddk”，则您可以在“\$HOME/tools/che/ddk/ddk”目录下执行如下命令查看DDK版本号：

```
cat ddk_info
```

返回结果如[图5-1](#)所示。

图 5-1 查询 DDK 版本号



```
ascend@szvphicpra61963:~/tools/che/ddk/ddk$ cat ddk_info  
{  
  "VERSION": "1.3.1.1",  
  "NAME": "DDK",  
  "TARGET": "ASIC"  
}ascend@szvphicpra61963:~/tools/che/ddk/ddk$
```

其中：

- VERSION：表示DDK版本号，查询结果请以实际环境中的为准。
- TARGET：表示DDK的运行环境，目前包括ASIC和Atlas DK。

6 FAQ

6.1 安装DDK过程中提示Warning: DDK functions except TE have been successfully installed.

6.1 安装 DDK 过程中提示 Warning: DDK functions except TE have been successfully installed.

问题描述

执行DDK安装脚本后，安装过程中提示“error : install te-0.4.0.egg failed”、“error : install topi-0.4.0.egg failed”等信息，安装最后提示“Warning: DDK functions except TE have been successfully installed.”，DDK安装成功，但是TE相关功能无法使用。

可能原因

该问题可能是由于安装DDK的服务器，root用户下存在太老的setuptools版本（0.9.8），切换到DDK安装用户（非root用户）安装DDK时，仍旧读取的是root用户下的setuptools路径，DDK安装用户下的setuptools未生效。

解决方法

若出现上述问题，则使用DDK安装用户先执行如下命令设置环境变量，使DDK安装用户下的setuptools生效。然后重新安装DDK。

```
export PATH=$HOME/bin:$HOME/.local/bin:$PATH
```

7 附录

- 7.1 软件包说明
- 7.2 开源第三方库说明

7.1 软件包说明

DDK安装包解压后包的详细内容请参见表7-1。

表 7-1 软件包说明

安装包	包内容	使用场景
MSpore_DDK-{version}-<uihost arch.os>-<host arch.os>-<device arch.os>.tar.gz 各参数解释请参见表 7-2。	ddk.tar.gz	DDK套件安装包。
	install.sh	安装脚本。
	check_sha.sh	校验以上两个文件的完整性，install.sh执行过程中会自动调用该脚本进行完整性校验。

表 7-2 DDK 安装包命名规范

参数	说明
{version}	版本号
<uihost arch.os>	UIhost侧 CPU架构和操作系统及版本，如：x86_64.ubuntu16.04、x86_64.centOS7.4等
<host arch.os>	Host侧 CPU架构和操作系统及版本，如：x86_64.ubuntu16.04、x86_64.centOS7.4等

参数	说明
<device arch.os>	Device侧 CPU架构和操作系统及版本，如： aarch64.ubuntu16.04、aarch64.miniOS等 <ul style="list-style-type: none">开发者板形态（ Atlas 200 DK ） device侧架构： aarch64.ubuntu16.04非开发者板形态（ Atlas 300等 ） device侧架构： aarch64.miniOS

7.2 开源第三方库说明

cereal

cereal是一个开源的（ BSD License ）、轻量级的、支持C++11特性的、仅仅包含头文件实现的、跨平台 的C++序列化库。它可以将任意的数据类型序列化成不同的表现形式，比如二进制、XML格式或JSON。当前使用版本： 1.2.2

详细介绍可参考cereal官网：<http://uscilab.github.io/cereal/>

gflags

gflags (the Global Flags Editor)是google的一个开源的处理命令行参数的库，比 getopt 更方便，更功能强大，使用c++开发，具备python接口。当前使用版本： 2.2.1

详细介绍可参考gflags官网：<https://github.com/gflags/gflags>

glog

glog是Google的一个开源库，用于实现应用级别的logging。它提供了一系列类似于C++流风格的logging API，以及一些预定义的宏。它有点类似于C里面的assert，但是它具备更丰富的输出信息以及使用灵活性。

详细介绍可参考glog官网：<https://github.com/google/glog>

opencv

OpenCV的全称是Open Source Computer Vision Library，是一个跨平台的计算机视觉库。OpenCV可用于开发实时的图像处理、计算机视觉以及模式识别程序。当前使用版本： 3.4.2

详细介绍可参考opencv官方网站：<https://opencv.org/>

Protobuf

Protobuf(Google Protocol Buffers)是Google提供一个具有高效的协议数据交换格式工具库；它是一种平台无关、语言无关、可扩展且轻便高效的序列化数据结构的协议，可以用于网络通信和数据存储。当前使用版本： 3.5.1

详细介绍可参考google官方网站：<https://developers.google.com/protocol-buffers/>